IP Documentation

I2C Master IP Core v.1.1
Specifications

I2C Master IP Core v.1.1 is an IP Core for synchronous half-duplex serial communication using I2C protocol. The Core complies with standard I2C specifications and is a pure RTL design which can be easily ported across platforms.

- 8-bit data frame, 7-bit slave address support.
- Supports two modes of operation: Fast (400 kHz SCLK), Slow (100 kHz SCLK).
- Open drain SDA line; external pull-ups required.
- Uni-directional SCLK control. Only Master can control SCLK.
- Single interrupt.
## I/O Ports

<table>
<thead>
<tr>
<th>Signal</th>
<th>Direction</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>in</td>
<td>-</td>
<td>Clock</td>
</tr>
<tr>
<td>rstn</td>
<td>in</td>
<td>-</td>
<td>Asynchronous reset **</td>
</tr>
<tr>
<td>mode_in</td>
<td>in</td>
<td>-</td>
<td>To select mode of operation: ‘0’ – Slow, ‘1’- Fast</td>
</tr>
<tr>
<td>start_in</td>
<td>in</td>
<td>-</td>
<td>To start/repeated start a transaction</td>
</tr>
<tr>
<td>stop_in</td>
<td>in</td>
<td>-</td>
<td>To stop the ongoing transaction</td>
</tr>
<tr>
<td>wr_in</td>
<td>in</td>
<td>-</td>
<td>To write data on to bus</td>
</tr>
<tr>
<td>rd_in</td>
<td>in</td>
<td>-</td>
<td>To read data on bus</td>
</tr>
<tr>
<td>en_ack_in</td>
<td>in</td>
<td>-</td>
<td>To enable Master acknowledgement</td>
</tr>
<tr>
<td>ack_out</td>
<td>out</td>
<td>-</td>
<td>Status of Slave acknowledgement: ‘1’ – Success, ‘0’ – Failed</td>
</tr>
<tr>
<td>intr_out</td>
<td>out</td>
<td>-</td>
<td>Interrupt signal – this flag is set for each byte transfer, and stop signalling</td>
</tr>
<tr>
<td>data_in</td>
<td>in</td>
<td>8</td>
<td>Byte to be sent on the bus</td>
</tr>
<tr>
<td>data_out</td>
<td>out</td>
<td>8</td>
<td>Byte read out from the bus</td>
</tr>
<tr>
<td>sda</td>
<td>inout</td>
<td>-</td>
<td>Serial Data line</td>
</tr>
<tr>
<td>sclk</td>
<td>out</td>
<td>-</td>
<td>Serial Clock generated by Master</td>
</tr>
</tbody>
</table>

** active-low signal
I2C Transaction

- **I2C Start**

The default state of SDA and SCLK is high. Every transaction on I2C bus starts by initiating a start condition. The IP is first initialised by setting up `mode` of operation, `datain` register. The `start` signal is pulsed afterwards. It marks the start of an I2C Transaction. The byte sent is Slave Address + ‘0’/’1’ bit in the order of MSB to LSB (‘0’ for write and ‘1’ for read). The `ack` flag is set or reset to signify whether the slave acknowledged or not, followed by setting of `intr` flag.

- **I2C Write**

A write operation can be initiated by setting up `datain` and pulsing `wr` signal. The byte is sent on the bus in MSB to LSB order. The `ack` flag is set or reset to signify whether the slave acknowledged or not, followed by setting of `intr` flag.

- **I2C Repeated Start**

Repeated start is initiated whenever read or multiple write or multiple reads or mixed write-read transactions are required. It makes sure that the Master doesn’t lose hold of the bus. It is initiated by setting up `datain` register and pulsing `start` signal while an ongoing transaction. The byte sent is Slave Address + ‘0’/’1’. The `ack` flag is set or reset to signify whether the slave acknowledged or not, followed by setting of `intr` flag.

- **I2C Read**

A read operation can be initiated by setting up `enack` register and then pulsing `rd` signal. After reading a byte from the bus, `intr` flag is set. The byte read out from the bus is available in `dataout` register. The enack bit is significant, as it decides whether Master acknowledge or not. Read operation continues until Master sends a NACK (No Acknowledge) after a byte read operation. Master regains the control on SDA line after NACK, and can initiate a stop condition or repeated start condition afterwards.

- **I2C Stop**

Stop condition can be initiated after a byte transfer by pulsing `stop` signal. The `intr` flag is set afterwards the signifies that the I2C Transaction has been stopped.
Summary of I2C Operations

Write operation:

1. Initialize I2C.
2. Generate START condition.
3. Send Slave device write address (SLA+W) and check for acknowledgement.
4. Write memory location address for memory devices to which we want to write.
5. Write data till last byte.
6. Generate STOP condition.

Read operation:

1. Initialize I2C.
2. Generate START condition.
3. Write device Write address (SLA+W) and check for acknowledgement.
4. Write memory location address for memory devices.
5. Generate REPEATED START condition with (SLA +R).
6. Read data and return acknowledgement.
7. Return Not acknowledgement for last byte.
8. Generate STOP condition.
Handshaking Protocol

Handshaking between interrupt and other control signals is discussed in this section. The handshaking protocol is shown in figures:

```
INTR ____________/
WR ______________/
RD ______________/
START ____________/
STOP ____________/
```

Not pulsing control signals after interrupt, stretches SCLK and the bus remains idle.
Notes

Pull-ups are required on SDA line, but not required on SCLK line. The mode of operation decides the value of pull-up. A generic formula for calculating pull-up resistor values:

\[
\text{Freq} < 100\text{kHz} \quad \implies \quad R_{\text{min}} = \frac{V_{cc} - 0.4V}{3\text{mA}}, \quad R_{\text{max}} = \frac{1000\text{ns}}{C_{\text{bus}}} \\
\text{Freq} > 100\text{kHz} \quad \implies \quad R_{\text{min}} = \frac{V_{cc} - 0.4V}{3\text{mA}}, \quad R_{\text{max}} = \frac{300\text{ns}}{C_{\text{bus}}}
\]

Schmitt triggers are recommended on both lines for clean signal transitions.
Notice
I2C Master IP Core v.1.1 © 2019
Open-source licensed
Developer: Mitu Raj, iammituraj@gmail.com