SBS IF Developer's Conference '98

Handheld Data Set Specification Proposal

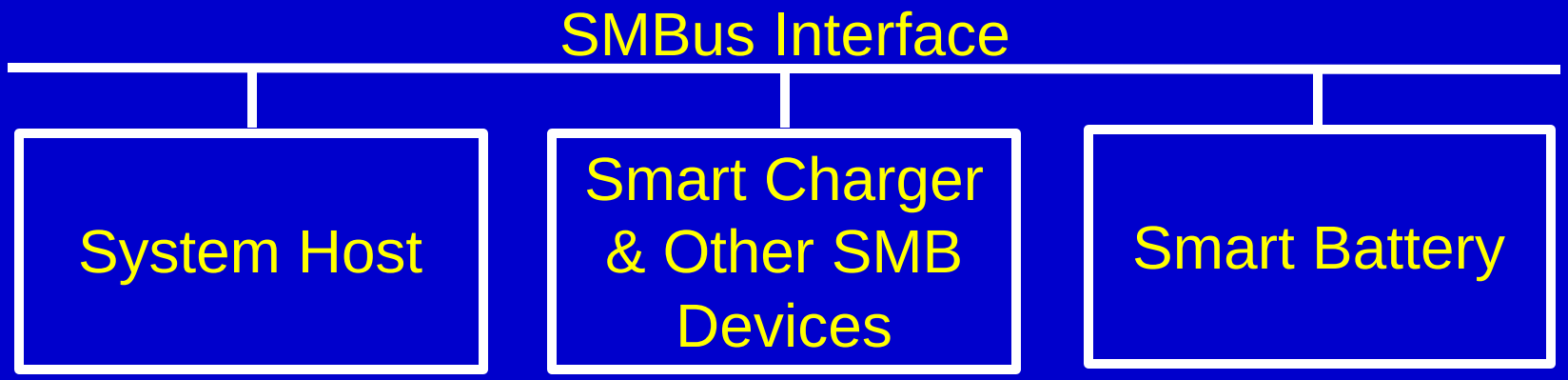
Dale Stolitzka,

National Semiconductor Corporation

Topics for Discussion

- Does SBS differ for Handhelds?
- Introduce Handheld Data Set Concepts
- Propose Addressing Protocols
- Current Handheld Data Set Proposals
- Review Popular Error Checking Methods
- Show a Path for Comments & Actions

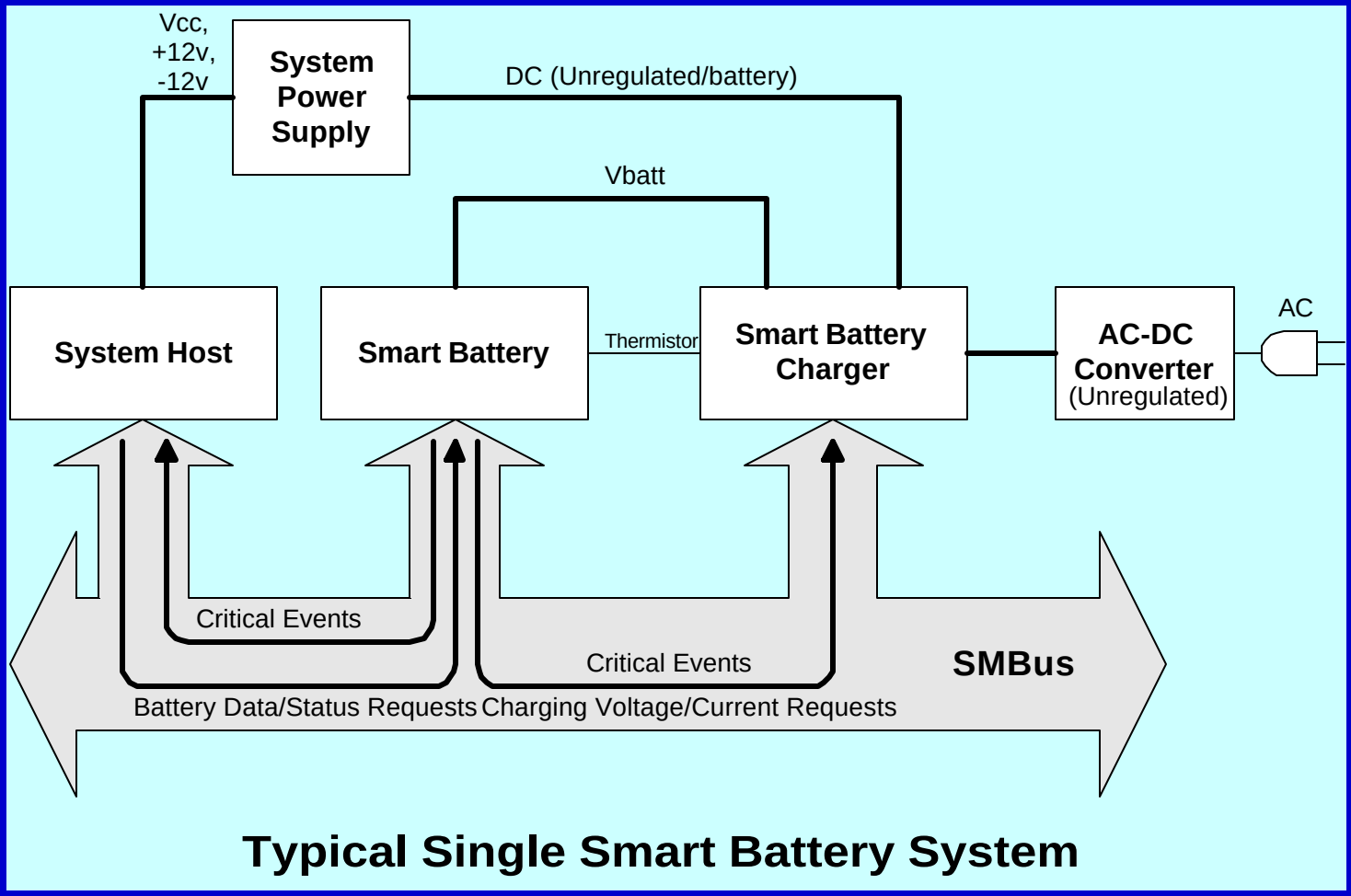
Smart Battery System Evolution



This figure is reproduced with permission from the SBS Core Group, 1997.

- This set of specs. compose a bus and command protocol standard (co-owned by ten companies)
- The physical layer is SMBus, a synchronous, serial, two-wire bus based upon *Philip's* i^2c.bus.

Typical SBS System



Typical Single Smart Battery System

"Handheld Data Set Specification and Proposal" by National Semiconductor

Parts of an Electronic Battery

- Despite standards, batteries carrying electronics contain:
  - fuel gauging
  - host-battery communications
  - fault protection (electronic and passive)
  - abuse-tolerant design

Parts of a Smart Battery

- Options found only in some batteries:
  - charge control using a Smart Battery Charger
  - in-pack charge control
  - historical usage pattern recording

Similar Issues Apply to Handsets

Notebooks

- Smart Battery is here
- Multiple batteries are in Notebooks
- Standardization is driving cost down
- SMBus is accepted

Mobile Handsets

- Smart Battery is coming
- Multiple batteries are in one charger base
- Cost is an overriding concern
- Many physical bus alternatives

"Handheld Data Set Specification and Proposal" by National Semiconductor

What about an SBS in Handhelds?

- The SBS chartered a new Special Working Group to define handheld smart batteries
  - ...works on communications, not the pack
  - ...must be far simpler than its notebook cousin
  - ...makes sense by driving costs down.
  - ...must encompass multi-bay battery charging

"Handheld Data Set Specification and Proposal" by National Semiconductor

Objectives of a Handheld Standard

SBDS in PC's

- SMBus only
- 16-bit data
- Ack/Nak Error in i^2c
- Master/Slave
- 33 commands

HDS

- Public domain bus
- 8 or 12-bit data
- Error handling Needed
- Slave only
- $\leq$10 commands

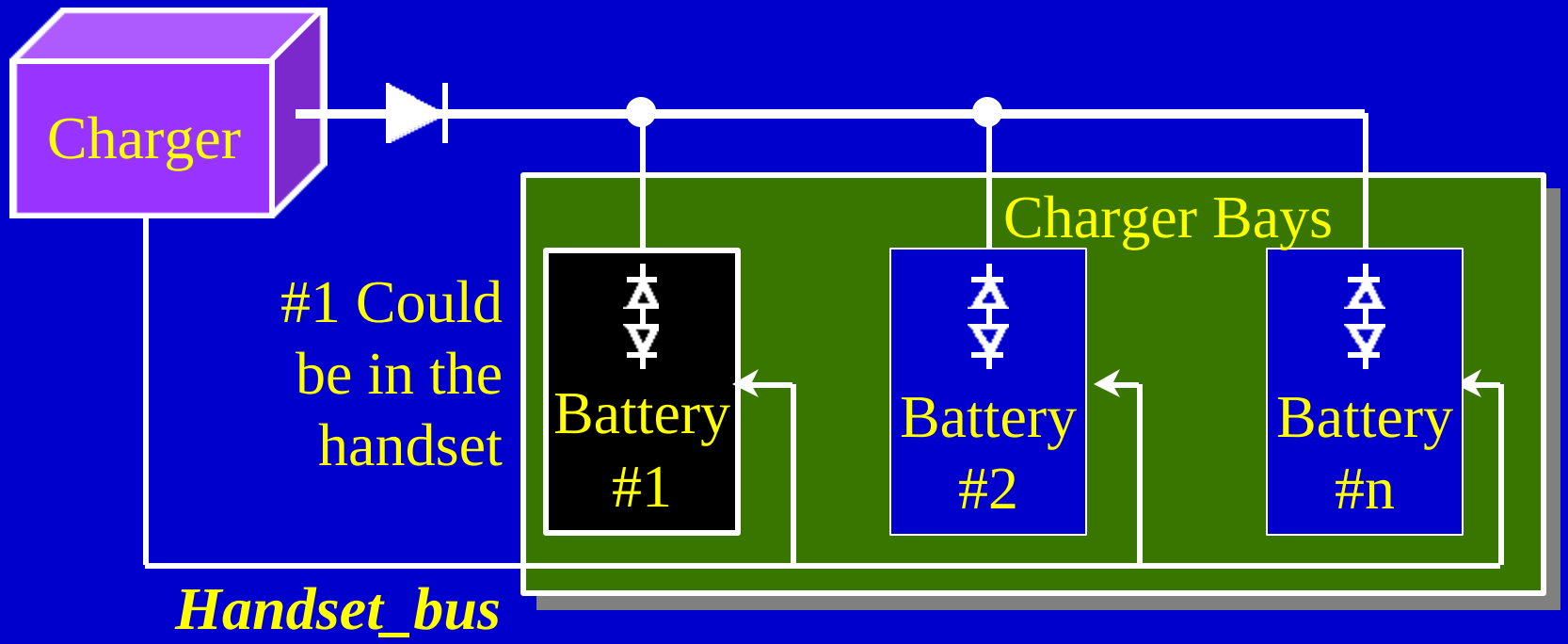
"Handheld Data Set Specification and Proposal" by National Semiconductor

Physical Layer Addressing

- There are a number of ways to connect the Host and Battery
  - Software addresses in some buses
    - SMBus, i^2c.bus, others
  - Point-to-point
    - UART, bit-by-bit interpretation
  - Hardware-select line methods
    - SPI™, Microwire™

Common Handset System

- When each battery has the same address, some selection method is needed



Physical Layer Addressing Leads to Architecture Choices

- SMBus used software addressing to relieve pin congestion in the notebook
- Multiple batteries use a default address
  - SMBus write address is 0x16
  - Use a selector to distinguish between multiple batteries
  - Selector uses its own address, 0x14
- Is selector cost-effective in the handset ?

"Handheld Data Set Specification and Proposal" by National Semiconductor

Selector Methods

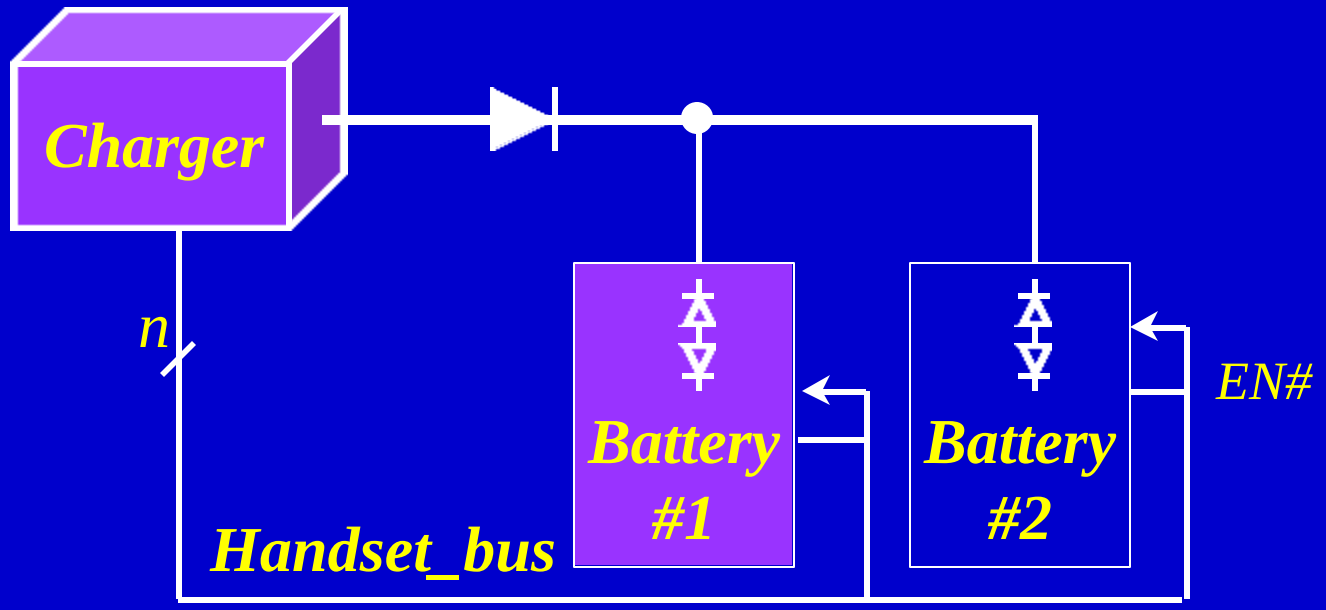
- (1) Use a selector-bridge to separate each battery from the bus unless selected
  - Selector switches power and bus lines
- (2) Use a selector to switch hardware enables
  - Introduce one additional pin / pack
- (3) Add software selection to the battery
  - Adds complexity to software; no switches

Selector Impact on the Data Protocol

- Hardware selector needs:
  - Selector requires an address location
  - Need selector commands & definition
- Software selector needs
  - Simple enumeration method
  - Default address and a “Bus Reset”

Hardware Enable in the Charger Bay

- Hardware enable can differentiate batteries
- Internal battery switches can serve as selector switches rather than just protection



Software Distinction -- Multiple Batteries

- Distinguish batteries in a system by reassigning addresses in low nibble of an 8-bit address
  - **Reset = Batteries use default addr (0x20)**
  - **ID = All batteries echo SERNO**
    - Arbitrate in favor of the first '0'
    - Host assigns 0x21 to first responder's address.
  - **Send ID again, non-default addr does not respond to ID command**
    - Host assigns next battery 0x22 and so forth

Software Addressing Example

- Reset command = 0x90
- ID command = 0xa

{Preamble}	Preamble, if required by bus
+ 0xa + 0x1	ID command + addr low-nibble
+ SERNO	Battery SERNO response (N bytes)
+ {buffering}	Optional buffering byte(s)
+ 0x21	Battery finished SERNO, sends addr
+ Check byte	Battery sends check, *e.g.*, CRC-8
+ {End}	End-of-msg bit or byte as required

Possible Command Sets

Proposed Data Set at Plugfest V

- 10 commands
- 5 Optional registers (free)
- 2 Optional-implementation Alarm registers
  - RemainingTimeAlarm()
  - RemainingCapacityAlarm()

Two Proposed Command Sets

- Command Set in ver. 0.6 @Plugfest V
  - Changed data width to 8 and 12-bits
  - Eliminated redundant commands
  - Stressed need for optional register space
- Sony Power'97 paper, "BODIP"
  - Single-wire in the hardware
  - Simple command structure
  - "One-way" communication and UART

"Handheld Data Set Specification and Proposal" by National Semiconductor

Proposed Data Set Commands

0x08	Temperature()	12-bit data
0x09	Voltage()	
0x0a	Current()	
0x0d	RelativeStateOfCharge()	
0x10	FullCapacity()	
0x17	CycleCount()	
0x18	DesignCapacity()	

0x00	Access()	12-bit field
0x16	Status()	

0x20	Configuration()	data block

Simple Command Protocols

- Read & Write Data
  - Start with synchroniztion preamble (option)
  - Command and Address byte
    - 16 commands
    - 16 addresses, if enumeration is used
  - Data byte(s)
  - End with Check byte

Block Protocol Usage

- Configure battery during production
- Read groups of data that otherwise require separate commands, *e.g.*,

Manufacturer's Name
Battery Model #
Battery Serial #
Battery Date Code
IC (code) Revision #
Optimal Charging Current

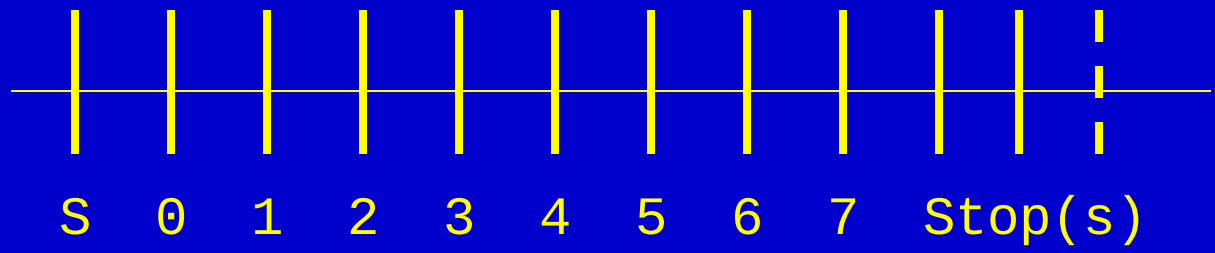
BODIP Discussion[1]

- Sony defines three fields, each 16 bytes
  - 1st field = customer data
  - 2nd & 3rd field = manufacturer data
- One-way broadcasting
- All fields sent in a message @1200bps

(1) All BODIP discussions reference material to Sony RME Company, *Power'97*, 10/97.

BODIP Protocol

- Each field has a two-byte start block (0x0707) and end block (0x8787)
- Data block in the field is 12 bytes
- Each byte contains a UART-style start bit, stop bits and 8 data bits



X-bus Protocol

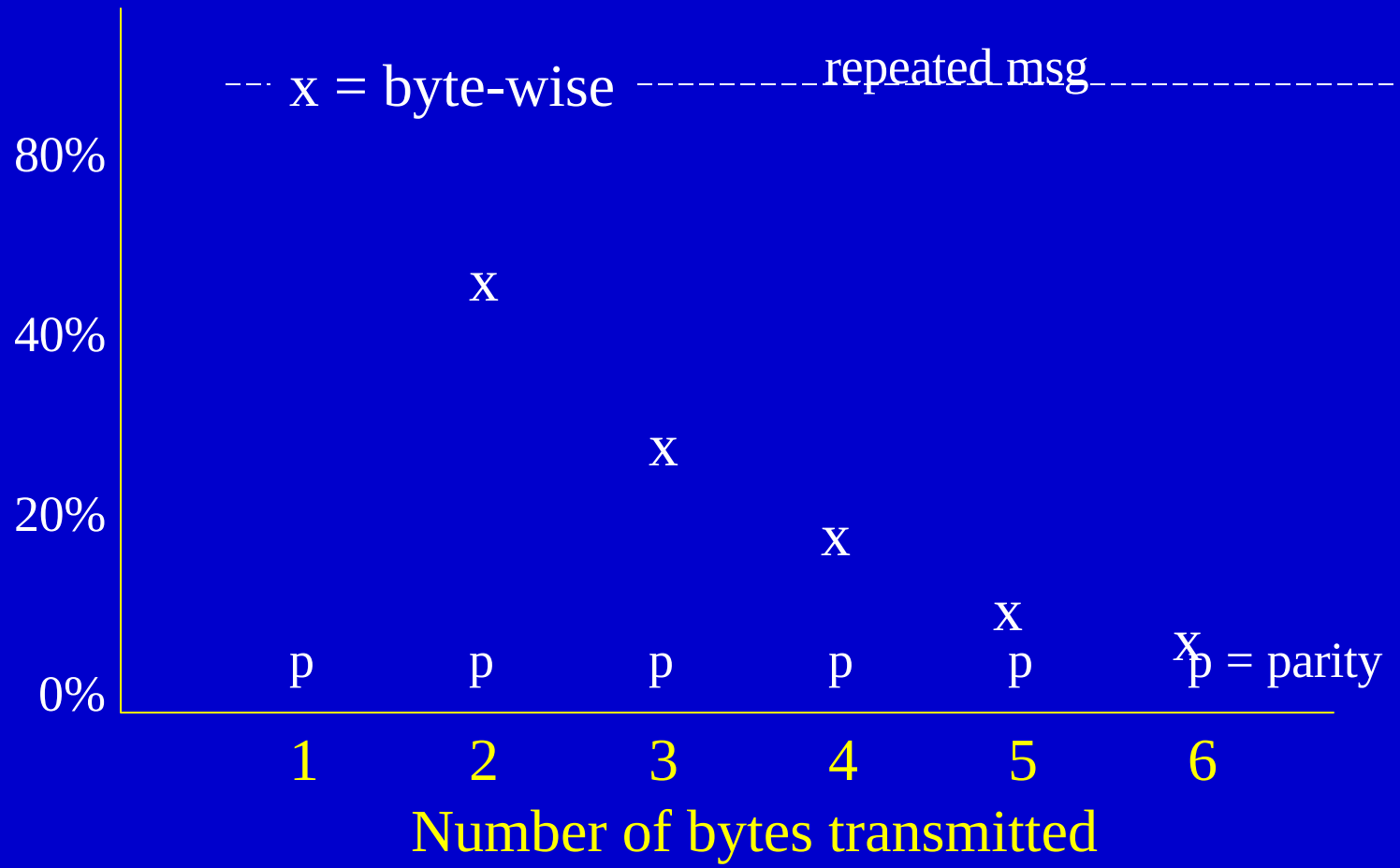
- Uses a simple protocol
- Released to public domain by Lenz
- Typical usage at 9600 BAUD but can go to 62.5kbps

Ref: http://www.lenz.com/techinfo/x-busfaq.htm

Common Error Handling Methods

- Bit-wise overhead on a byte-wise basis
  - Ack/Nak, *e.g.*, SMBus
  - Parity
- Byte-size mechanisms
  - Exclusive-OR of previous bytes
  - 2-D Parity
  - Checksum
  - CRC-8

Error Handling Overheads

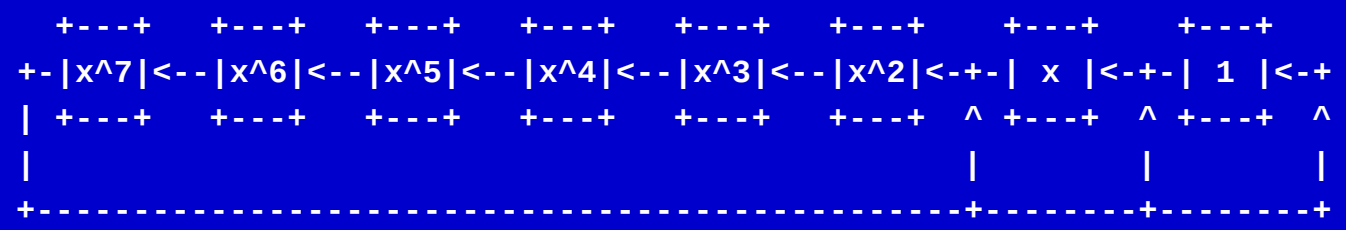


Byte-size Error Handling

- Exclusive-OR
  - Error = byte 1 ⊕ byte_2 ⊕ byte_3 ⊕ …
- Checksum
  - Error = Σ(message bytes)
- 2-D Parity
- CRC-8
  - $C(x) = x^8 + x^2 + x + 1$

CRC-8 Hardware Model

- Hardware shift register generates the CRC-8 [ref: M. Maa, 1997]



Summary

An SBS for handheld communications--

- Covers communications, not the pack
- Can be simpler than its Notebook cousin
- Makes sense by driving costs down.
- Encompasses multi-bay battery charging
- Ensures data integrity in RF environment

Tasks for the Special Working Group

- Select the physical layer
- Amend the Data Set?
- Select an error-reporting method
- What is the best way to handle multiple battery chargers?

How To Act

- Vote in the Working Group's Referendum
- Actively interested?
  - Join & co-author the new specification
- Send your comments to the Handheld Data Set Special Working Group
- SBS website: http://www.sbs-forum.org
- email: hhdata@sbs-forum.org