Serial Communications
*CSEE W4840*
Prof. Stephen A. Edwards
Columbia University

**Early Serial Communication**

Defined in early 1960s
Serial, Asynchronous, Full-duplex,
Voltage-based, point-to-point, 100 ft+ cables
+12V  SPACE = 0
+3V
−3V  MARK = 1
−12V

<table>
<thead>
<tr>
<th>Start</th>
<th>LSB</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>B5</th>
<th>B6</th>
<th>MSB</th>
<th>Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RS-232 Signals**

Signal DB-9 DTE ... Meaning

- **RxD** 2 → Data received by DTE
- **TxD** 3 → Data sent by DTE
- **SG** 5 → Ground
- **DSR** 6 → Data Set Ready (I’m alive)
- **DTR** 4 → Data Terminal Ready (me, too)
- **DCD** 1 → Carrier Detect (hear a carrier)
- **RTS** 7 → Request To Send (Yo?)
- **CTS** 8 → Clear To Send (Yo!)
- **RI** 9 → Ring Indicator

Most UARTs actually use 16× clocks

**Variants**

Parity bit: (Even = true when even number of 1s)

<table>
<thead>
<tr>
<th>Start</th>
<th>LSB</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>B5</th>
<th>B6</th>
<th>Parity</th>
<th>Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Two stop bits:

<table>
<thead>
<tr>
<th>Start</th>
<th>LSB</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>B5</th>
<th>B6</th>
<th>MSB</th>
<th>Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tx</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Baud Rate**

Baud: bits per second

<table>
<thead>
<tr>
<th>Baud</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>ASR-33 Teletype</td>
</tr>
<tr>
<td>300</td>
<td>Early acoustic modems</td>
</tr>
<tr>
<td>1200</td>
<td>Direct-coupled modems c. 1980</td>
</tr>
<tr>
<td>2400</td>
<td>Modems c. 1990</td>
</tr>
<tr>
<td>9600</td>
<td>Serial terminals</td>
</tr>
<tr>
<td>19200</td>
<td></td>
</tr>
<tr>
<td>38400</td>
<td>Typical maximum</td>
</tr>
</tbody>
</table>

**Physical Variants**

Connectors: DB-25, DB-9, Mini DIN-8
RS-422: Differential signaling
RS-485: Bus-like
OPB UART Lite

Serial port peripheral for the Microblaze
Full duplex operation
16-character transmit and receive FIFOs
Parameters that can be set at build time:

Parameter | Value
---|---
Base Address | 0xFEFF0100
High Address | 0xFEFF01FF
Baud Rate | 9600
Bits per frame | 8
Parity | None

OPB UART Lite Registers

<table>
<thead>
<tr>
<th>Address</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xFEFF0100</td>
<td>Read characters from Receive FIFO</td>
</tr>
<tr>
<td>0xFEFF0104</td>
<td>Write characters to Receive FIFO</td>
</tr>
<tr>
<td>0xFEFF0108</td>
<td>Status register (read only)</td>
</tr>
<tr>
<td>0xFEFF010C</td>
<td>Control register (write only)</td>
</tr>
</tbody>
</table>

Status and Control Registers

<table>
<thead>
<tr>
<th>Bit</th>
<th>Status</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>Parity Error</td>
<td>-</td>
</tr>
<tr>
<td>25</td>
<td>Framing Error</td>
<td>-</td>
</tr>
<tr>
<td>26</td>
<td>Overrun Error</td>
<td>-</td>
</tr>
<tr>
<td>27</td>
<td>Interrupts Enabled</td>
<td>Enable Interrupts</td>
</tr>
<tr>
<td>28</td>
<td>Tx buffer full</td>
<td>-</td>
</tr>
<tr>
<td>29</td>
<td>Tx buffer empty</td>
<td>-</td>
</tr>
<tr>
<td>30</td>
<td>Rx buffer full</td>
<td>Clear Rx buffer</td>
</tr>
<tr>
<td>31</td>
<td>Rx buffer non-empty</td>
<td>Clear Tx buffer</td>
</tr>
</tbody>
</table>

Non-empty Rx buffer or emptying of Tx buffer generates an interrupt.

The I²C Bus

Philips invented the Inter-IC bus c. 1980 as a very cheap way to communicate slowly among chips
E.g., good for setting control registers
100, 400, and 3400 kHz bitrates

I²C Bus Transaction

USB: Universal Serial Bus

1.5 Mbps, 12 Mbps, and 480 Mbps (USB 2.0)
Point-to-point, differential, twisted pair
3–5m maximum cable length

USB Connectors

USB signaling

NRZI: 0 = toggle, 1 = no change
Bit stuffing: 0 automatically inserted after six consecutive 1s

USB Packets

Always start with SYNC
Then 4-bit type, 4-bit type complemented
2 bits distinguish Token, Data, Handshake, and Special, other two bits select sub-types
Then data, depending on packet type
Data checked using a CRC
Addresses (1-128) assigned by bus master, each with 16 possible endpoints
USB: Flash Card Device

- Bus 001 Device 002: ID 05E3:0760 Genesys Logic, Inc.
- bUSB: 2.0
- bMaxPacketSize: 32
- bInterfaceClass: 8 Mass Storage
- bInterfaceSubClass: 0 IDE
- bInterfaceProtocol: 0 Bulk (Zip)
- bInterfaceNumber: 1 Mass Storage Interface
- bNumEndpoints: 1 Endpoint 1 IN
- bMaxPacketSize: 64
- bInterval: 2
- wMaxPacketSize: 64

USB: Mouse Device

- Bus 001 Device 002: ID 04B4:0001 Cypress Semiconductor Mouse
- bUSB: 2.0
- bMaxPacketSize: 32
- bInterfaceClass: 8 Human Interface Devices
- bInterfaceSubClass: 1 Boot Interface Subclass
- bInterfaceProtocol: 2 Mouse
- bInterfaceNumber: 1 Mouse Interface
- bNumEndpoints: 3 Endpoints
- bMaxPacketSize: 64
- bInterval: 2
- wMaxPacketSize: 64

The CY7C68001 USB Interface

- Operates as a peripheral (i.e., not a host)
- Operates at 12 or 480 Mbps speeds
- Control endpoint 0
- Four other user-configurable endpoints
- 4 kB FIFO buffer
- 500 bytes of descriptor RAM (Vendor, Product)
- I2C bus interface for configuration from EEPROM (Unused on the XSB board—processor must configure)

The CY7C68001 software interface

- Five memory locations: one for each FIFO, one for control registers
- Internal registers written by first applying address to control register, then reading or writing data to control register.
- 33 different configuration registers, including 500-byte descriptor “register”