Development and research of different architectures of I²C bus controller

E. Vasiliev, MIET
# I²C and its alternatives

I²C (Inter-Integrated Circuit) is a multi-master serial computer bus invented by Philips that is used to attach low-speed peripherals to a motherboard, embedded system, or cellphone.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Advantages over I²C</th>
<th>Disadvantages over I²C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI</td>
<td>Synchronous serial connection, 3 or 1 line for transmission, data is sent synchronized by the clock, transmission is based on push/pull technology</td>
<td>Higher speed</td>
<td>No addressing (many devices on one bus are not allowed)</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver/Transmitter. Fixed baudrate is used for transmission.</td>
<td>Ability to serve as master and slave at the same time. Physical layer can be used, allowing to bridge larger distances</td>
<td>No addressing (many devices on one bus are not allowed)</td>
</tr>
<tr>
<td>CAN</td>
<td>Complex protocol of CAN allows for data integrity check, device addressing, error recovery and several advanced features</td>
<td>Physical layer can be used, allowing to bridge larger distances</td>
<td>Complexity</td>
</tr>
<tr>
<td>1-Wire®</td>
<td>Just one wire plus ground are used (i.e. two wires). It is even possible to supply power to connected components over these two wires.</td>
<td>Physical layer can be used, allowing to bridge larger distances</td>
<td>Strict time keeping on both, master and slave side, lower speed</td>
</tr>
</tbody>
</table>
Designer benefits

- Functional blocks on the block diagram correspond with the actual ICs; designs proceed rapidly from block diagram to final schematic.
- No need to design bus interfaces because the I2C-bus interface is already integrated on-chip.
- Integrated addressing and data-transfer protocol allow systems to be completely software-defined.
- The same IC types can often be used in many different applications.
- Design-time reduces as designers quickly become familiar with the frequently used functional blocks represented by I2C-bus compatible IC.
- ICs can be added to or removed from a system without affecting any other circuits on the bus.
- Fault diagnosis and debugging are simple; malfunctions can be immediately traced.
- Software development time can be reduced by assembling a library of reusable software modules.
Manufacturer benefits

- The simple 2-wire serial I2C-bus minimizes interconnections so ICs have fewer pins and there are not so many PCB tracks; result — smaller and less expensive PCBs.
- The completely integrated I2C-bus protocol eliminates the need for address decoders and other ‘glue logic’.
- The multi-master capability of the I2C-bus allows rapid testing and alignment of end-user equipment via external connections to an assembly-line.
Example of I²C bus applications
Example of an I2C-bus configuration using two microcontrollers
## Applicability of I2C-bus protocol features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single master</td>
</tr>
<tr>
<td>START condition</td>
<td>M</td>
</tr>
<tr>
<td>STOP condition</td>
<td>M</td>
</tr>
<tr>
<td>Acknowledge</td>
<td>M</td>
</tr>
<tr>
<td>Synchronization</td>
<td>n/a</td>
</tr>
<tr>
<td>Arbitration</td>
<td>n/a</td>
</tr>
<tr>
<td>Clock stretching</td>
<td>O</td>
</tr>
<tr>
<td>7-bit slave address</td>
<td>M</td>
</tr>
<tr>
<td>10-bit slave address</td>
<td>O</td>
</tr>
<tr>
<td>General Call address</td>
<td>O</td>
</tr>
<tr>
<td>Software Reset</td>
<td>O</td>
</tr>
<tr>
<td>START byte</td>
<td>n/a</td>
</tr>
<tr>
<td>Device ID</td>
<td>n/a</td>
</tr>
</tbody>
</table>

*M = mandatory; O = optional; n/a = not applicable.*
Devices with a variety of supply voltages sharing the same bus

$V_{DD1} = 5\,\text{V} \pm 10\%$

$V_{DD2}$ and $V_{DD3}$ are device dependent (e.g., 12 V).
Bit transfer on the I2C-bus

SDA

SCL

data line stable; data valid
change of data allowed
START and STOP conditions
Data transfer on the I2C-bus

- **SDA**
  - MSB
  - acknowledgement signal from slave

- **SCL**
  - 1 2 7 8 9 1 2 3 to 8 9
  - ACK
  - repeated START byte
  - complete, clock line held LOW
  - condition
  - interrupt within slave
  - while interrupts are serviced
  - START or STOP
  - or
  - Sr or P

- **Sr or P**
  - STOP or repeated START condition
Clock synchronization during the arbitration procedure

- CLK 1
- CLK 2
- SCL

- Wait state
- Counter reset
- Start counting HIGH period
Arbitration procedure of two masters

master 1 loses arbitration
DATA 1 ≠ SDA
A complete data transfer
A master-transmitter addressing a slave receiver with a 7-bit address (the transfer direction is not changed)

<table>
<thead>
<tr>
<th>S</th>
<th>SLAVE ADDRESS</th>
<th>R/W</th>
<th>A</th>
<th>DATA</th>
<th>A</th>
<th>DATA</th>
<th>A/Ā</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>'0' (write) data transferred (n bytes + acknowledge)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- from master to slave
  - A = acknowledge (SDA LOW)
  - Ā = not acknowledge (SDA HIGH)
- from slave to master
  - S = START condition
  - P = STOP condition
A master reads a slave immediately after the first byte

---

<table>
<thead>
<tr>
<th>S</th>
<th>SLAVE ADDRESS</th>
<th>R/W</th>
<th>A</th>
<th>DATA</th>
<th>A</th>
<th>DATA</th>
<th>A</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(read) (n bytes + acknowledge)
**Combined format**

<table>
<thead>
<tr>
<th>S</th>
<th>SLAVE ADDRESS</th>
<th>R/W</th>
<th>A</th>
<th>DATA</th>
<th>A/Ã</th>
<th>Sr</th>
<th>SLAVE ADDRESS</th>
<th>R/W</th>
<th>A</th>
<th>DATA</th>
<th>A/Ã</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- read or write
- (n bytes + ack.)*
- read or write

*not shaded because transfer direction of data and acknowledge bits depends on R/W bits.

Sr = repeated START condition

Direction of transfer may change at this point.
Bus speeds

- **Standard-mode (Sm)**, with a bit rate up to 100 kbit/s
- **Fast-mode (Fm)**, with a bit rate up to 400 kbit/s
- **Fast-mode Plus (Fm+)**, with a bit rate up to 1 Mbit/s
- **High-speed mode (Hs-mode)**, with a bit rate up to 3.4 Mbit/s.
I2C applications

I²C is appropriate for peripherals where *simplicity* and *low manufacturing cost* are more important than *speed*. Common applications of the I²C bus are:

- Reading configuration data from SPD EEPROMs on SDRAM, DDR SDRAM, DDR2 SDRAM memory sticks (DIMM) and other stacked PC boards.
- Supporting systems management for PCI cards, through an SMBus 2.0 connection.
- Accessing NVRAM chips that keep user settings.
- Accessing low speed DACs and ADCs.
- Changing contrast, hue, and color balance settings in monitors (Display Data Channel).
- Changing sound volume in intelligent speakers.
- Controlling OLED/LCD displays, like in a cellphone.
- Reading hardware monitors and diagnostic sensors, like a CPU thermostat and fan speed.
- Reading real time clocks.
- Turning on and turning off the power supply of system components.
Architectures of I²C bus controller

- State machine architecture
- Demultiplexer/Multiplexer chains architecture
- Two shift registers architecture
Mealy state machine for slave I\textsuperscript{2}C bus controller

- Waiting for START
- Receiving address, \( R/W \) bit
- Transmitting acknowledge
- Transmitting data bit
- Receiving data bit
- Transmitting acknowledge
- Receiving acknowledge (not acknowledge) bit

*BC - Bit Counter (reverse)
Demultiplexer/Multiplexer chains architecture of slave I2C bus controller

D1 - data, received from bus
D2 - data, transmitted on bus
A – slave address
A.L. – additional logic

D - demux
D-tr. – control D-triggers
M – mux
C – comparator

address/data signal

R/W & addr/data signal

SCL \_/-

from SDA bus

to SDA bus

1
Two shift registers architecture of slave I²C bus controller

Comparing with address, transmitting on output

Data from input


A.L.


A.L.

from SDA bus 0

R/\overline{W} & addr./data signal

to SDA bus

neg. edge of SCL
Symbol of slave I2C bus controller
Verification of slave I2C bus controller

M to S tests (one addressing, permanent addressing, addressing and data transmission with NACK, transmission of different numbers of bytes)

S to M tests (one addressing, permanent addressing, transmission of different numbers of bytes)

Arbitration tests (with permanent addressing)
Comparative characteristics of I2C bus architectures

<table>
<thead>
<tr>
<th>Architecture name</th>
<th>Area of non-combination elements (triggers), μm</th>
<th>Area of combinational elements, μm</th>
<th>Number of cells</th>
<th>Whole-time addressing</th>
<th>Number of nets</th>
</tr>
</thead>
<tbody>
<tr>
<td>State machine architecture</td>
<td>2031,6</td>
<td>3108,9</td>
<td>194</td>
<td>No</td>
<td>216</td>
</tr>
<tr>
<td>Multiplexer/demultiplexer chains architecture</td>
<td>2056,2</td>
<td>1990,7</td>
<td>144</td>
<td>No</td>
<td>168</td>
</tr>
<tr>
<td>Two shift registers architecture</td>
<td>2011,1</td>
<td>1445,9</td>
<td>102</td>
<td>Yes</td>
<td>127</td>
</tr>
</tbody>
</table>
Area of architectures (data are based on cell areas)

- State machine architecture
- Demultiplexer/multiplexer chains architecture
- Two shift registers architecture

Device area, square μm
Power of architectures (data are based on cell powers)

Consumed power, $\mu$W

- State machine architecture
- Multiplexer/demultiplexer chains architecture
- Two shift registers architecture