Error Mitigation of Point-to-Point Communication for Fault-Tolerant Computing

Authors: Robert L Akamine, Robert F. Hodson, Brock J. LaMeres, and Robert E. Ray
Contents

• Introduction to the Problem
• Solution Strategies
• Advantages and Disadvantages to each Strategy
• Suggested Applications
Introduction to the Problem

• How to detect and recover from physical damage or ‘glitches’ between two communication points

• Physical damage can be caused by:
  - Impact from debris
  - Damage from use

• ‘Glitches’ refer to Single Event Upsets (SEU) and Single Event Transients (SET)

• Important problem for space and military applications

• Even more important since Commercial Off The Shelf (COTS) parts are being used in more rugged environments
  - COTS parts have higher performance and cost less than radiation hardened parts
Solution Strategies

- Two main types of methods detect and recover from disrupted or disabled communication lines
  
  Masking – Detecting, isolating, and recovering from errors in real-time
  
  Non-masking – Errors are detected in real-time. However, communication needs to halt to isolate and recover from errors

- Masking example
  
  Forward Error Correction (FEC) codes and Triple Modular Redundancy (TMR)

- Non-masking example
  
  Automatic Repeat Query (ARQ) – A command and response protocol. The transmitter sends data and expects an acknowledgement from the receiver
  
  Uses parity, Cyclic Redundancy Checks (CRC), or other Error Detection Codes (EDC) to detect errors
Solution Strategies

Communication Error Detection and Correction Techniques

Masking
- Parallel
  - TMR
- Serial
  - FEC

Non Masking
- Parallel
  - TMR
- Serial
  - ARQ
Solution Strategies - FEC

- Montana State Hamming code demonstration
- Uses Hamming codes to detect and correct a single bit error and detect two errors
  
  Limits Hamming to working with only one error condition (physical damage or SEU)
- Included an ARQ type protocol to switch from damaged to undamaged wires
- Meant to emulate communication between ROM and a microcontroller
- Demonstration implemented on Xilinx ML-505 Virtex-5 FPGA development boards
Solution Strategies - TMR

- TMR is a simple masking technique where the transmitter replicates the same data three times and the receiver votes to decide what is valid data
- The probability of two SEU’s causing erroneous data to exit the voter is very low
- The probability of two damaged wires affecting the same three bit cluster is low
- Demonstration implemented on Xilinx ML-505 Virtex-5 FPGA development boards
Solution Strategies – ARQ with Parity

- Errors are detected with simple parity
- Data is sent with a parity bit and an additional sequence bit from the transmitter to the receiver
  
  The sequence bit allows the receiver to know if data was repeated.
- The transmitter expects an acknowledgement from the receiver in order to continue communication
- In the case of a single case of bad parity cause by an SEU data is retransmitted
- If damaged wired are detected the affected data, parity, or sequence bit is rerouted to spare wires
Solution Strategies – ARQ with Parity
Solution Strategies – Ethernet ARQ

- A serialized example that demonstrates an ARQ
- Ethernet packets by definition include a 32 bit CRC check also called a FCS
- Multiple physical links are required to account for damages wires
- Each data packet an acknowledgement packet

  If the receiver encounters a bad FCS then it requests a resend of the last data packet

  If no response packet is received by the transmitter then serial wires (TX or RX) must be damaged. The system switches to a backup physical link.
Solution Strategies – Ethernet ARQ

- Send Frame
- Resend Frame
- Receive Frame (from Ethernet Physical Layer)
  - Frame Valid
    - Yes: Send ‘Good’ Response Packet on Both Physical Layers
    - No: Send ‘Bad’ Response Packet on Both Physical Layers
- Two Response Packets Received?
  - Yes: At Least One Valid Response?
    - Yes: Resend Frame Requested?
      - Yes: Resend Frame
      - No: Switch to Backup TX Pair
    - No: One Response Packet Received?
      - Yes: Response Packet Valid?
        - Yes: Resend Frame Requested?
          - Yes: Resend Frame
          - No: Request Response Packet
        - No: Resend Frame
      - No: Resend Frame
  - No: One Response Packet Received?
    - Yes: Response Packet Valid?
      - Yes: Resend Frame Requested?
        - Yes: Resend Frame
        - No: Request Response Packet
      - No: Resend Frame
    - No: Resend Frame
## Advantages and Disadvantages to each Strategy

<table>
<thead>
<tr>
<th>Technology</th>
<th>Channel Efficiency</th>
<th>Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallel</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEC</td>
<td>Average</td>
<td>Average</td>
</tr>
<tr>
<td>TMR</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>ARQ</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td><strong>Serial</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMR</td>
<td>Good</td>
<td>Average</td>
</tr>
<tr>
<td>ARQ</td>
<td>Average</td>
<td>Good</td>
</tr>
</tbody>
</table>

- **FEC**
  - Generally speaking the channel efficiency \(\text{data_bits}/\text{total_bits_with_error_detection}\) of FECs increase with the number of data bits used. In the case of Hamming codes 8 data bits require 4 parity bits. In this case the channel efficiency is 67%.

- **TMR**
  - The channel efficiency of any TMR system is 33%.

- **ARQ**
  - As with hamming codes the channel efficiency increases with the number of data bits. Assuming one spare wire the channel efficiency of a system with 8 data bits is 73%.

- **Serial**
  - The channel efficiency of serial system is a measure on how well the system utilizes the link \(\text{data_time}/\text{total_time}\). Assuming an ethernet system the channel efficiency is 98%.

- **ARQ**
  - Since a response packet (assuming a minimum size packet) is required for each data packet under the best conditions the channel efficiency is around 94%.

- **Mass**
  - The mass is a function of the channel efficiency. Parallel FEC systems have less mass than TMR systems and fair well with ARQ systems as the number of data wires increases.

- The mass of a TMR system is always higher than an ARQ system with one spare link.
### Advantages and Disadvantages to each Strategy

<table>
<thead>
<tr>
<th>Technology</th>
<th>Latency</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallel</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEC</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>TMR</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>ARQ</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

Since FEC systems detect and correct errors without interaction from the source or sink the latency is low.

All parallel systems suffer from intersymbol interference, noise, and transmission line properties.

The additional command and response protocol increases the latency of by at least a factor of two when compared to FEC designs.

**Serial**

| TMR        | Low     | High   |
| ARQ        | High    | High   |

Since the receiver votes on the content of three packets and does not need to send an acknowledgment packet the latency is low.

Serial technology has the ability to propagate efficiency over considerably longer cable lengths.

The data and response packet protocol increases the latency.
## Suggested Applications

<table>
<thead>
<tr>
<th>Technology</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parallel</strong></td>
<td></td>
</tr>
<tr>
<td>FEC</td>
<td>Communication within the same Printed Circuit Board (PCB)</td>
</tr>
<tr>
<td>TMR</td>
<td></td>
</tr>
<tr>
<td>ARQ</td>
<td></td>
</tr>
<tr>
<td><strong>Serial</strong></td>
<td></td>
</tr>
<tr>
<td>TMR</td>
<td>Long range communication where mass is not a concern and the most bandwidth is needed</td>
</tr>
<tr>
<td>ARQ</td>
<td>Long range communication where low mass is required</td>
</tr>
</tbody>
</table>