



Serial Communications

CSEE W4840

Prof. Stephen A. Edwards

Columbia University

Early Serial Communication

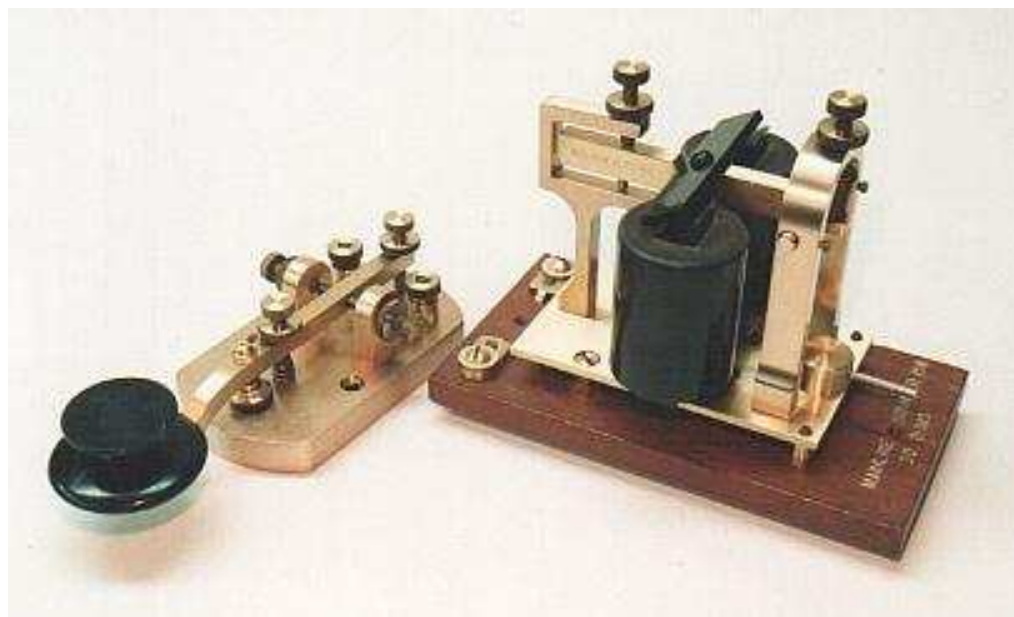
Morse code key

Letters

A	•—
B	—•••
C	—•—•
D	—••
E	•
F	••—•
G	——•
H	••••
I	••
J	•— — —
K	—•—
L	•—••
M	— —
N	—•
O	— — —
P	•— — •
Q	— — • —
R	•—•
S	•••
T	—
U	••—
V	•••—
W	•— —
X	—••—
Y	—•— —
Z	— — ••

Numbers

1	• — — — —
2	•• — — —
3	••• — —
4	•••• —
5	•••••
6	—••••
7	— —•••
8	— — —••
9	— — — —•
0	— — — — —



Later Serial Communication



Data Terminal Equipment



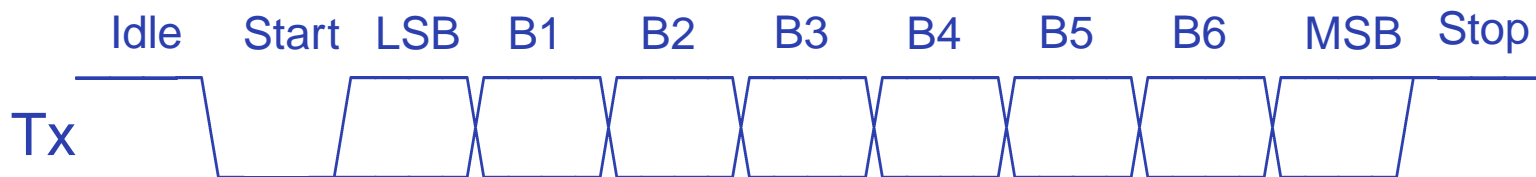
Data
Communications
Equipment

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)



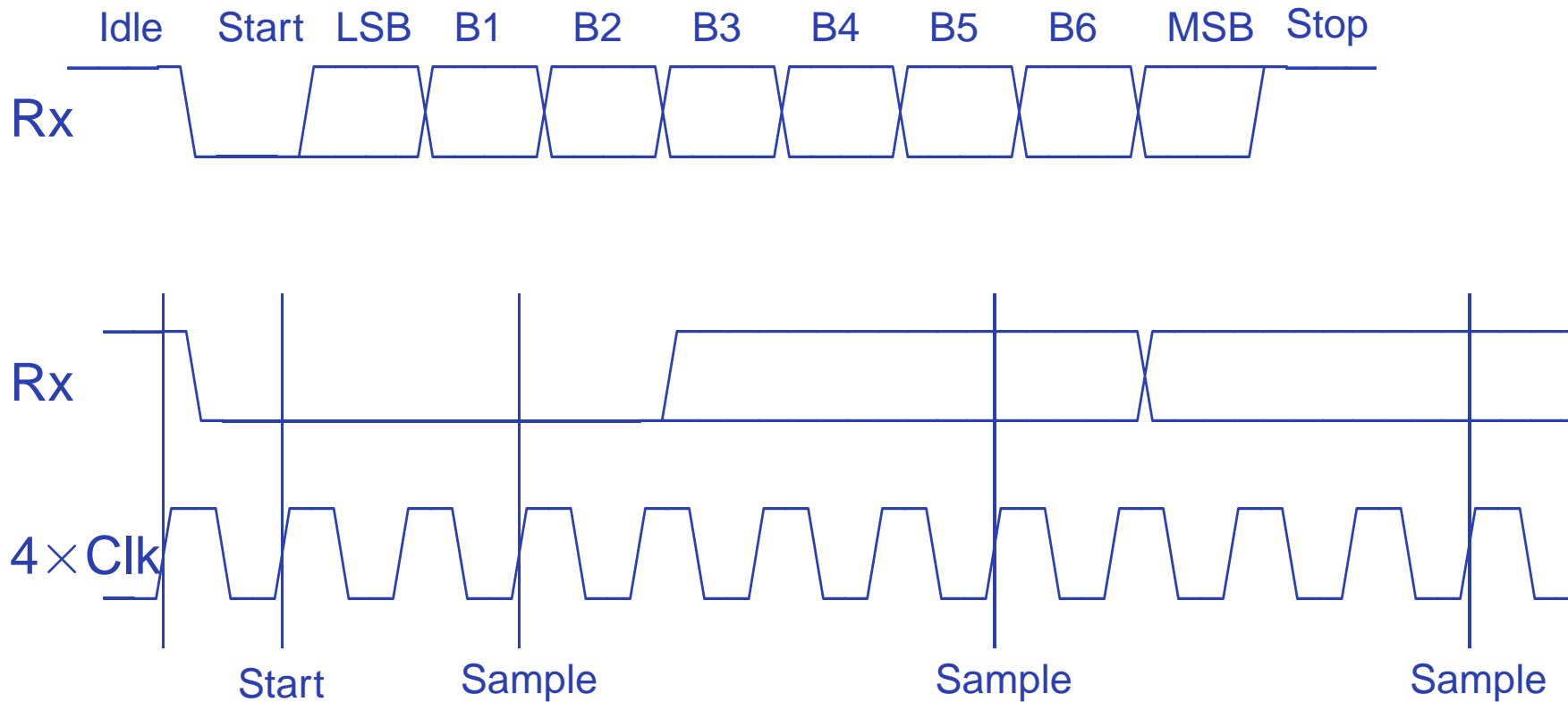
RS-232 Signals



Signal DB-9 DTE ... Meaning

	pin	DCE	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

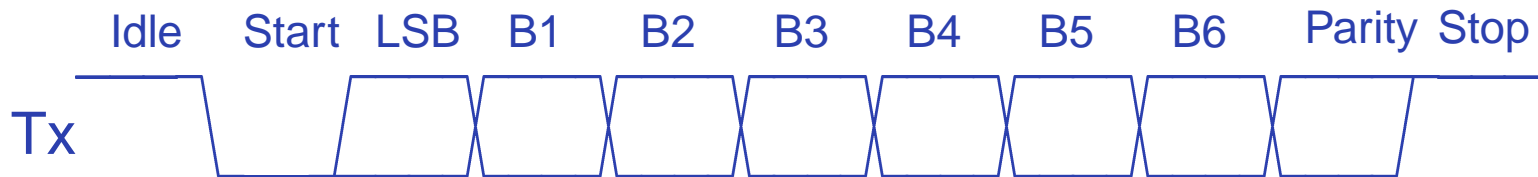
Receiving RS-232



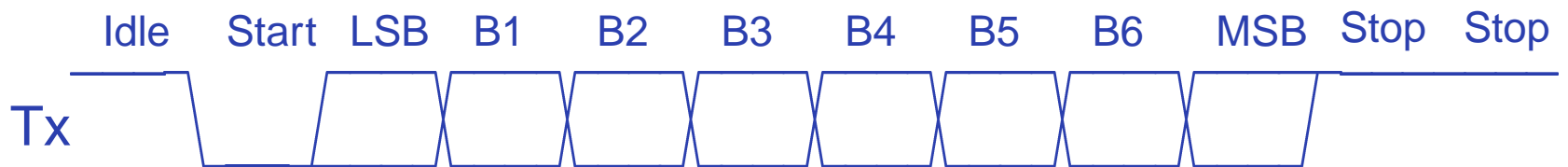
Most UARTs actually use $16 \times$ clocks

Variants

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



Two stop bits:



Baud Rate

Baud: bits per second

Baud	Application
-------------	--------------------

110	ASR-33 Teletype
-----	-----------------

300	Early acoustic modems
-----	-----------------------

1200	Direct-coupled modems c. 1980
------	-------------------------------

2400	Modems c. 1990
------	----------------

9600	Serial terminals
------	------------------

19200	
-------	--

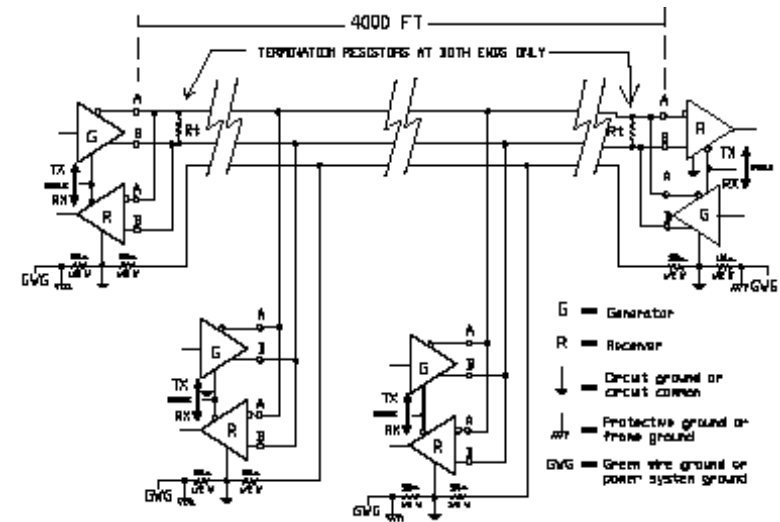
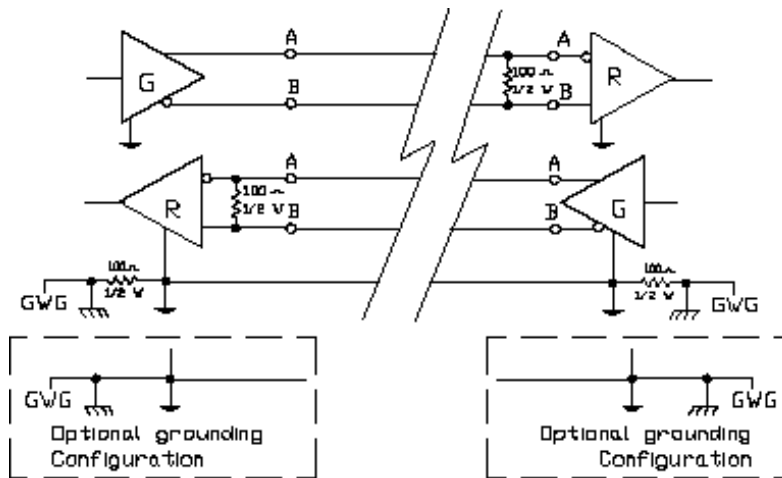
38400	Typical maximum
-------	-----------------

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

Address	Role
0xFEFF0100	Read characters from Receive FIFO
0xFEFF0104	Write characters to Receive FIFO
0xFEFF0108	Status register (read only)
0xFEFF010C	Control register (write only)

Status and Control Registers

Bit	Status	Control
24	Parity Error	-
25	Framing Error	-
26	Overrun Error	-
27	Interrupts Enabled	Enable Interrupts
28	Tx buffer full	-
29	Tx buffer empty	-
30	Rx buffer full	Clear Rx buffer
31	Rx buffer non-empty	Clear Tx buffer

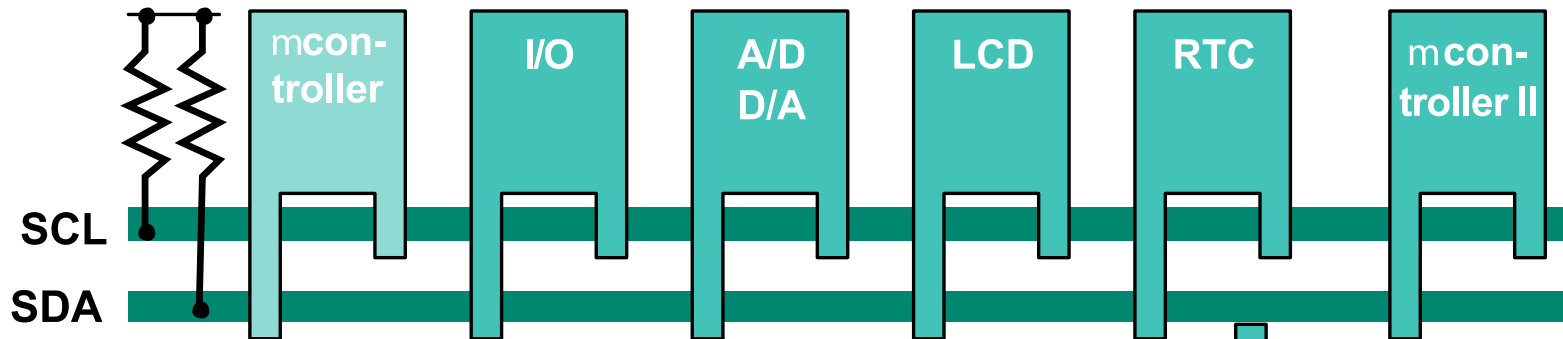
Non-empty Rx buffer or emptying of Tx buffer generates an interrupt.

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

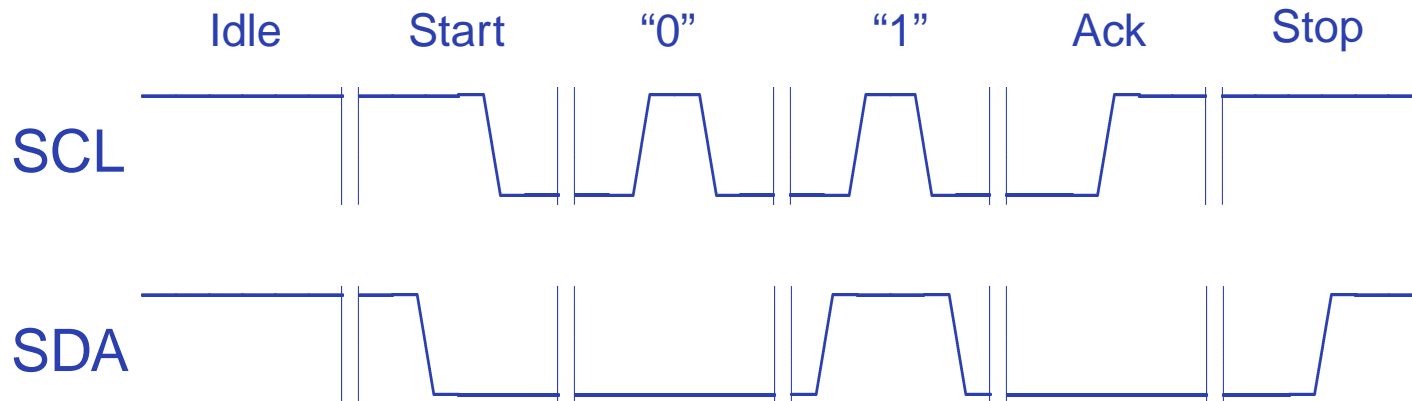
100, 400, and 3400 kHz bitrates



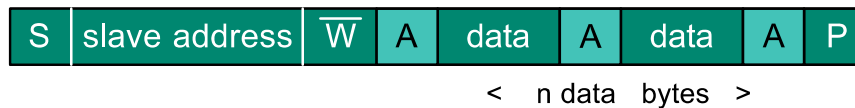
SCL: Clock, generated by a single master

SDA: Data, controlled by either master or slaves

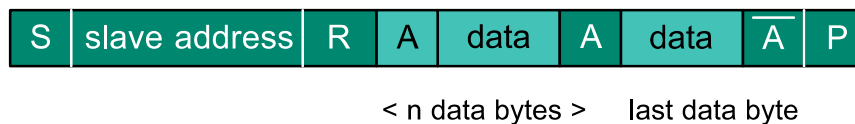
I²C Bus Transaction



Write data



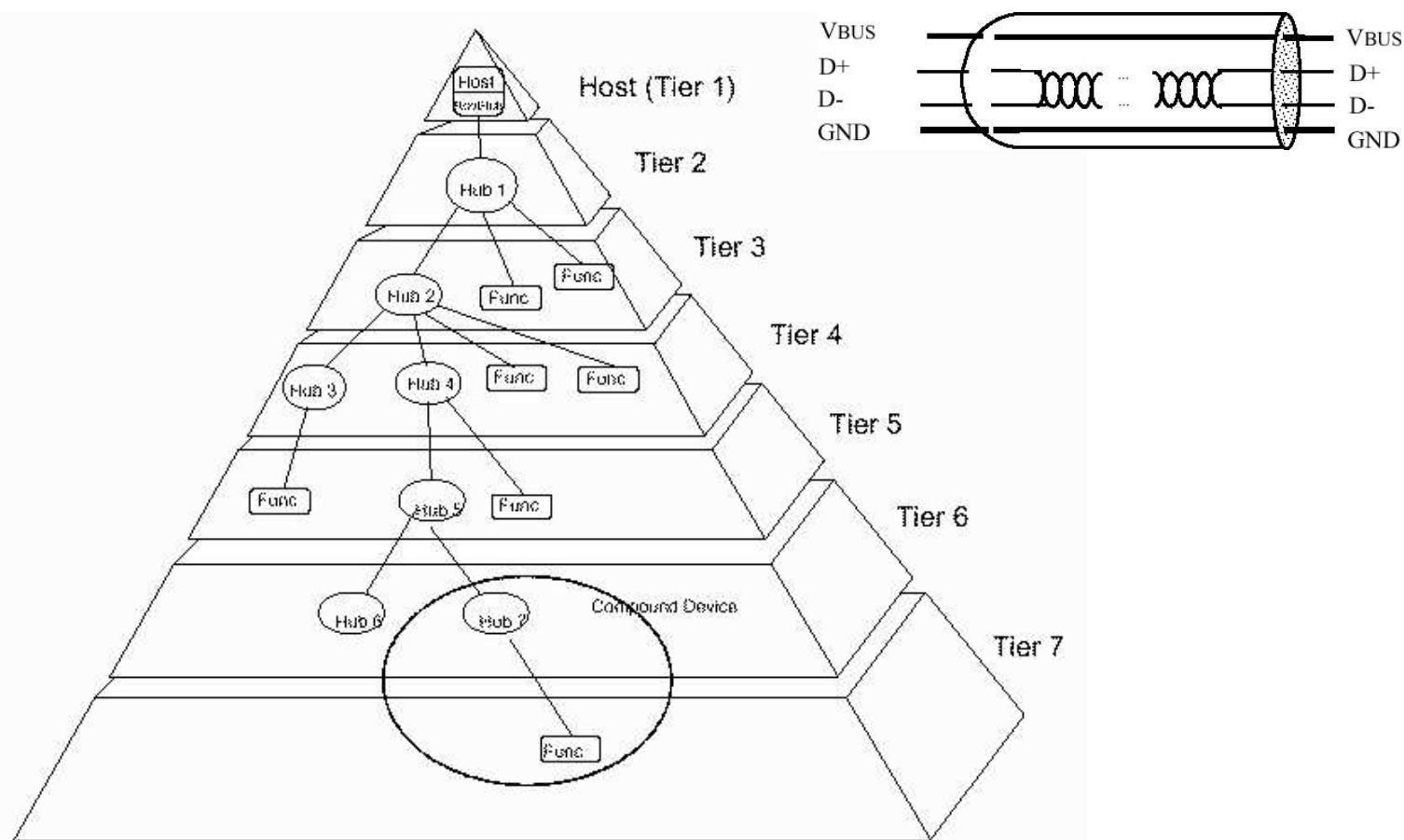
Read data



S = Start condition
 \overline{W} = read / write not
 A = Acknowledge
 P = Stop condition
 \overline{A} = Not Acknowledge

USB: Universal Serial Bus

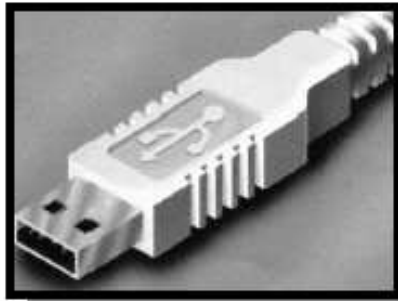
1.5 Mbps, 12 Mbps, and 480 Mbps (USB 2.0)
Point-to-point, differential, twisted pair
3–5m maximum cable length



USB Connectors

Series "A" Connectors

- ◆ Series "A" plugs are always oriented **upstream** towards the *Host System*



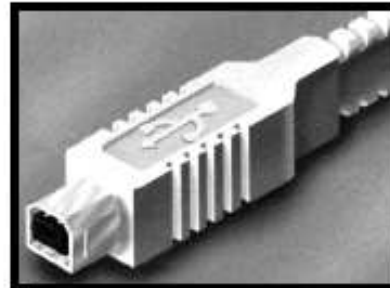
"A" Plugs
(From the
USB Device)

"A" Receptacles
(Downstream Output
from the USB Host or
Hub)



Series "B" Connectors

- ◆ Series "B" plugs are always oriented **downstream** towards the *USB Device*



"B" Plugs
(From the
Host System)

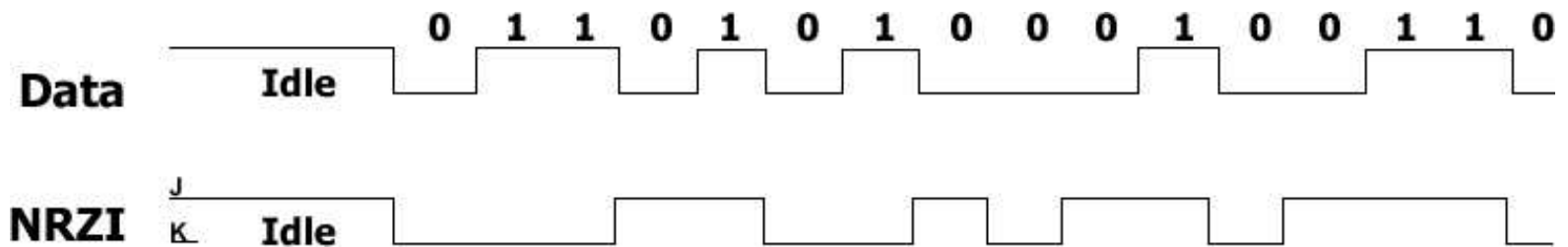
"B" Receptacles
(Upstream Input to the
USB Device or Hub)



USB signaling

NRZI: 0 = toggle, 1 = no change

Bit stuffing: 0 automatically inserted after six consecutive 1s



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:

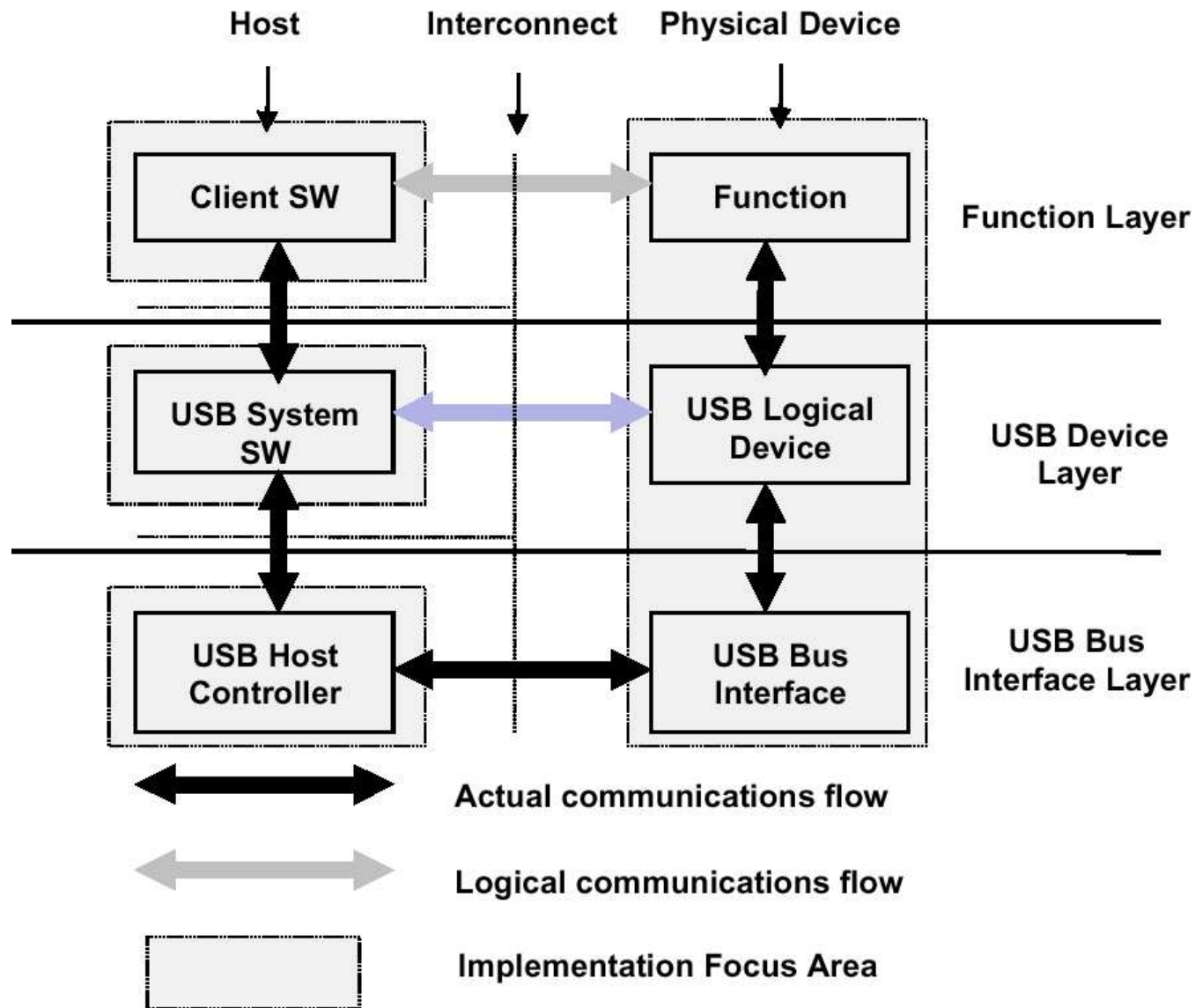
- “Token” packet from host requesting data
- Data packet from target
- Acknowledge from host

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



USB: Flash Card Device

```
Bus 001 Device 002: ID 05e3:0760 Genesys Logic, Inc.
bcdUSB                2.00
bMaxPacketSize0      64
idVendor              0x05e3 Genesys Logic, Inc.
idProduct             0x0760
bcdDevice             1.14
iManufacturer         2 Genesys
iProduct              3 Flash Reader
iSerial               4 002364
Configuration Descriptor:
  bNumInterfaces       1
  MaxPower             300mA
  Interface Descriptor:
    bNumEndpoints      2
    bInterfaceClass    8 Mass Storage
    bInterfaceSubClass 6 SCSI
    bInterfaceProtocol 80 Bulk (Zip)
    Endpoint Descriptor:
      bEndpointAddress 0x81 EP 1 IN
      bmAttributes     2
        Transfer Type  Bulk
        Synch Type     none
      wMaxPacketSize   64
    Endpoint Descriptor:
      bLength           7
      bDescriptorType   5
      bEndpointAddress 0x02 EP 2 OUT
      bmAttributes     2
        Transfer Type  Bulk
        Synch Type     none
      wMaxPacketSize   64
  Language IDs: (length=4)
    0409 English(US)
```

USB: Mouse Device

Bus 002 Device 002: ID 04b4:0001 Cypress Semiconductor Mouse

Device Descriptor:

```
bcdUSB          1.00
idVendor        0x04b4 Cypress Semiconductor
idProduct       0x0001 Mouse
bcdDevice       4.90
iManufacturer   1 Adomax Sem.
iProduct        2 USB Mouse
iSerial         0
```

Configuration Descriptor:

```
bNumInterfaces  1
bmAttributes    0xa0
  Remote Wakeup
MaxPower        100mA
```

Interface Descriptor:

```
bNumEndpoints  1
bInterfaceClass 3 Human Interface Devices
bInterfaceSubClass 1 Boot Interface Subclass
bInterfaceProtocol 2 Mouse
iInterface      5 EndPoint1 Interrupt Pipe
```

HID Device Descriptor:

```
bDescriptorType 34 Report
wDescriptorLength 52
```

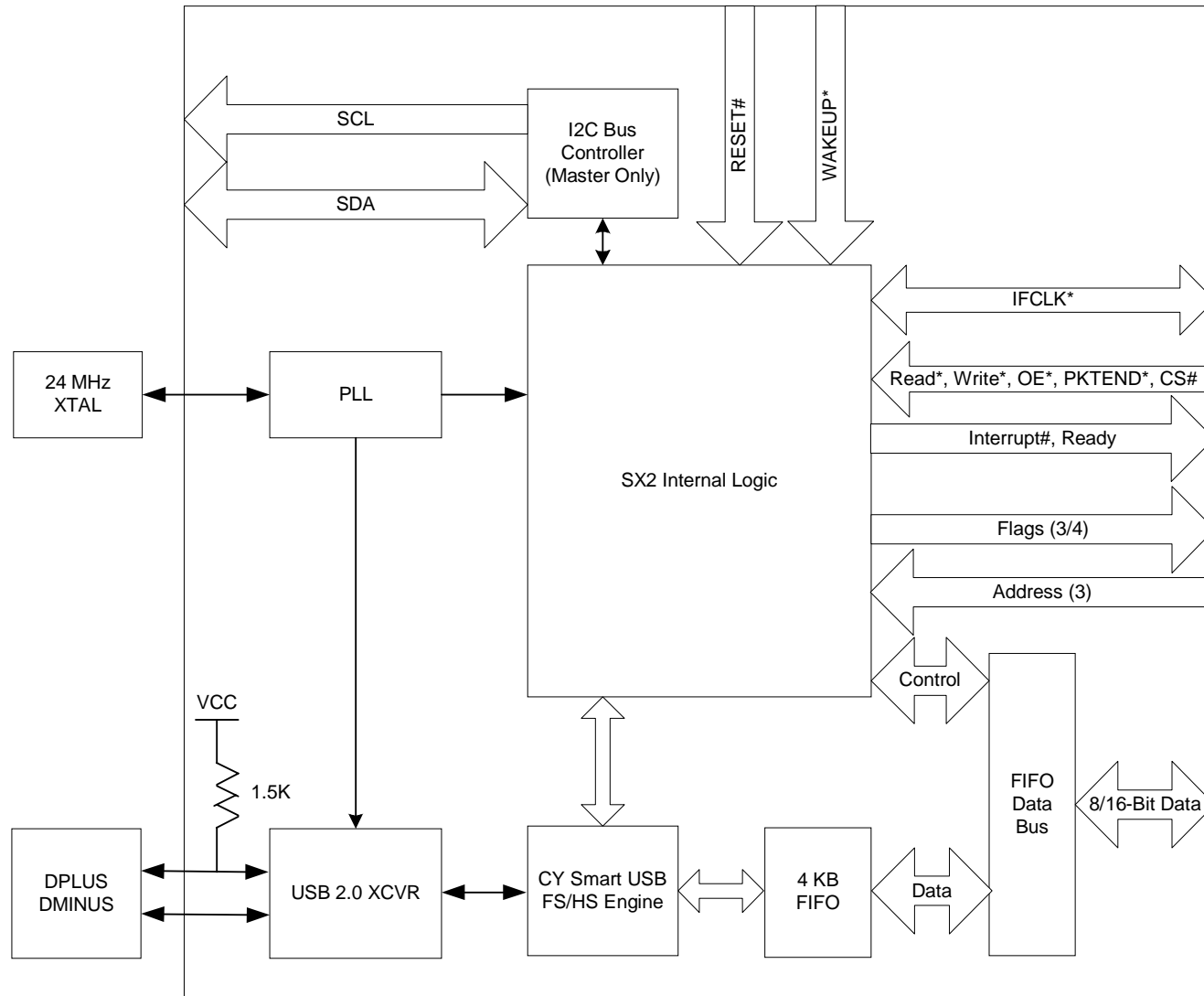
Endpoint Descriptor:

```
bEndpointAddress 0x81 EP 1 IN
bmAttributes      3
  Transfer Type   Interrupt
  Synch Type      none
wMaxPacketSize    4
bInterval         10
```

Language IDs: (length=4)

```
0409 English(US)
```

The CY7C68001 USB interface



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”