Serial Communications
Prof. Stephen A. Edwards
sedwards@cs.columbia.edu
Columbia University
Spring 2006

Early Serial Communication

Later Serial Communication

RS-232
Defined in early 1960s
Serial, Asynchronous, Full-duplex,
Voltage-based, point-to-point, 100 ft+ cables
+12V
SPACE = 0
+3V
−3V
MARK = 1
−12V

RS-232 Signals
Signal DB-9 DTE ... Meaning
pin DCE
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

Receiving RS-232
Most UARTs actually use 16× clocks

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

Baud Rate
Baud: bits per second

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Address</td>
<td>0xFEFF0100</td>
</tr>
<tr>
<td>High Address</td>
<td>0xFEFF01FF</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>9600</td>
</tr>
<tr>
<td>Bits per frame</td>
<td>8</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
</tbody>
</table>

Parameters that can be set at build time:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Address</td>
<td>0xFEFF0100</td>
</tr>
<tr>
<td>High Address</td>
<td>0xFEFF01FF</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>9600</td>
</tr>
<tr>
<td>Bits per frame</td>
<td>8</td>
</tr>
<tr>
<td>Parity</td>
<td>None</td>
</tr>
</tbody>
</table>

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

SCL: Clock, generated by a single master
SDA: Data, controlled by either master or slaves

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

<table>
<thead>
<tr>
<th>Series &quot;A&quot; Connectors</th>
<th>Series &quot;B&quot; Connectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Series &quot;A&quot; plugs are always oriented upstream towards the Host System</td>
<td>Series &quot;B&quot; plugs are always oriented downstream towards the USB Device</td>
</tr>
<tr>
<td>&quot;A&quot; Plugs (From the USB Device)</td>
<td>&quot;B&quot; Plugs (From the USB Device)</td>
</tr>
<tr>
<td>&quot;A&quot; Receptors (Downstream Connectors from the USB Host or Hub)</td>
<td>&quot;B&quot; Receptors (Upstream Input to the USB Device or Hub)</td>
</tr>
</tbody>
</table>

USB signaling

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

<table>
<thead>
<tr>
<th>Data</th>
<th>Idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0110</td>
<td>0010</td>
</tr>
<tr>
<td>0100</td>
<td>0010</td>
</tr>
<tr>
<td>1110</td>
<td>0010</td>
</tr>
<tr>
<td>0111</td>
<td>0010</td>
</tr>
</tbody>
</table>

Each packet prefixed by a SYNC field: 3 0s followed by two 1s
Low- vs. full-speed devices identified by different pull-ups on D+/D- lines

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 Bus Protocol

Polled bus: host initiates all transfers.
Most transactions involve three packets:
- “Token” packet from host requesting data
- Data packet from target
- Acknowledge from host

Supports both streams of bytes and structured messages (e.g., control changes).

USB Data Flow Types

- Control
  For configuration, etc.
- Bulk Data
  Arbitrary data stream: bursty
- Interrupt Data
  Timely, reliable delivery of data. Usually events.
- Isochronous Data
  For streaming real-time transfer: prenegotiated bandwidth and latency

Layered Architecture

- Host
- Interconnect
- Physical Device
- Function Layer
- USB Device Layer
- USB Bus Interface Layer
- Actual communications flow
- Logical communications flow
- Implementation Focus Area

USB: Flash Card Device

Bus 001 Device 002: ID 05e3:0760 Genesys Logic, Inc.
  bMaxUSB 2.00
  bMaxPacketSize 64
  bDeviceClass 8 Mass Storage
  bDeviceSubClass 6 SCSI
  bDeviceProtocol 120 Bulk (Zip)
  iManufacturer 2 Genesys
  iProduct 3 Flash Reader
  iSerial 2 Genesys
  Configuration Descriptor:
    bNumInterfaces 1
    MaxPower 300mA
    Interface Descriptor:
      bNumEndpoints 2
      bInterfaceClass 8 Mass Storage
      bInterfaceSubClass 6 SCSI
      bInterfaceProtocol 120 Bulk (Zip)
      Endpoint Descriptor:
        bEndpointAddress 0x01 EP 1 IN
        bMaxPacketSize 64
        bAttributes 2
        Transfer Type Bulk
        Synch Type none
        Language ID: (length=4)
        0409 English(US)

USB: Mouse Device

Bus 001 Device 002: ID 04b4:0001 Cypress Semiconductor Mouse
  Device Descriptor:
    bNumUSB 1.00
    bMaxUSB 1.00
    bMaxPacketSize 64
    bDeviceClass 3 Human Interface Devices
    bDeviceSubClass 1 Boot Interface Subclass
    bDeviceProtocol 1 Mouse
    iManufacturer 2 Adomax Sem.
    iProduct 1 USB Mouse
    iSerial 4 003064
    Configuration Descriptor:
      bNumInterfaces 1
      bMaxPower 100mA
      Interface Description:
        bNumEndpoints 2
        bInterfaceClass 1 Human Interface Devices
        bInterfaceSubClass 1 Boot Interface Subclass
        bInterfaceProtocol 1 Mouse
        Endpoint Descriptor:
          bEndpointAddress 0x81 EP 1 IN
          bMaxPacketSize 64
          bAttributes 2
          Transfer Type Bulk
          Synch Type none
          Language ID: (length=4)
          0409 English(US)
        Endpoint Descriptor:
          bEndpointAddress 0x02 EP 2 OUT
          bMaxPacketSize 64
          bAttributes 2
          Transfer Type Bulk
          Synch Type none
          Language ID: (length=4)
          0409 English(US)

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)

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”