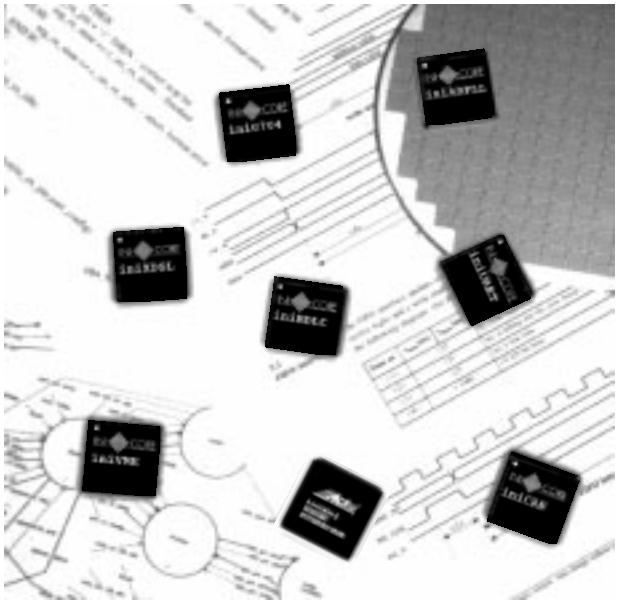
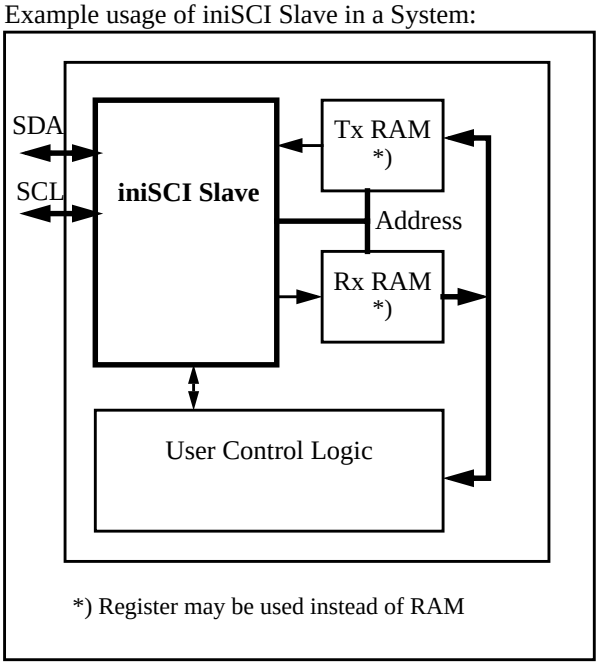
# iniSCI Slave *I2C compatible* data sheet



**Features:**

- I2C-Compatible
- Slave Function
- 7Bit Slave ID Address
- Automatic Incremented Address Pointer(AP)
- AP Initialization by transferring Word Address on I2C Bus
- Message Acknownledgement
- Customizable for Special Requirements
- Structured Model Description (SD)
- Synchronous, Synthesisable VHDL Model

Example usage of iniSCI Slave in a System:




*) Register may be used instead of RAM

INICORE - the reliable Core and System Provider. We provide high quality IP, design expertise and leading edge silicon to the industry.

Inicore's **iniSCI Slave** is a synthesizable, flexible, and structured VHDL implementation of a Serial Controller Interface (SCI) that uses a two-wire bus for communicating between integrated circuits or standard peripherals like smart LCDs and keypads.

The **iniSCI Slave** complies with the definition of the 'Inter-Integrated Circuit Bus' (I2C). It is intended to be used as an interface block between the I2C buslines (SDA,SCL) and two mailbox memories. A local Mail-bus interface consisting of data, address and control lines allows you to interface registers or RAM's up to a size of 256 bytes.

The **iniSCI Slave** contains the entire physical and data link layers, allowing to handle bus timing and frame generation/extraction, and thus reducing overhead from the system application.
INICORE's strategy is not to compete with the standard chip manufacturers, but to use the ASIC and FPGA technologies for `system on chip' design, which demands standards like I2C with customer specific user functions.

**US Sales Office:**
**INICORE INC.**
5600 Mowry School Road, Suite 180,
Newark, CA 94560
Tel: 510 445 1529  Fax: 510 656 0995
E-mail: ask_us@inicore.com
Web: www.inicore.com

**INICORE AG**
Mattenstrasse 6a,CH-2555 Brügg, Switzerland
Tel: ++41 32 374 32 00, Fax: ++41 32 374 32 01
E-mail: ask_us@inicore.ch
Web: www.inicore.ch

Ref.Nr.: 402-DS-11 / 07/99

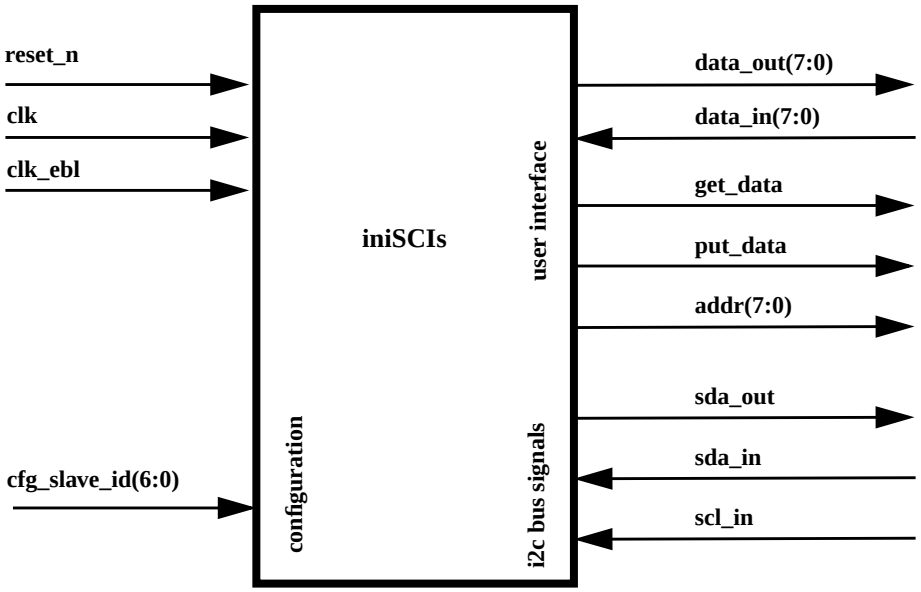
## 1 Overview

This document describes the entity and the functional behavior of the iniSCI Slave core. The iniSCIs complies with the definition of the 'Inter-Integrated Circuit Bus' and is intended to be used as an interface block between the I2C bus (SDA, SCL) and the user defined application logic (e.g. RAM, FIFO, registers,...)

The user side interface consists of data, address and control lines, which can easily be interfaced to registers and RAM's up to a size of 256 bytes.

### 1.1 Structure of iniSCI Slave

**Pinout of iniSCI Slave:**




### 1.2 Recommendation

The iniSCIs requires a local clock. Its frequency has an influence on the glitch suppression characteristic. For application with very high clock, a clock enable pin is available for slowing down the effective clock rate in the module. This way, fully synchronous design methods can be always used. It is recommended to use a effective clock in the range of 2-5 MHz. Absolute min. clock rate is 1MHz and below 2MHz, glitch tolerance is reduced. Glitches must be shorter than one effective clock cycle.

## 2 IO description

The following part lists the input and output ports of INICORE's iniSCI Slave core and explains their functionality.

### 2.1 General inputs

These pins are use to clock and initialize the whole iniSCIs core . There are no other clocks in this core.

| Pin Name | Type | Description |
|---|---|---|
| clk | in | System clock, min. frequency is 1 MHz.<br>The max. frequency is limited by the circuit speed. |
| clk_ebl | in | Clock enable. A high level enables the core logic . It is recommended to use the clk_ebl signal if the clk frequency exceeds 5MHz. For clk frequency lower 5MHz clk_ebl should be fixed at logic '1'. |
| reset_n | in | Asynchronous system reset, active low<br>All registers are initialized. |

### 2.2 iniSCI Slave IO's

#### 2.2.1 Configuration of iniSCI Slave

The configuration pins are used to select the operating mode. They are static and must be stable during operation.

| Pin Name | Type | Description |
|---|---|---|
| cfg_slave_id(6:0) | in | Slave identification. This seven bits defines the physical I2C slave address. Signals must be stable during transmission. |

#### 2.2.2 iniSCI Slave user interface

| Pin Name | Type | Description |
|---|---|---|
| data_out(7:0) | out | Data output. Data received from the I2C master. The received data byte is valid when the signal put_data is logic '1'. |
| data_in(7:0) | in | Data input. Data to be transmitted to the master. The data byte will be registered (fetched) in the iniSCIs at rising clk edge and when get_data signal is logic '1'. |
| get_data | out | Get data event. Active high pulse for one clk cycle. At the end of the high cycle, data_in value is registered in the iniSCIs at rising clk edge and will be transmitted to the master. |
| put_data | out | Put data event. Active high pulse for one clk cycle. It signals that the data at the data_out port is valid. This signal can be used as a write enable signal to register the received data byte. |
| addr(7:0) | out | Address pointer. This pointer can be used to address a receive and a transmit memory (or register bank). It is stable during data_get and data_put pulses and is initialized when a valid word address is received and incremented after each sent/received data byte. |

### 2.2.3 iniSCIs I2C bus signals

| Pin Name | Type | Description |
|---|---|---|
| sda_out | out | SDA. Serial bus data out. |
| sda_in | in | SDA. Serial bus data in. Schmitt trigger input ports should be used. |
| scl_in | in | SCL. Serial bus clock. Schmitt trigger input ports should be used. |

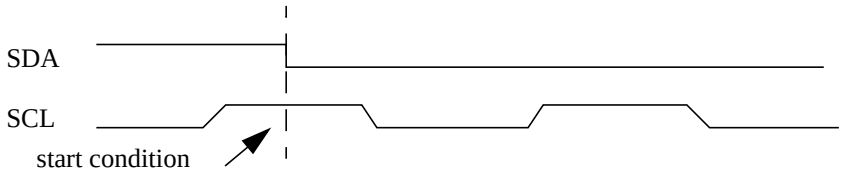
# 3 Functional description

## 3.1 Initialization

A logic ‘0’ at reset_n will initialize the iniSCIs. The address pointer will be = “00000000”.

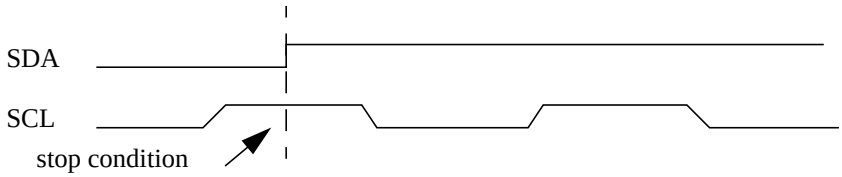
## 3.2 Start condition

A start condition is detected if a falling edge on SDA occurs during a high level on SCL. After a start condition, the interface is ready for receiving the device address (Slave Address). If a start condition occurs during a running transfer, the transfer is aborted.



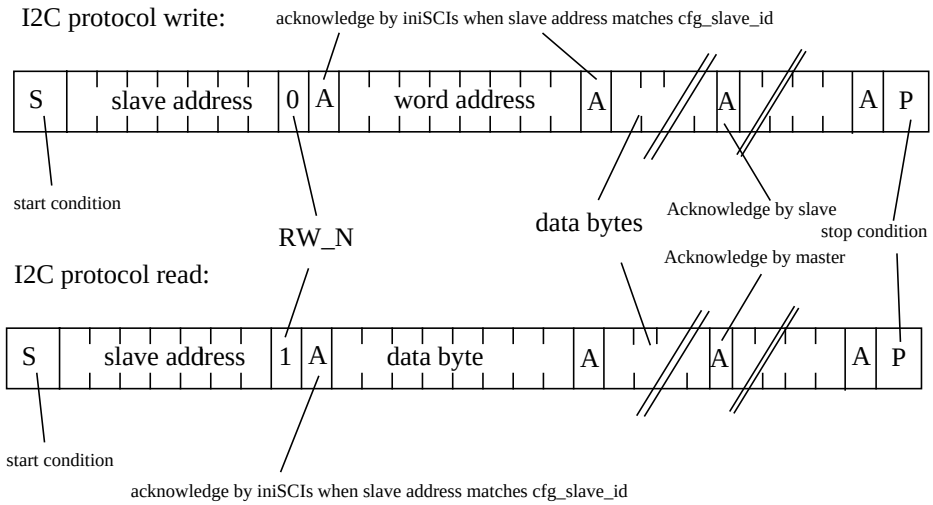

## 3.3 Stop condition

A stop condition is detected, if a rising edge on SDA occurs during a high level on SCL. After a stop condition the iniSCI returns to the idle state and is ready for a new start condition.




## 3.4 Slave addressing

The first byte after the start condition is interpreted as the slave address. If it matches the local slave ID (cfg_slave_id) the iniSCIs responds with a ACK pulse and is ready for further byte transfers. If the slave address does not match, the iniSCIs returns to idle state.

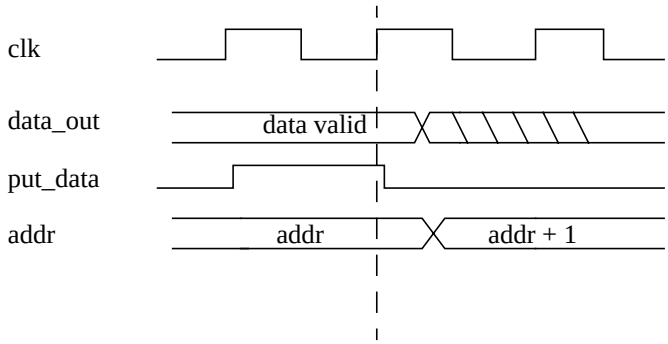


**3.5 Word addressing**

If the slave address did match the slave ID and the R_WN bit is low level then the iniSCIs is ready to receive a one byte word address. The word address is stored and defines the address (addr) signal for fetching or putting data bytes. The address remains unchanged until a new word address transfer occurs or a data byte is transferred.
After reception of a word address the iniSCI will generate a I2C acknowledge pulse.

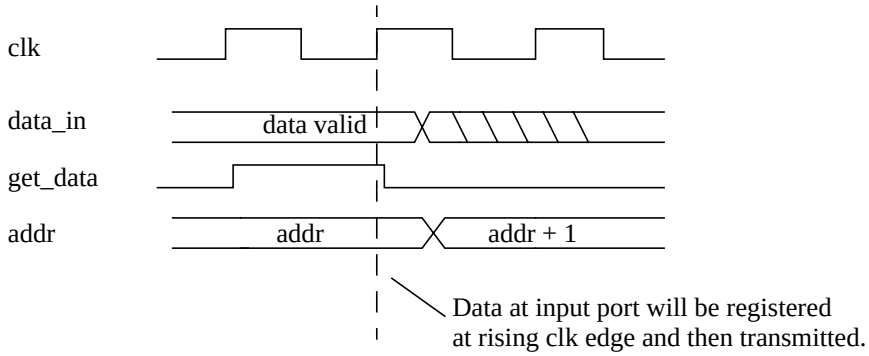
**3.6 Writing data bytes**

After a word address transfer the iniSCIs is ready to receive one or several data bytes. It will transfer these data bytes to the data_out port. The first address value is given by the word address transferred previously. After each data byte transfer the address is autoincremented by one.
After each received data byte the iniSCIs will generate the ACK pulse.
Data byte transfers are terminated after detection of stop condition.




**3.7 Reading data bytes**

If the slave address did match the slave ID and the RW_N bit is at high level the iniSCIs is ready to transmit a string of data bytes. Unless a stop condition occurs it will transmit one data byte, which is fetched from the data_in port. If the I2C master responds by an ACK pulse, the next data byte is transmitted by iniSCIs. If no ACK from the I2C master occurs, data transmission is terminated and the iniSCIs returns to idle state.
After each data byte transfer the address pointer is autoincremented by one.
Data byte transfers are terminated after detection of stop condition.




**3.8 Noise and glitches**

The noise immunity of the iniSCIs interface depence on the internal clock rate (clk, clk_ebl). A transition is only detected, if two successive samples show the same level.

Use schmitt trigger input buffer for SDA and SCL input ports.