Error Correction techniques on NoC protocol layers

Ahmed Garamoun

Supervisor: Dipl.-Inf. Adan Kohler
Agenda

1. Motivation
2. Analysis of Error recovery schemes
3. Error Control Policies
4. Conclusion
1. Motivation

- Technology Scaling
  - Integration of more IP cores
    - Motivates usage of NoCs as interconnects
  - Unreliable wires
    - Increasingly susceptible to various noise sources
    - Error rates increases drastically
  - Error recovery for NoCs
    - Powerful Error recovery schemes
    - Low power consumption

Intel TERAFLOP 80 cores with 2D mesh NoC
Agenda

1. Motivation

2. Analysis of Error recovery schemes

3. Error Control Policy

4. Conclusion
2. Analysis of Error Recovery Schemes

2.1. Retransmission Scheme

2.2. Error Correcting Scheme

2.3. Hybrid scheme
2.1 Retransmission Scheme

- Receiver checks the received data
- Send ACK/NACK to the sender
  - ACK, data received correctly
  - NACK, send the data again
- Low area overhead $\rightarrow$ power efficient
- Time needed for retransmission
2.1 Retransmission Scheme

- Totally fails if permanent errors occur
- Possible solutions:
  - Sending the flits through two different routes → Double the network load
  - Usage of **Error Correction codes**
2. Analysis of Error Recovery Schemes

2.1. Retransmission Scheme

2.2. Error Correcting Scheme

2.3. Hybrid scheme
2.2. Error Correction Scheme

- Transient errors
  - Crosstalk, Electromagnetic interference, Alpha particle hits, Cosmic radiation

- Why crosstalk?
  - Increase the probability of errors
  - Large propagation delay
  - Increase power consumption

- Goal
  - Minimize crosstalk effect
  - Provide error correction
2.2. Error Correction Scheme

- Crosstalk understanding
- Crosstalk Avoidance Single Error Correction codes
  - DAP
  - MDR
  - BSC
- Crosstalk Avoidance Multiple Error Correction codes
  - CADEC
  - JTEC
  - Optimized JTEC
- Power consideration
2.2. Error Correction Scheme

- Crosstalk: unwanted coupling of energy between two adjacent wires
  - $C_{\text{eff}}$ is the resultant capacitance on the victim wire
  - $C_{\text{eff}}$ depends on the change of voltage between victim and aggressors

<table>
<thead>
<tr>
<th>$C_{\text{eff}}$</th>
<th>Transition Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $C_s$</td>
<td>$(\uparrow,\uparrow,\uparrow)$ $(\downarrow,\downarrow,\downarrow)$</td>
</tr>
<tr>
<td>2. $C_s + C_c$</td>
<td>$(-,\uparrow,\uparrow)$ $(-,\downarrow,\downarrow)$ $(\uparrow,\uparrow,-)$ $(\downarrow,\downarrow,-)$</td>
</tr>
<tr>
<td>3. $C_s + 2C_c$</td>
<td>$(-,\uparrow,-)$ $(-,\downarrow,-)$ $(\uparrow,\uparrow,\downarrow)$ $(\uparrow,\downarrow,\uparrow)$ $(\downarrow,\downarrow,\uparrow)$</td>
</tr>
<tr>
<td>4. $C_s + 3C_c$</td>
<td>$(-,\uparrow,\downarrow)$ $(-,\downarrow,\uparrow)$ $(\uparrow,\downarrow,-)$ $(\downarrow,\uparrow,-)$</td>
</tr>
<tr>
<td>5. $C_s + 4C_c$</td>
<td>$(\uparrow,\downarrow,\uparrow)$ $(\downarrow,\uparrow,\downarrow)$</td>
</tr>
</tbody>
</table>

↑: positive transition
↓: negative transition
–: no transition

$T$ (Thickness)
$L$ (Length)
$H$ (Height)
$S$ (Spacing)
$A_1$
$A_2$
$V$
$C_c$
$C_s$
2.2. Error Correction Scheme

- The effect of crosstalk on the signal of victim wire
  - Positive Glitch
  - Negative Glitch
  - Falling Delay
  - Rising Delay

<table>
<thead>
<tr>
<th>Link Lines</th>
<th>Fault Effect</th>
<th>Positive Glitch</th>
<th>Negative Glitch</th>
<th>Falling Delay</th>
<th>Rising Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>link line 1 (aggressor)</td>
<td></td>
<td><img src="image1" alt="positiveglitch" /></td>
<td><img src="image2" alt="negativeglitch" /></td>
<td><img src="image3" alt="fallingdelay" /></td>
<td><img src="image4" alt="risingdelay" /></td>
</tr>
<tr>
<td>link line i-1 (aggressor)</td>
<td></td>
<td><img src="image1" alt="positiveglitch" /></td>
<td><img src="image2" alt="negativeglitch" /></td>
<td><img src="image3" alt="fallingdelay" /></td>
<td><img src="image4" alt="risingdelay" /></td>
</tr>
<tr>
<td>link line i (victim)</td>
<td><img src="image1" alt="positiveglitch" /></td>
<td><img src="image2" alt="negativeglitch" /></td>
<td><img src="image3" alt="fallingdelay" /></td>
<td><img src="image4" alt="risingdelay" /></td>
<td></td>
</tr>
<tr>
<td>link line i+1 (aggressor)</td>
<td><img src="image1" alt="positiveglitch" /></td>
<td><img src="image2" alt="negativeglitch" /></td>
<td><img src="image3" alt="fallingdelay" /></td>
<td><img src="image4" alt="risingdelay" /></td>
<td></td>
</tr>
<tr>
<td>link line n (aggressor)</td>
<td><img src="image1" alt="positiveglitch" /></td>
<td><img src="image2" alt="negativeglitch" /></td>
<td><img src="image3" alt="fallingdelay" /></td>
<td><img src="image4" alt="risingdelay" /></td>
<td></td>
</tr>
</tbody>
</table>

- expected behavior
- faulty behavior
2.2. Error Correction Scheme

- Inject 0 after each bit
  - **01010** → **0010001000**
  - **10110** → **100101000**
  - No additional wires, but data rate is halved

- Duplication of the data wires
  - **01010**
    - **10110**  ([↑,↓,↑]) fifth case, worst case
    - 0011001100
    - 1100111100  ([↑,↓,↓]) third case
  - Hamming distance is doubled
2.2. Error Correction Scheme

- Crosstalk understanding

- **Crosstalk Avoidance Single Error Correction codes**
  - DAP
  - MDR
  - BSC

- Crosstalk Avoidance Multiple Error Correction codes
  - CADEC
  - JTEC
  - Optimized JTEC

- Power consideration
2.2. Error Correction Scheme

- Duplicate Add Parity (DAP)
  - Encoder
    - Duplicate the data wires, add parity bit
  - Can be improved by intelligent wire routing during design time

<table>
<thead>
<tr>
<th>Data word</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>100110000</td>
</tr>
<tr>
<td>0011</td>
<td>000001111</td>
</tr>
<tr>
<td>1010</td>
<td>111001100</td>
</tr>
</tbody>
</table>
2.2. Error Correction Scheme

- Duplicate Add Parity (DAP)
  - Decoder

```
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
```

```
\begin{align*}
&= 1 \\
&= 1 \\
&= 1 \\
&= 1 \\
&= 1 \\
&= 1 \\
&= 1 \\
&= 1 \\
\end{align*}
```
2.2. Error Correction Scheme

- **Modified Dual Rail (MDR)**
  - Basically the same DAP principle
  - Duplicate the parity bit to reduce its crosstalk effect

<table>
<thead>
<tr>
<th>Data word</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>1100110000</td>
</tr>
<tr>
<td>0011</td>
<td>0000001111</td>
</tr>
<tr>
<td>1010</td>
<td>1111001100</td>
</tr>
</tbody>
</table>
2.2. Error Correction Scheme

- Crosstalk understanding

- **Crosstalk Avoidance Single Error Correction codes**
  - DAP
  - MDR
  - BSC

- Crosstalk Avoidance Multiple Error Correction codes
  - CADEC
  - JTEC
  - Optimized JTEC

- Power consideration
2.2. Error Correction Scheme

- **Boundary Shift Code (BSC)**
  - *Invalid transition*  \(01\overline{1}010\rightarrow01\overline{0}1100\)
    - Code is “self shielding” if it avoids invalid transition
  - *Dependent boundary*  \(01\overline{1}0\overline{0}0\overline{1}\rightarrow\{1, 3, 7\}\)
    - 1-bit circular right shift convert code with only even dependent boundary to code with only odd dependent boundary
      
      \(1\overline{1}00\overline{1}00\rightarrow\{2, 4, 6\}\) circular right shift

      \(01\overline{1}00\overline{1}10\rightarrow\{1, 3, 5\}\)

  - No overlapping dependent boundaries between two consecutive codewords \(\rightarrow\) No invalid transition
2.2. Error Correction Scheme

- **Boundary Shift Code (BSC)**

  - Code with no Odd dependant boundaries
  - 1-bit circular right-shift
  - Code with no Even dependant boundaries
  - Alternating between codes gives self-shielding code

- Duplication generates only even dependant boundary code
2.2. Error Correction Scheme

- Boundary Shift Code (BSC)

  - BSC Encoder

<table>
<thead>
<tr>
<th>Clock</th>
<th>Data word</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0100</td>
<td>100110000</td>
</tr>
<tr>
<td>2</td>
<td>0011</td>
<td>000011110</td>
</tr>
<tr>
<td>3</td>
<td>1010</td>
<td>111001100</td>
</tr>
</tbody>
</table>

Clock Data word Codeword

1 0100 100110000
2 0011 000011110
3 1010 111001100
2.2. Error Correction Scheme

- **Boundary Shift Code (BSC)**
  - BSC Decoder
2.2. Error Correction Scheme

- Crosstalk understanding
- Crosstalk Avoidance Single Error Correction codes
  - DAP
  - MDR
  - BSC
- Crosstalk Avoidance Multiple Error Correction codes
  - CADEC
  - JTEC
  - Optimized JTEC
- Power consideration
2.2. Error Correction Scheme

- Crosstalk Avoidance Double Error Correcting (CADEC) code
  - Combination of DAP with SEC Hamming code
  - Hamming distance of $7 = (2 \times 3) + 1$
    - duplication of Hamming code $2 \times 3 = 6$
  - Parity bit
- CADEC encoder
2.2. Error Correction Scheme

- Crosstalk Avoidance Double Error Correcting (CADEC) code
  - CADEC decoder

<table>
<thead>
<tr>
<th>Case</th>
<th>Errors in A</th>
<th>Errors in B</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Error Correction Scheme:

- Start
  - Separate A, B, P0
  - Compute parity PA for A and PB for B
  - Choose A
    - Perform (38,32) Hamming Decoding
    - Output Decoded
  - Choose B
    - If B error free
      - Send B for syndrome detection
    - If B error
      - Perform (38,32) Hamming Decoding
      - Output Decoded

Cases:
- Case 1, 4, 5, 6
- Case 2, 3
- Case 4, 6
- Case 1, 5

Cases:
- Case 2
- Case 3
2.2. Error Correction Scheme

- Crosstalk Avoidance Double Error Correcting (CADEC) code
  - CADEC decoder Circuit implementation
2.2. Error Correction Scheme

- (Optimized) Joint crosstalk avoidance and Triple Error Correction codes (JTEC) code
  - CADEC encoder is used „Minimum Hamming distance of 7“
  - Always one copy of the data is error-free or has only 1-error

<table>
<thead>
<tr>
<th>Errors in A</th>
<th>Errors in B</th>
<th>Total number of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

- More information is discussed in the report
2.2. Error Correction Scheme

- Crosstalk understanding

- Crosstalk Avoidance Single Error Correction codes
  - DAP
  - MDR
  - BSC
  - DAPBI

- Crosstalk Avoidance Multiple Error Correction codes
  - CADEC
  - JTEC
  - Optimized JTEC

- Power consideration
2.2. Error Correction Scheme

- Power consideration

- Error Correction Capability
  - high area and power overhead for the Encoder & Decoder
  - complex error correction mechanism
  - Error probability increases
  - Error correction capability

- Power
  - decrease the voltage
  - decrease the power dissipation
  - noise margin is increased
  - Encoder & Decoder
  - increase the power dissipation
  - Power consideration
2.2. Error Correction Scheme

- Power consideration
  - Voltage increases with error probability decreases
  - Power for different coding schemes at certain error probability
    - butterfly-fat-tree (BFT) based NoC architecture with 64 IP blocks
2.3 Hybrid Scheme

- Combination of retransmission and Error correction schemes
  - Decoder tries to recover the error
  - If the number of error more than correction code capability, the receiver send NACK to the sender to retransmit the data

- Fails if there are more errors in the header flit than can be corrected by the code → the packet is dropped
Agenda

1. Motivation
2. Analysis of Error recovery schemes
3. Error Control Policy
4. Conclusion
3. Error Control Policy

- End-to-End Error Control Policy
- Switch-to-Switch Error Control Policy
3. Error Control Policy

- End-to-End Error Control Policy
3. Error Control Policy

- **End-to-End Error Control Policy**
  - Switch Architecture for End-to-End
  - Area overhead:
    - 1 Encoder
    - 1 Decoder
    - Extend the I/p & O/p Buffers to codeword width
  - Additional Delay
    \[ T_{\text{encoder}} + T_{\text{decoder}} \]

![Diagram of Crossbar & Router with Encoder and Decoder modules connected by I/p and O/p buffers.](image)
3. Error Control Policy

- End-to-End Error Control Policy
- Switch-to-Switch Error Control Policy
3. Error Control Policy

- **Switch-to-Switch Error Control Policy**

Error Correction techniques on NoC protocol layers
3. Error Control Policy

- **Switch to Switch Error Control Policy**
  - Switch Architecture for Switch-to-Switch
  - Area overhead:
    - 4 Encoder
    - 4 Decoder
    - Extend the o/p Buffers to codeword width
  - Additional Delay
    \[ = N(T_{\text{encoder}} + T_{\text{decoder}}) \]
    where \( N \) is the number of hops
1. Motivation
2. Analysis of Error recovery schemes
3. Error Control Policy
4. Conclusion
4. Conclusion

- Error recovery Schemes
- Error control policies
- Tradeoff
  - Correction capability
  - Performance
  - Power consumption
  - Area overhead
- No optimal solution in one recovery scheme
- No optimal solution in one layer but multiple layers should be taken into account
Thank you for your attention!