ECE 471 – Embedded Systems
Lecture 18

Vince Weaver

http://www.eece.maine.edu/∼vweaver

vincent.weaver@maine.edu

6 November 2014

Announcements

- HW6 grades were posted. Let me know if somehow you aren't getting your grades.
- Project ideas! Rememeber to send them by Friday!

One-Wire Bus

- From Dallas Semiconductor
- One wire plus ground (how do you get power?)
- Devices have capacitor to provide power when data line low
- Low speed data and power over one wire (you also need ground)
- One master

- 16.3kbit/s
- Up to 300m twisted pair (phone or ethernet wire)

One-Wire Protocol – Detailed

1. Write 1 – Master pull bus low for 1-15us

2. Write 0 – Master pull bus low for 60-120us

3. Read – Master pull bus low for 15us (checks after another 15us). Slave does nothing if it's a 1. If it's a 0 it pulls the bus low for another 45us.

4. Reset/Prescence – master pulls bus low for 480us. If a device is present it bus pull bus low for 60us starting within 60us after the reset pulse.

Hardware Interface

- Use a GPIO and a pull-up resistor
- Use a serial UART. Needs extra circuitry to hook both TXD and RXD to bus
- USB/i2c/network connected
- Dedicated hardware?

One-Wire Protocol

- Each device has unique 64-bit ID; 8-bits of type, 48 bit ID, 8-bit CRC

- Typically 8-bit command followed by 8-bit data chunks

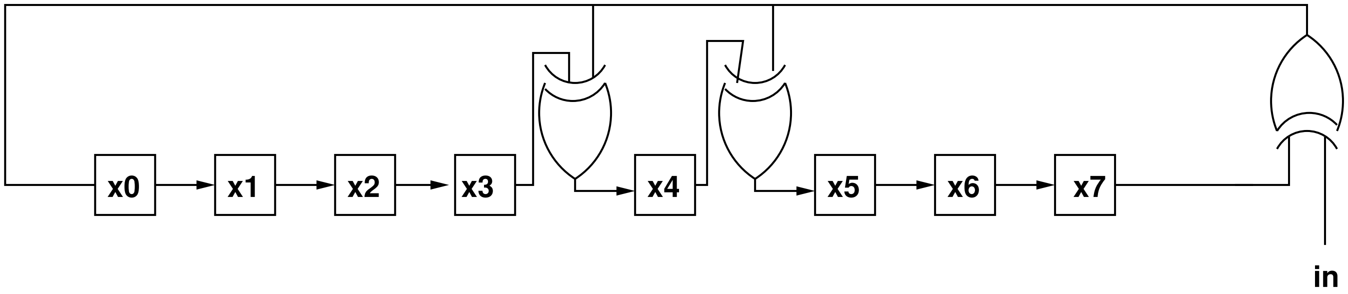
Enumerating BUS (ROM commands)

- send a READ ROM request, returns 64-bit address. If multiple slaves, then and of all of them. (How do you detect this? Invalid CRC).
- SKIP ROM request sends command to all devices
- MATCH ROM request sends 64-bit address and only matched slave responds
- SEARCH ROM – reads first address bit from all devices

on bus (gets and of all) then slaves send complement. So if all start with 1, get 10. If all start with 0, get 01. If both, gets 00. If a conflict, it notes it. Then it sends 0 or 1 indicating path to follow. Then it does a binary search to enumerate all devices on bus. Faster than probing all 2^{64} possible.

CRC check

- Can detect all double-bit errors, any double bit errors, any custer within an 8-bit window
- if CRCs with itself gets 0 at the end, how hardware detects correct address.
- $X^8 + X^5 + X^4 + X^1$
- Fill with zero, shift values in.


x0
x1
x2
x3
x4
x5
x6
x7
in

Linux Interface

- "w1" driver merged in 3.6 kernel (about two years ago)
  - Driver for various interfaces, including bit-banging over GPIO (w1-gpio)
  - `/sys/bus/w1/devices/22-0000001d84f2/w1_slave`
  - read value and get ASCII dump of transaction
- OWFS – another driver, not in main kernel. Lets you export one-wire devices as a filesystem

One-Wire on Raspberry Pi

- by default driver expects to be hooked up to GPIO4.
- `sudo modprobe w1-gpio`
- `sudo modprobe w1-therm`
- `cd /sys/bus/w1/devices/`
- `ls`
- `cd 28-000005aaf7ed` The serial number will differ

(each unique)

- `cat w1-slave`

```
82 01 4b 46 7f ff 0e 10 70 : crc=70 YES
82 01 4b 46 7f ff 0e 10 70 t=24125
```

- Valid if the last value in first line is YES (passes CRC)
- second line has temperature in mili-degrees Celsius

DS18B20

- -55 to 125C
- +/- 0.5C from -10 to 85C
- 9 to 12 bit resolution
- Converts temp in 750ms
- Can set alarm

Other Busses not found on Rasp-Pi

Wireless

- No built-in wireless support, can use USB wireless

Bluetooth

- No built-in support.
- Sort of like serial, but wireless
- Not going to go over it in detail in this class though.

CANbus

- Automotive. Introduced by BOSCH, 1983
- One of OBD-II protocols
- differential, 2 wires, 1MBps important things like engine control
- single wire, slower cheaper, hvac, radio, airbags

CANbus Protocol

- id, length code, up to 8 bytes of data id (usually 11 or 29 bits) type and who is sending it. Also priority (lower is higher) length is 4 bits. some always send 8 and pad with zeros
- Type is inferred from id. Can be things like engine RPM, etc
- DBC database has the ids and values. ASCII text database, hard to get legally.

- Dominant/Recessive. Message with lowest ID wins arbitration.
- CAN-FD – extended version with larger sizes

CANbus Linux

- Can4linux – `open("/dev/can0"); read(); write();` External project?

- SocketCAN – contributed by Volkswagen. In kernel. Uses socket interface. /Documentation/networking/can.txt

CANbus on Pi

- No

ISA Bus

- Introduced with IBM-PC in 1981
- 8-bit (4.77MHz) then 16-bit (8MHz)
- +/-5V, +/-12V, 8 data, 20 address, DMA, IRQ
- Replaced by VLB (more pins, extra header), EISA (double pins in same connector), MCA micro-channel (different proprietary from IBM)

- Not enumerable at first, set jumpers. Later "Plug-n-Play"

LPC Bus

- Low-pin-count bus
- Intel, 1998, try to get rid of ISA
- Things like PS/2, Serial ports, floppy, etc.
  Still used for TPM Trusted Computing nonsense
- Replace 16-bit 8.33MHz parallel bus with 4-bit wide 33.3MHz bus. Only 7 wires. Easier to route than 72

“Conventional” PCI Bus

- Peripheral Component Interconnect
- Enumerable
- 1993, intel
- 62-pins, parallel, 133MB/s
- Extended with 32 or 64-bit versions, 33 or 66MHz, 3.3 or 5V. All slight differences in connectors to support all that.

- AGP (Accelerated Graphics Card) for graphics cards. 1997. Direct connect to CPU (not shared), multiple channels, faster clock

- PCI-X 1998, extension to 133MHz. Not to be confused with PCI-Express (PCIe)

PCI protocol

- 256B Config space, mapped into CPU address. Small area system can probe, used to setup larger mappings
- Can have on-board ROM that can be executed. Problem when using on non-x86 systems (emulators needed? special [expensive] PowerPC versions?)
- Latency timers keep bus-master from hogging bus
- 4 interrupt lines, can be shared. Level rather than edge-triggered interrupts make sharing easier

PCIe

- PCI-express, 2003
- Serial, replaced point-to-point with lanes, packet-based x1, x2, x4, x8 x16, x32
- Compatible with PCI at software level
- Differential Signaling
- External – Thunderbolt

- Serial better due to timing skew
- New x86 audrino quark has PCIe

PCIexpress Mini

- PCIe x1, USB, SMBus, etc
- Smaller card

PCMCIA Bus

- Personal Computer Memory Control International Association
- 16-bit
- Cardbus, 32-bit
- Mostly replaced these days

PC/104 Bus

- Stackable small x86 boards usually
- Run ISA or PCI signals up vertically

VME Bus

- m68k bus but generic enough
- Still found in some embedded systems

Other

- SATA, eSATA, PATA, SCSI (disk drives)
- Firewire
- RapidIO
- Quickpath QPI
- Hypertransport