SBS IF Developer's Conference '98

Aloha

SMBus in Action

John Milios

Agenda

- SMBus current status
- Implementing an SMBus port
- Bus Failures
- Designing an SMBus system
- More Addresses
- Looking forward

SMBus

On-board Serial Control Bus

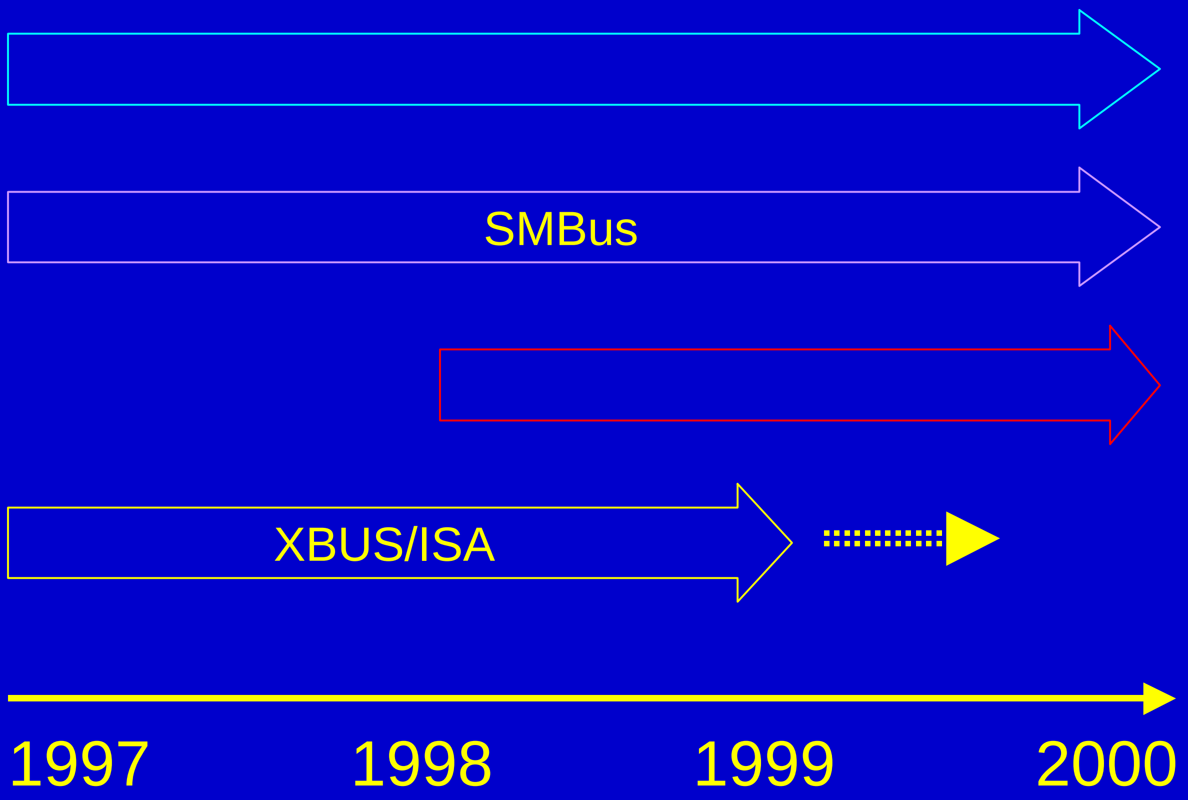
- Multi-drop
- Accommodates Devices with Different

- Low Power
- Random Access
- Multi-Master
- Now with O/S Support through ACPI

Factors that will accelerate the adoption of SMBus

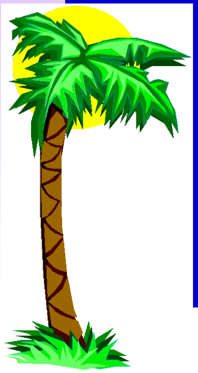
- System architecture changes
  - Existing functions will move around to new
- New applications and needs
  - New functions will need to be serviced in

ISA Departs the Scene

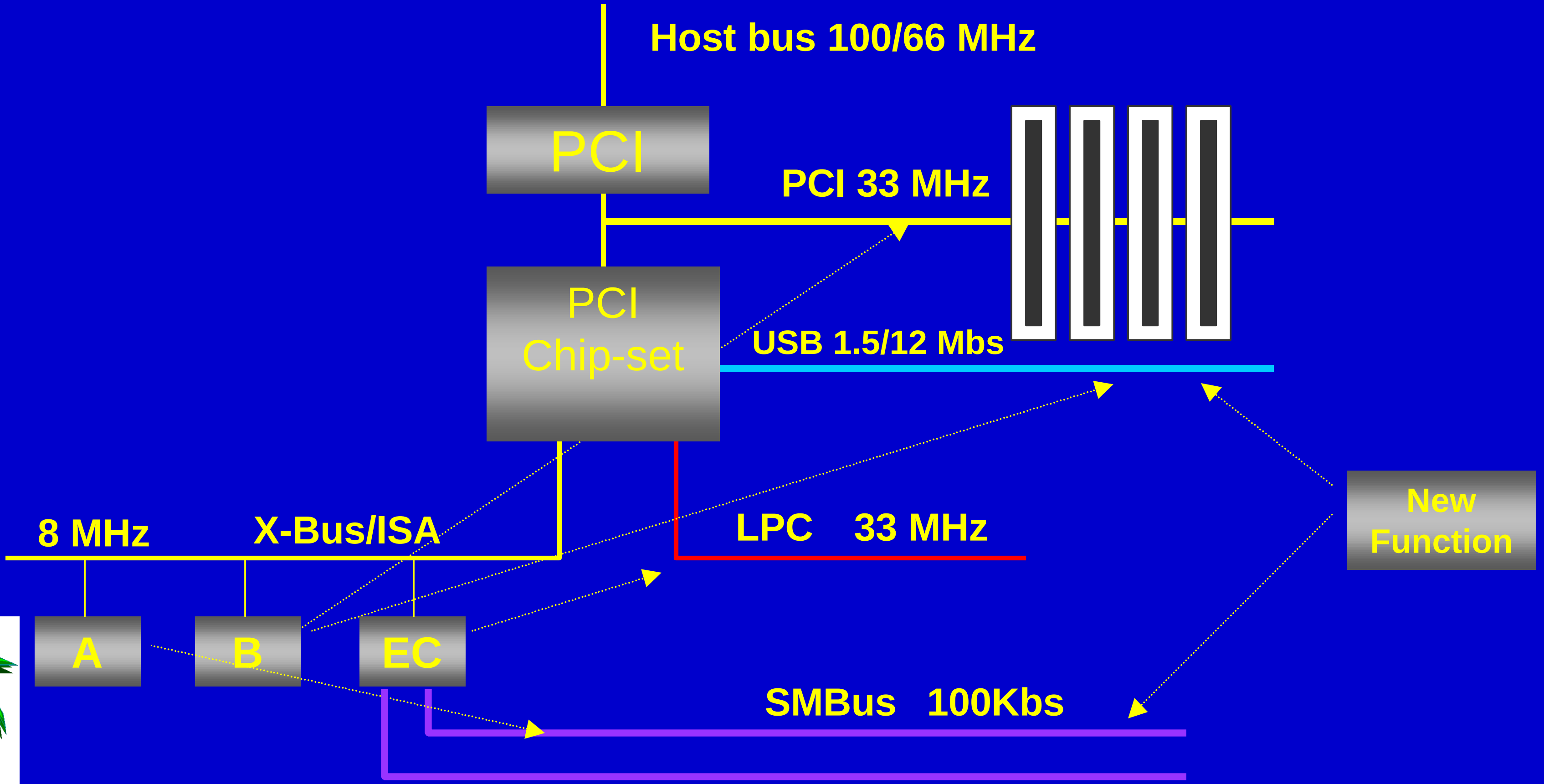


SMBus

XBUS/ISA



Functions will Move Around

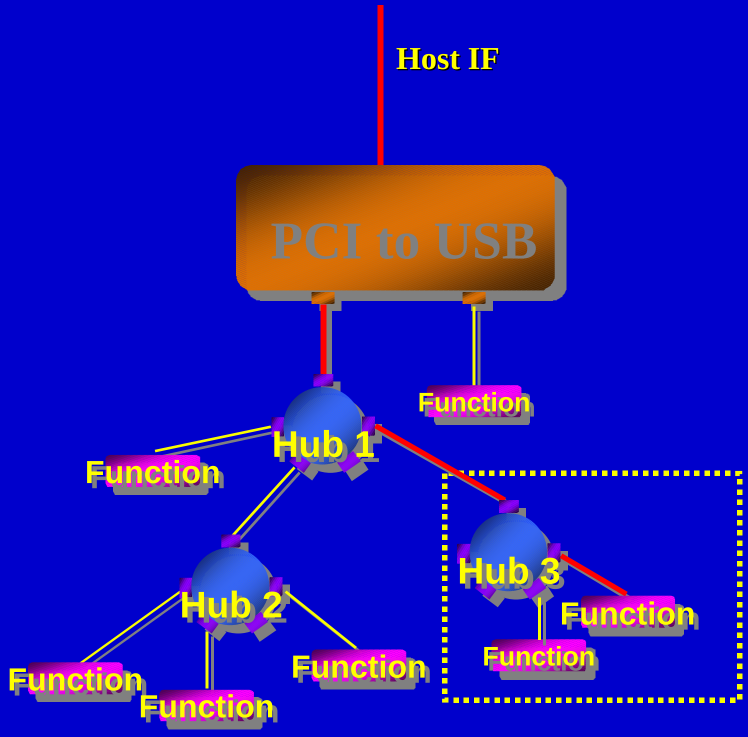


Uniqueness of SMBus

- EC provides preprocessing of SMBus
  - Specific tasks can be handled by the EC without CPU intervention
- Low Power
- Low cost

Advantages over alternative On-board Serial Buses

- USB
  - $1^{1/4}$ IC per Function (1/4 for HUB)
  - Every Transaction involves the CPU
  - High Power Consumption



Type of Functions

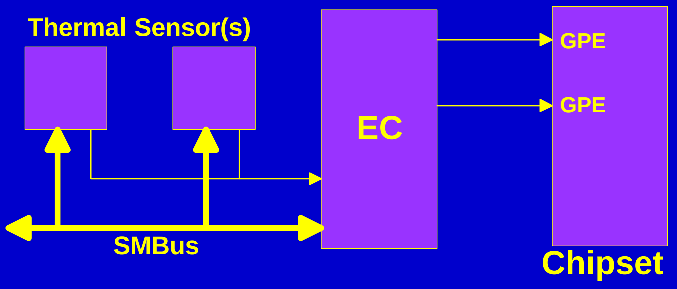
- System Management Functions
- Functions that can be served by the EC
- Functions that need to be served even when the system is suspended
- Low throughput and/or infrequent bus
  - A 1KByte message would take 170-200 ms

Factors that will accelerate the adoption of SMBus

- System architecture trends
  - Existing functions will move around to new
- New applications and needs
  - New functions will need to be serviced in

Current Applications

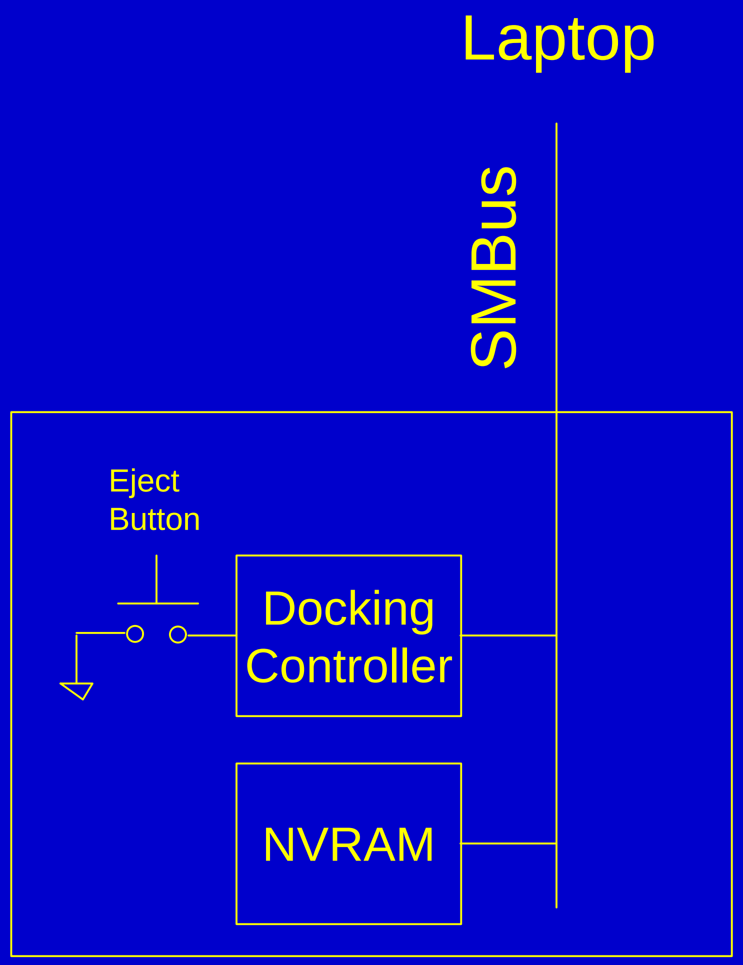
- Smart Battery System
- Temperature Sensors
  - Specification Version 0.6
  - Silicon by Maxim



- Flat Panel Display Digital Controls
  - Specification Version 0.6
  - Silicon by O2Micro

Applications On the Works

- Docking
- Network Cards
- Bus Control



Future Applications

- Remote Radio Control
- Security (Smart Card Interface)
- Simple Messaging Services
- Other

SMBus Success Factors

- Number 1: It has to work reliably
- Number 2: It's irrelevant if the #1 criterion is not met

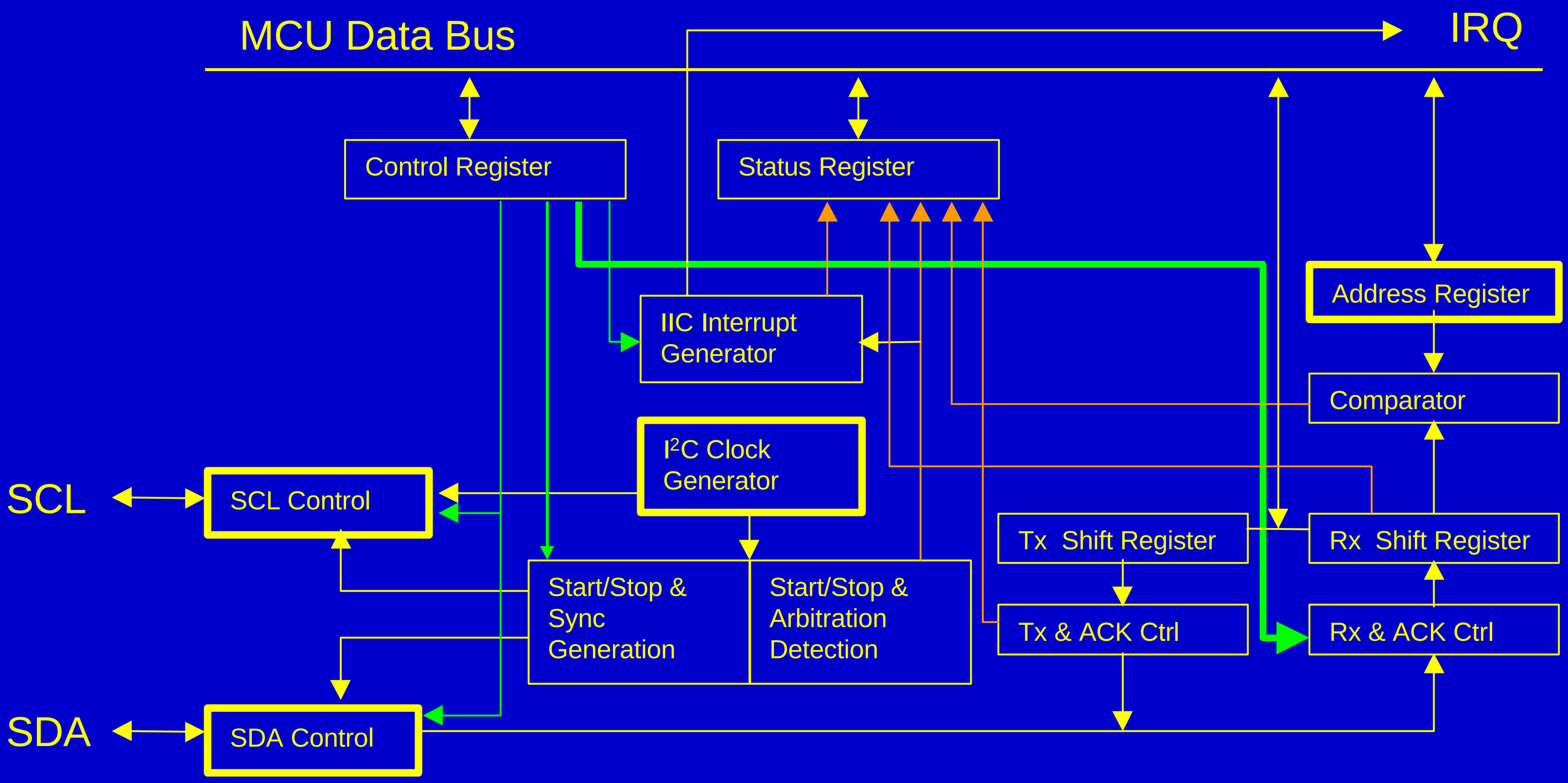
Agenda

- SMBus current status
- Implementing an SMBus port
- Bus Failures
- Designing an SMBus system
- More Addresses
- Looking forward

The SMBus Device Interface

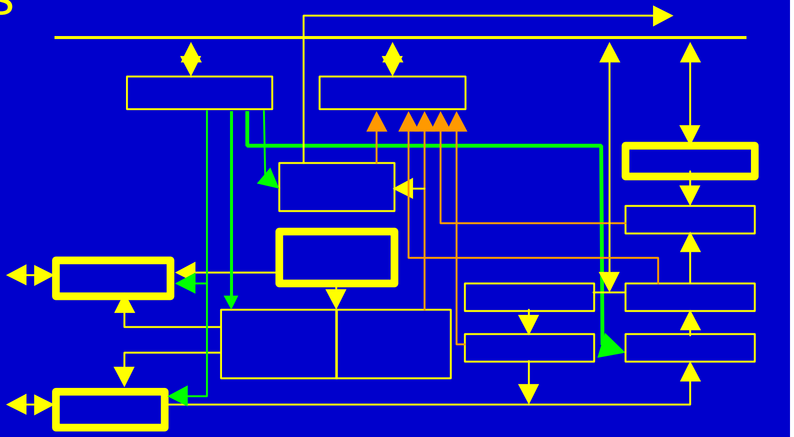
- Unlike USB, no common VHDL
- Implementations
  - I^{2}C Based
  - ASIC (State Machine)
  - MCU Firmware
  - SMBus Specific Designs
- Diversified implementations increase the possibility of incompatibilities

SMBus Interface using an I²C Port



I^{2}C Architectural Shortcomings

- ACK/NACK Preset
  - Inflexibility in issuing Resends
- Single Address Register
  - Lack of Support for General Address Calls
- Assumes MCU Clock Running (Wait Mode)
  - Increased Power Consumption
  - Not a big problem in specific applications, if secondary low frequency clock is available



Other I^2

- Logic Levels Incompatibility
  - Addressed with Level Shifters
  - May or may not be an issue, depending on the type of device
    - For SBS components it is a problem
    - For an SMBus host it may not pose a problem

Lower Power Consumption

- Ideally SMBus will operate without any MCU clock running
  - Important in Smart Battery
- Precludes the use of a watchdog timer
  - Alternative through periodic wake up mechanism either external or internal to

To ACK or NACK

- Issue of NACK during a message transaction is the only Resend method supported in SMBus
- NACK should be used
  - On invalid or improperly received
  - When data are not available or valid
- NACK should cause the Master to re-try

Multiple Address Support

- Allows implementation of General Call
- Enables creation of SMBus composite
  - Charger/Selector/Host

Agenda

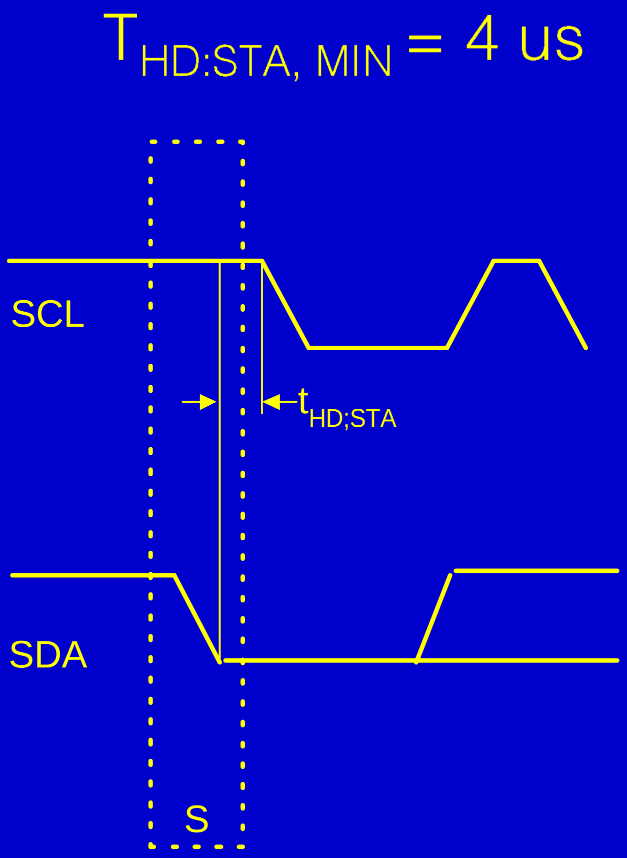
- SMBus current status
- Implementing an SMBus port
- Bus Failures
- Designing an SMBus system
- More Addresses
- Looking forward

Bus Failures

- Communication failures may occur because of hardware/firmware incompatibilities among SMBus
- The SMBus specification, if followed religiously, will eliminate most of the bus

Some Real Life Failure Examples

- Device A stretches the clock prior to 4us
- Every other device on the bus fails to recognize the START condition

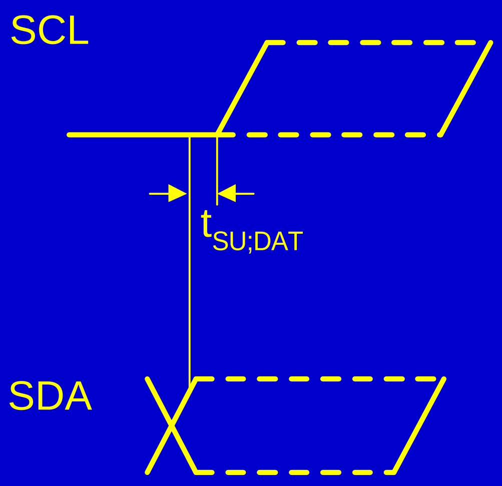


More Examples

- Master A changes both SCL & SDA lines simultaneously, causing an I^{2}C based device to fail
- Device B toggles the data line while SCL low (OK). Device C gets confused (Not OK)

Data Setup Time

$T_{SU::DAT,MIN} = 250$ ns



How to Avoid Causing Such Problems

- Use the SMBus Compatibility Checklist

  http://www.sbs-forum.org/mem/jmil/proposals.htm

- Still in proposal state
- Send your comments and provide your experience for improving it

Checklist Components

- Parameter under test definition
- Test objective
- Significance of compliance
- Test procedure description
- Test outcome
- Pass conditions

Checklist Usage

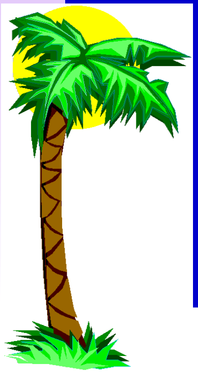
- Self check
- Trust it
  - Reflects a lot of development and troubleshooting misery
- Help your product
- Help the standard
- Help the market grow

Agenda

- SMBus current status
- Implementing an SMBus port
- Bus Failures
- Designing an SMBus system
- More Addresses
- Looking forward

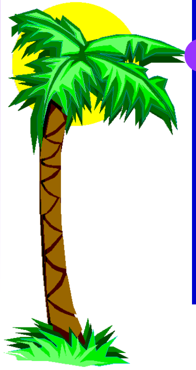
Designing with SMBus

- Use Multiple branches for safety and performance improvement
- Take advantage of the EC pre-processing capabilities
- Group together components that make
  - Example: Temperature Sensor and fan speed control (EC can handle policies)



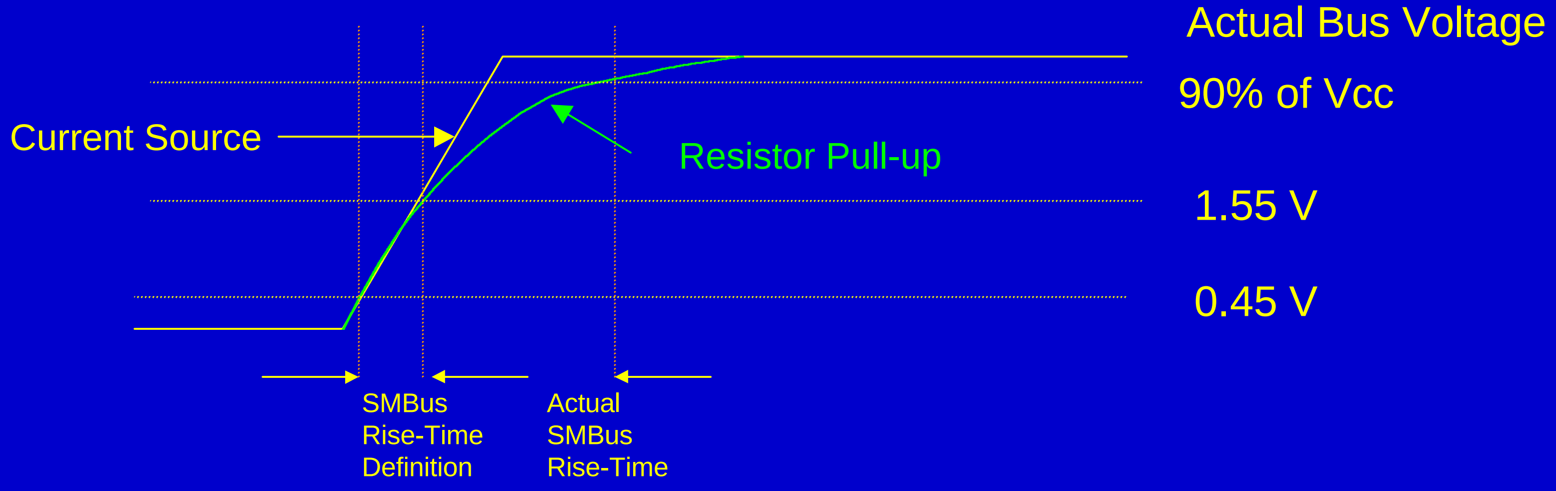
Putting the System Together

- Maximum Pull-up Current: 350 uA
  - Determined primarily from the current sinking capabilities of SMBus low power devices
- Maximum Rise Time: 1000 ns
  - Determined by need to achieve maximum bit transfer rate of 100Kbs, as 20% of clock high half period (4 us min)
- Apply to each individual SMBus branch



Rise Time Considerations

- Actual vs. Specified



Actual Bus Voltage

90% of Vcc

1.55 V

0.45 V

Current Source

Resistor Pull-up

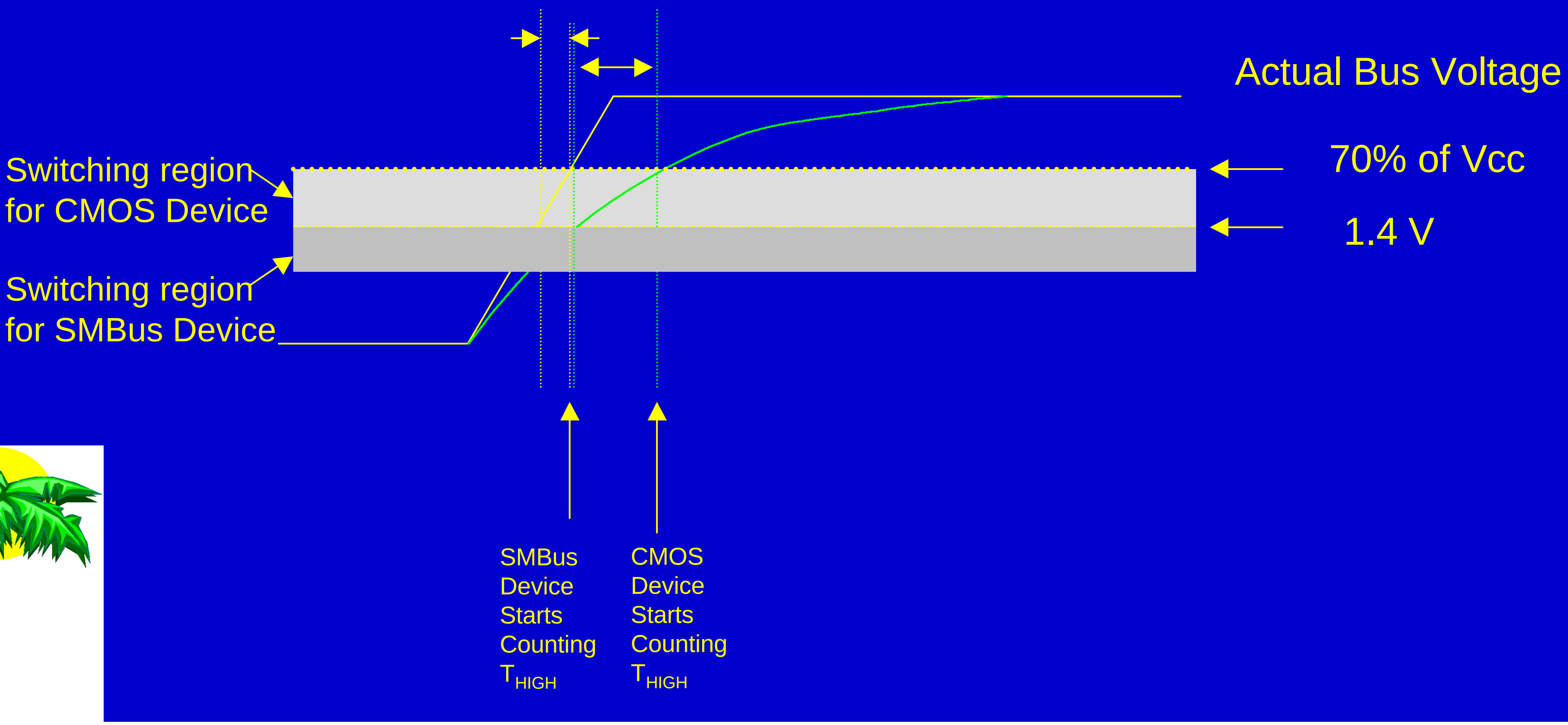
SMBus
Rise-Time
Definition

Actual
SMBus
Rise-Time

$$\text{Rise Time} = (V_{ILMAX} - 0.15) \text{ to } (V_{IHMIN} + 0.15)$$

Mixing SMBus with other CMOS devices

- Actual switching points


Actual Bus Voltage
70% of Vcc
1.4 V
Switching region for CMOS Device
Switching region for SMBus Device
SMBus Device Starts Counting T_{HIGH}
CMOS Device Starts Counting T_{HIGH}

Rise Time

- The Rise Time specification is challenging to meet when intermixing SMBus and CMOS (host usually) devices
- If not properly taken care off it will result in data loss on the bus
- Current sources instead of resistor pull-ups

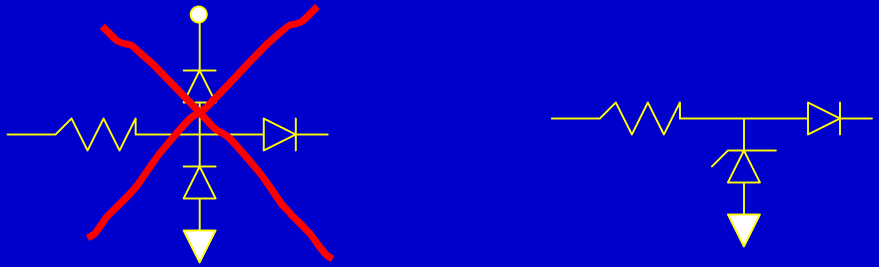
- Lower bus voltages ease the problem

Rise Time (continued)

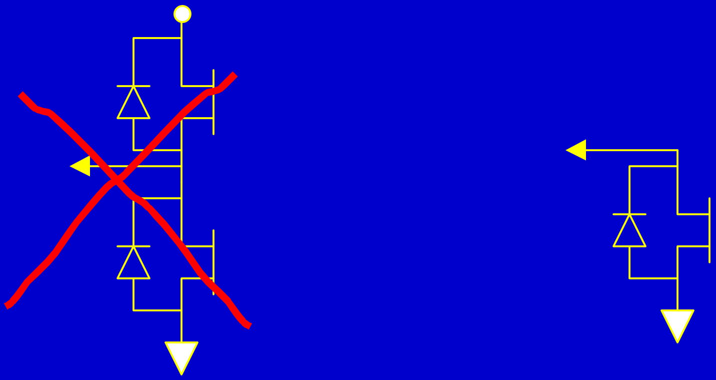
- In a 5 Volt bus, with a current source the total bus capacity can be as low as 87
- In a 3 Volt bus the total capacity can be as (still too restrictive)
- The designer will have to trade off bus speed

Powering Down Bus Devices

Input Stage



Output Stage

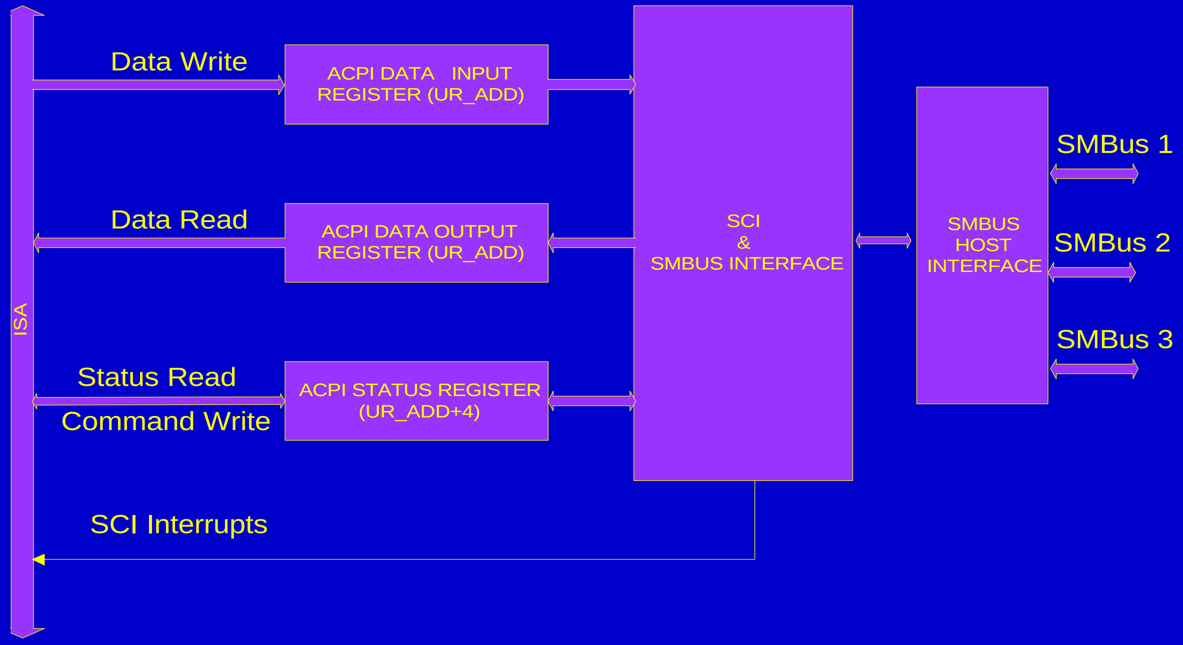


Agenda

- SMBus current status
- Implementing an SMBus port
- Bus Failures
- Designing an SMBus system
- More Addresses
- Looking forward

Address Contention

- Can be resolved by utilizing multiple



When to Ask for an Address

- If your device represents a generic
- If you need standardized support in the
- If you plan to open your specifications to

When Not to Bother

- If your device is particular to specific(s)
- If you don't need standardized OS
- Then use dedicated SMBus branches and/or the open SMBus addresses

Looking Forward

- More devices and applications
- Resolution on SMBus address contention problem
  - Join the SMBus Working Group