Introduction of Low Speed Serial Comm.

이기응 이사
Application Engineer Manager
Agenda

- Introduction
- Probing Digital Signals and Digital Thresholds
- Review of Serial Buses
  - I²C, SPI, Audio
  - RS-232,
  - CAN, LIN, FlexRay

Example automotive application (source: Infineon)
Debug Challenges of Serial Data

- Parallel Buses
  - Each line has its own signal path
  - Clock is generally a separate line
  - Easy to decode
  - State and Pattern triggering and decoding are straightforward with a logic analyzer or mixed signal oscilloscope

- Serial Buses
  - Signals are spread over time
  - Clock is sometimes embedded
  - Decode is tedious
  - Must decode first to trigger on packet
  - Analysis solutions available on some oscilloscopes

*Serial data complicates bus troubleshooting*
Embedded Serial Bus Debug Overview

- Bit rates below about 10 Mb/s
- Various bus topologies are used, not limited to point-to-point

- Signal Integrity is usually not a big issue
  - Physical layer compliance testing is usually not required
  - 10 Mb/s FlexRay is an exception

- Primary applications for test equipment are:
  - Hardware debug
  - Hardware / software integration
  - System-level performance verification and optimization
Probing Serial Digital Buses

- Digital buses are not digital
- Digital probes are not digital
- Everything you know about analog probing still applies

The real signal must be delivered to the oscilloscope’s comparator, where it can be compared to the digital threshold value.
What is the right probe to use for digital signals?

- Analog and digital probes are both voltage probes
- Minimize loading:
  - Use high resistance to minimize DC loading, especially on low-power, high-impedance circuits
  - Use low capacitance to minimize impact on risetime, and to accurately acquire pulse widths
- Voltage measurements made relative to ground reference
  - Common to all channels
- Use short grounds for high-frequency signals
  - Ground lead inductance causes ringing
- Connect to DUT
  - Requires a selection of probe tips, clips, headers, and adapters
  - May require connection to distant test points
- Usually, you can use the standard MSO probe
Probing differential signals

- Differential signaling used to improve noise immunity with some serial standards
- Differential probing responds only to differences between signals
  - Rejects common-mode noise
  - Ignores DC offsets
  - Provides most accurate results
- Single-ended probing may be used where:
  - Signals are large
  - Rise- and fall-times are fairly fast
  - Noise is small
  - Thresholds are set carefully
Thresholds

- For digital channels
  - thresholds are analog voltages applied to hardware comparators
  - comparison made during acquisition
  - threshold range is hardware-limited
  - threshold accuracy is $\pm 100\, \text{mV}$, including some hysteresis
  - automatically set by Autoset

- For analog channels
  - thresholds are software values applied during decoding
  - threshold value on trigger source tied to trigger level
  - threshold range is limited like trigger level
  - only trigger level is set by Autoset
How do I debug a potential threshold problem?

- Decoding on this FlexRay signal is sometimes incorrect, as shown at right.
- What one display will point to the problem in the oscilloscope setup?

- Using the Bus and Waveforms display, you can directly see the effects of the threshold settings:
  - Digital waveform shows spurious transitions
  - Upper threshold level is likely a little too high
How do I debug a double-probing problem?

- Decoding on both signals is sometimes incorrect, as shown at right
- Decoding on the CAN signal on channel 1 works OK alone
- Decoding on the CAN signal on channel D0 works OK alone

- What one display will point to the problem in the oscilloscope setup?
- Using the Bus and Waveforms display, by the differences in pulse widths, you can directly see the effects of the threshold settings
Serial Bus Review

- Separate clocks:
  - I²C
  - SPI
  - Audio (I²S, LJ, RJ, TDM)

- Embedded clocks:
  - RS-232 (RS-422, RS-485, UART)
  - CAN
  - LIN
  - FlexRay
I²C (Inter-Integrated Circuit)

- Developed in the 1980s by Philips
- Used for chip-to-chip communication between microcontrollers and A/Ds, D/As, FPGAs, sensors, etc.
- Uses two single-ended, bi-directional signals: clock and data
- Any I²C device can be attached to the bus
- Data rates:
  - Standard Mode (100 kbps)
  - Fast Mode (400 kbps)
  - High Speed Mode (3.4 Mbps)
Introduction to I²C

- I²C stands for Inter Integrated Circuit (a.k.a I2C and IIC)
- Network uses two-wires: clock (SCL) and data (SDA)
- A serial synchronous, multi-master, bi-directional, layered communication network
- Data rates: standard mode = 100Kbps, fast mode = 400kbps, high speed mode = 3.4Mbps
- Maximum number of devices connected to the bus is roughly 20 – 30; limited by the maximum bus capacitance of 400pF
- Used for chip-to-chip communication, such as with: I/O, A/D’s, D/A’s, FPGA’s, sensors, microcontroller, …
- 7 or 10 bit addressing modes
I^2C Data Link Layer

- Start – Device is taking control of the bus
- Repeated Start – Start occurs without a previous stop
- Address – Address of the device (7 or 10 bit)
- Read/Write – Indicating if the data will be read ("1") or write ("0")
- Acknowledge – Slave device acknowledging the master’s actions. (ack = 0)
- Stop - The message is complete and the master has released the bus.
Example - I²C Hand Decoding

Start of packet: A HIGH to LOW transition on the SDA line while SCL is HIGH

Stop of packet: A LOW to HIGH transition on the SDA line while SCL is HIGH

Missing ACK = 1
ACK = 0

Read = 1
Write = 0

<table>
<thead>
<tr>
<th>Start/Stop</th>
<th>Addr</th>
<th>R/W</th>
<th>Ack</th>
<th>Data 0</th>
<th>Ack</th>
<th>Data 1</th>
<th>Ack</th>
<th>Data N</th>
<th>Ack</th>
<th>Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 bit</td>
<td>7-bits</td>
<td>1-bit</td>
<td>1-bit</td>
<td>8-bits</td>
<td>1-bit</td>
<td>8-bits</td>
<td>1-bit</td>
<td>0-8 bytes</td>
<td>8-bits</td>
<td>1-bit</td>
</tr>
</tbody>
</table>
Example - I²C Bus Decoding
Automated Decode with Tektronix’ Oscilloscopes

Start

Address
[W] for Write, [R] for Read
Displayed in hex or binary

Data
Displayed in hex or binary

Stop

Tektronix
MSO/DPO4000B Series
Automated Decode
Event Table for Viewing Bus Traffic

- Shows decoded message content with time stamps
- View bus traffic in tabular format
- Compare with software listings
- Easy timing measurements
Capture a Specific Message

- Even if you can easily decode messages, the message of interest probably wasn’t captured
- Need to specify messages to capture:
  - In the language of the serial bus standard
  - On all critical elements of the serial message
  - With full or partial specification
- Tektronix’ oscilloscopes offer serial data triggers
An Example: Faulty Thermal Management System

- The product is overheating and shutting off.
- Microprocessor-controlled thermal management system should sense the product’s internal temperature and adjust the fan speed.
  - All of the circuits appear to have the correct power applied.
  - The processor is running and appears to be communicating with the sensors and the fan control module.
  - The software team is sure that the software is running as designed.
- Yet, the product is getting hot and the fan is not turning on.
Trigger on Packet Content

- Trigger on address 18 (sensor).
- Software tries to communicate with the sensor – twice!
- No response.
- Moves to the next address, as designed.
- Upon close inspection of the board, a cold solder joint was found on the fan controller IC.
Search for a Specific Message

- Even if you capture the message of interest, now you have to find it!
- Need to specify messages to **search** for:
  - In the language of the serial bus standard
  - On all critical elements of the serial message
  - With full or partial specification
  - Same conditions as needed for **capture**
- Tektronix’ oscilloscopes provide automatic search and mark capabilities
I2C Bus Variants

- **SM (System Management Bus) Bus**
  - Protocol standard is similar to I2C except the last data byte is used to represent Packet Error Check

- **PM (Power Management) Bus**
  - A standard way to communicate with power converters over a digital communication bus
  - Similar to I2C bus at protocol layer
  - User configure, control and monitor different power supplies in a electronic system
I2C Bus Variants

- **IPMI Bus**
  - Intelligent Platform Management Bus based on I2C bus
  - Intel Initiative along with HP, DELL and NEC
  - Monitors Temp, Voltage, Fan and general assistant in Automatic alerting, auto restart and so forth
  - Used in enterprise servers and telecom systems

- **HDMI Bus**
  - Display Data Channel (DDC) is I2C. This bus is used to control and setup source and sink configuration
Target Industry-I2C Bus

- Consumer Electronics – LCD and Plasma TV
- Computing Market-Network/Enterprise Servers
- Switch Mode Power Supply Design Market
- Mobile Phone Industry
I2C variants in Emerging Applications

- **I2C in HDMI**
  - HDMI Bus
    - Display Data Channel (DDC) is I2C. This bus is used to control and setup source and sink configuration
  - PM (Power Management) Bus
    - A standard way to communicate with power converters over a digital communication bus
    - Similar to I2C bus at protocol layer
    - User configure, control and monitor different power supplies in a electronic system

- **PM Bus: similar to I2C**

![Diagram of HDMI and PM Bus]
I2C variants in Emerging Applications

- **FB-DIMM: SM Bus**

- **SM (System Management Bus) Bus**
  - Protocol standard is similar to I2C except the last data byte is used represent Packet Error Check

- **IPMI Bus**
  - Intelligent Platform Management Bus based on I2C bus
  - Intel Initiative along with HP, DELL and NEC
  - Monitors Temp, Voltage, Fan and general assistant in Automatic alerting, suto restart and so forth
  - Used in enterprise servers and telecom systems

- **Other**
  - DDR memory
  - LCD/Plasma TV
  - Reading/writing into EEPROM
SPI (System Peripheral Interface)

- Developed in the 1980s by Motorola to communicate between microcontrollers and their immediate peripheral devices
- Typical configuration has four signals: SCLK, MOSI, MISO, SS
  - Data is simultaneously transmitted and received
  - SS line used to specify slave device
  - Each unique device on bus has its own SS signal from master
- Multiple bus configurations are allowed
  - Network can use 2-, 3-, or 4-wire bus topology
- Data rates up to 10 Mbps
  - SS – enables slave device to accept data
  - MOSI – data from the master to a slave
  - MISO – data from a slave to the master
  - SCLK – serial clock driven by Master
SPI (System Peripheral Interface)

- 3 Wire bus
  - MOSI, CS(SS) and Clock

- 4 Wire Bus
  - MOSI, MISO, CS(SS) and Clock
PDS-R On-line SPI Protocol Decode Software

- Fully Integrated with DPO7000/70000 and DSA70000 series oscilloscopes
- PDS-R will occupy bottom half of the oscilloscope display
- Simultaneously view live waveform and decoded data
PDS-R On-line SPI Protocol Decode Software

- Decodes live and stored SPI Signals
  - Simultaneously displays decoded MOSI, MISO data
  - Time Stamp
  - Zoomed time domain waveform is displayed at the top of the scope for selected frame in decoded listing table
  - Data can be saved in CSV file format
PDS-R On-line SPI Protocol Decode Software

- In detailed view mode:
  - Waveform is plotted
  - Decoded bitwise information is overlaid on the waveform
- Ability to view data 8 bit, 16 bit or 32 bit format
- Search by MOSI and MISO
Audio

- Used for communicating serial audio data between ICs in audio electronic products. Examples include:
  - Inter-IC Sound (I²S)
  - Left Justified (LJ)
  - Right Justified (RJ)
  - Time Division Multiplexed (TDM)

- Synchronous, bi-directional serial interfaces
  - High-resolution, low cost, low jitter data transmission
  - Primary differences are in timing

- Physical bus consists of three lines: Bit Clock, Word Select, and Data
- Data rates up to a few Megabits/second
RS-232 (Recommended Standard-232)

- Point-to-point communication at slow speeds over short distances
- Two single-ended signals provide point-to-point, full-duplex communication
- Standard does not specify character encoding, data framing, or protocols
- Transmission systems:
  - Managed by Universal Asynchronous Receiver/Transmitters (UARTs)
    - Pre-determined bit rate
  - RS-232 is an inverting, single-ended high-voltage interface
  - RS-422 or RS-485 are differential interfaces
  - Or ICs can be connected directly
Local Area Network (LAN)
CAN (Controller Area Network) Review

- Used for system-to-system communication in Automotive, Industrial Automation and Medical Equipment
- A serial asynchronous, multi-master, layered communication network
- Data rates of 10 kbps - 1 Mbps
- Sophisticated error detection and error handling mechanisms
- Flexible signaling support for low-cost implementation
- Physical bus is single-wire, dual-wire, and fault tolerant

CAN Physical Layer

CAN Controller

Electronic Control Unit

CAN L

Tx

Rx

Rx

Tx

2013-07-10
CAN Data and Remote Frame Overview

- **SOF** - begins with a start of frame (SOF) bit
- **Arbitration** - includes Identifier (address) and Remote Transmission Request (RTR) bit
  - Identifier can be standard format (11 bits - version 2.0A) or extended format (29 bits - version 2.0B)
  - RTR used to distinguish between a data frame and a data request frame, also called a remote frame
- **Control** - 6 bits including the Identifier Extension (IDE) bit and Data Length Code (DLC)
  - IDE distinguishes between CAN 2.0A (11 bit identifier) standard frame and CAN 2.0B (29 bit identifier) extended frame
  - DLC is a 4 bit indication of the # of bytes in the data field of a Data frame or the # of bytes being requested by a Remote frame
- **Data** - consists of 0-8 bytes of data
- **CRC** - a 15 bit cyclic redundancy check code and a recessive delimiter bit
- **ACK** - acknowledge field is two bits long
  - First is the slot bit, transmitted as recessive, but then overwritten by dominant bits transmitted from any node that successfully receives the transmitted message
  - Second bit is a recessive delimiter bit
- **EOF** - 7 recessive bits indicate the end of frame (EOF)
- **INT** - intermission field of 3 recessive bits indicates the bus is free
  - Bus Idle time may be any arbitrary length, including 0
CAN Solution on the DPO/MSO4000 Series Oscilloscope

Search and Mark feature

Event table displays decoded CAN message frames with timestamps
LIN (Local Interconnected Network) Review

- LIN communication is based on UART/SCI with data being sent in eight-bit bytes along with a start bit, stop bit and no parity.
- Data rates range from 1kbps to 20kbps
- LIN frames consist of two main parts, the header and the response. The header is sent by the master while the response is sent by the slave.
LIN Message Frame

- **Header components**
  - Break Field: Used to signal the beginning of a new frame
  - Sync Field: Used by the slave devices to determine the baud rate being used by the master node and to synchronize themselves accordingly
  - Identifier Field: Specifies which slave device is to take action

- **Response components**
  - Data: Specified slave device responds with 1 - 8 bytes of data
  - Checksum: Computed field used to detect errors in data transmission
LIN Solution on the DPO/MSO4000 Series Oscilloscope

- LIN triggering capability in the 4000 Series activates on:
  - Sync field (shown)
  - ID
  - Data & ID/Data
  - Wakeup & Sleep Frame
  - ID parity errors
  - Checksum errors

- Trigger on LIN bus (Bus1), with Bus2 and Bus3 being captured and decoded simultaneously

- All bus traffic can now be viewed as it goes from one bus to another through a gateway, and the system
FlexRay Review

- FlexRay is a relatively new automotive bus was developed by a group of leading automotive companies and suppliers known as the FlexRay Consortium.

- The physical bus can be unshielded twisted pair or shielded twisted pair to improve EMC performance.

- FlexRay is a differential serial bus configured in three recurring segments: Header, Payload and Trailer.

- Each frame contains a static and dynamic segment, and bus idle time concludes each frame.

- Transmitted data rates up to 10 Mbps.
FlexRay Frame Structure

- **Header Segment**: Contains Indicator Bits, Frame ID, Payload Length (in words), Header CRC, and Cycle Count
- **Payload Segment**: Contains data transferred by the frame. Maximum payload length is 127 words (254 bytes)
- **Trailer Segment**: Contains a single 24 bit field [three 8 bit CRC registers] for header and payload protection
The DPO/MSO4000 Series is triggered on the FlexRay buses TSS (Transmission Start Sequence) which initiates network connection setup, followed by frame Id (02), Header CRC, and Payload (data).
FlexRay Solution on the DPO7000 Series Oscilloscope

- Fully integrated with DPO7000 series oscilloscopes
- Half screen mode operation
- Decoded message is linked to waveform display by cursor
- Any row can be linked to waveform and zoomed waveform is displayed

PDF-R FlexRay Protocol decode software
Characterizing System Timing

- Characterize timing between bus messages and system operation
  - Requires waveform displays time-correlated with decoded messages

- Characterize timing differences which occur when adding a new network node to an existing network

- Automotive application example:
  - Measure worst-case time from crash sensor output to airbag activation
  - Measure variations in timing of airbag activation with varying levels of CAN bus traffic
Summary

- Serial buses are pervasive, creating a unique set of measurement and analysis needs
- Making measurements needs to be easier, faster, and more accurate
- Requires an oscilloscope with triggering, decoding and analysis tools for serial protocols

**MSO/DPO Series**

- Models from 100 MHz to 1 GHz
- Up to 10 M record length
- Serial data triggering and analysis options:
  - I²C, SPI
  - RS-232/422/485/UART
  - CAN, LIN
  - FlexRay *(MSO/DPO4000 Series)*
  - Audio (I²S, LJ, RJ, TDM) *(MSO/DPO4000 and MSO/DPO3000 Series)*
Thank you