lecture 22

Input / Output (I/O) 4

- asynchronous bus, handshaking
- serial bus

Mon. April 4, 2016
"synchronous" bus

= clock based
  (system bus clock is slower than CPU clock)

"asynchronous" bus

= not clock based
  (distance between sender and receiver is too great for accurate timing i.e. physical variability)
Communication between device controllers and devices often is asynchronous (e.g. USB - universal serial bus)
"Handshaking": one method for an asynchronous bus

It can be initiated by source (sender) or by destination (receiver).
sender initiated

e.g. printer controller to printer

data

data ready

receiver

ACK

delay of 'data ready' is needed here because of variability of signal on (physical) wire
Example:

"parallel port" for (pre-USB) printer

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/STROBE</td>
<td>14</td>
<td>/AUTOFEED</td>
</tr>
<tr>
<td>2</td>
<td>DATA 0</td>
<td>15</td>
<td>/ERROR</td>
</tr>
<tr>
<td>3</td>
<td>DATA 1</td>
<td>16</td>
<td>/INIT</td>
</tr>
<tr>
<td>4</td>
<td>DATA 2</td>
<td>17</td>
<td>/SELECT</td>
</tr>
<tr>
<td>5</td>
<td>DATA 3</td>
<td>18</td>
<td>GROUND</td>
</tr>
<tr>
<td>6</td>
<td>DATA 4</td>
<td>19</td>
<td>GROUND</td>
</tr>
<tr>
<td>7</td>
<td>DATA 5</td>
<td>20</td>
<td>GROUND</td>
</tr>
<tr>
<td>8</td>
<td>DATA 6</td>
<td>21</td>
<td>GROUND</td>
</tr>
<tr>
<td>9</td>
<td>DATA 7</td>
<td>22</td>
<td>GROUND</td>
</tr>
<tr>
<td>10</td>
<td>/ACKNOWLEDGE</td>
<td>23</td>
<td>GROUND</td>
</tr>
<tr>
<td>11</td>
<td>BUSY</td>
<td>24</td>
<td>GROUND</td>
</tr>
<tr>
<td>12</td>
<td>PAPER-END</td>
<td>25</td>
<td>GROUND</td>
</tr>
<tr>
<td>13</td>
<td>SELECTED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Set 13 to select a device.

Set 25 to choose a function.
receiver initiated

data request

data

data ready

delay of 'data ready' is needed here because of variability of signal on (physical) wire
Handshaking can work for shared buses too.

However, note that there needs to be a mechanism to ensure that only one device can write to bus at one time.

For example, we saw solutions to this problem already when we discussed the (clocked) system bus, e.g. dedicated BR/BG or IRQ/IACK lines including daisy chaining. We will see another scheme later in the lecture when we discuss USB.
e.g. sender initiated

address (receiver)

data

control

data ready

receiver ACK

Note that address, data, and control may be many bits each.
e.g. receiver initiated

data request
control
address (source)
data
data ready
lecture 22

Input / Output (I/O) 4

- asynchronous bus, handshaking
- serial bus

Mon. April 4, 2016
Parallel bus
- multiple lines
- data needs time to stabilize

Serial
- one line
- data doesn't stabilize (next slide)
Serial bus e.g. USB

- one line only!

- data doesn't stabilize, but rather it travels down the wire like a wave

- sender and receiver both have clocks (but not synchronized and may have very different frequencies)
Sender and receiver must agree on a signal pulse duration (greater than their clock pulse duration (typically GHz)) i.e. lower signal pulse frequency than clock pulse frequency).

e.g. 56,000 bits/sec (baud)

1.5 M bits/sec

100 M bits/sec

500 M bits/sec (USB 2.0)

signal pulse durations are all the same
How to send bits? (simple model -- one byte)

MSB first in this example
Receiver observers a 1 to 0 transition (start), and waits $T/2$. If signal is still 0, then it must be a start bit (not noise).

Receiver samples at a much higher frequency than the pulse frequency.

Receiver samples every $T$ (duration of pulse) until byte is read (samples in middle of pulses).
How to send a file?

Special ASCII characters:

e.g.

- **0x01 SOH**: Start of header
- **0x02 SOT**: Start of text
- **0x03 ETX**: End of text
- **0x04 EOT**: End of transmission

![ASCII Codes](image)
Detecting Errors - parity bit

Add and extra bit to a byte so that the total number of 1's is even (called 'even parity').

The original ASCII was 7 bits with the 8th bit used for parity.

We can detect only if there was an odd number of errors.

Note the difference between detection and correction. Parity bits do not allow us to correct the error(s). We would need to send the message again.

Modern methods of 'error correcting codes' use abstract algebra (theory of finite fields)
UART - universal asynchronous receiver transmitter

- I/O controller and I/O device typically both have one
- contain shift registers
- can do parity checking too
USB - universal serial bus

A: plug into computer (host)
B: plug into device
USB controller ("host")
- polls the USB device ("speak when spoken to")
- interrupts the CPU
USB packets

Fields of a packet include:

- **synch** - used to synchronize the clocks of controller and device
- **PID (packet ID)** -- what type of packet is it? (r/w ? handshake? ..)
- **address** -- which device is it for?
  (each device gets a number when it is plugged in)
- **data**
- **error correction info**
A USB bus can have up to 127 devices on it.

Despite the tree structure, the system behaves as a bus. All devices and host see all signals.
"Plug and play"  (plug and pray)

- USB host senses a new device, so it interrogates it ("who is the new device who just plugged in?")

- USB host then sends info about the device to the OS

- OS loads the driver from disk (or tries to get on www)
  device manufacturers give drivers to Microsoft, Apple
EXERCISE:

Why can't you just rip out your USB stick from your computer? Why do you have to formally 'eject' it?
On Windows 7

Control Panel -> Hardware -> Device Manager -> USB controllers

when I plug in my printer, this one appears
Optical Mouse

Surface (rough at scale of $\frac{1}{10}$ mm)
- image frame is ~20 x 20 pixels
- camera capture is 1500 frames/sec
- image processing software finds best shift between images
Analog video port

I/O controller (graphics card) has an A/D converter
Digital video port

DVI (digital video interface)

HDMI (high definition multimedia interface - carries audio too)

display port
Announcements

A3:  - TA email are on page 1 of A3.pdf
     - grading scheme is posted on public web page

A4:  MIPS register conventions abused in my evaluate function. See discussion board.

- pick up your quizzes

TODO (not on final exam)

Wed: thinking about graduate school?
next Mon: Quiz 6 + more on caches
next Wed: JVM