Self-calibrating Asynchronous NoC Links

Mathankumar Gopalsamy

Supervisor: Dipl.-Inf. Stefan Holst
1. Motivation
2. Asynchronous NoC
3. Calibrating asynchronous links
4. Calibrating upstream logic
5. Summary
1 Motivation

- Effect of Process Variations
  - Difficult to control nanoscale processes
  - Reduces operational margins
  - Yield reduction

- Increased Energy Consumption

- Poor Reliability
  - Metastability and data losses
  - Environmental variations
  - Ageing of components
Overview

1. Motivation

2. Asynchronous NoC

4. Calibrating asynchronous links

5. Calibrating upstream logic

6. Summary
GALS Design

- Clock distribution consumes ~ 40%-70% of total energy in NoC
- Global clock for all modules leads to sub-optimal solution
- Clock skew control is becoming arduous
- Globally Asynchronous Locally Synchronous (GALS) System

- Asynchronous
- Mesochronous
Clock Domain Crossing

- Clock Domain
  - Part of the design driven by same clock or clocks with constant phase relationships

- Clock Domain Crossing (CDC)
  - Signals that interface between asynchronous clock domains

Same Clock Domain

Different Clock Domain
Metastability

CLK A

D_{CD}

CLK B

D_{OU}

CLK B samples incoming data while it is changing
Metastability

Synchronizer

Metastability Detector
Data loss

CLK A

D_{CD}

CLK B

D_{OU}

CLK B slower and leads CLK A

Domain 1

Domain 2

FF

FF

D_{IN}

D_{CDC}

D_{OUT}
Overview

1. Motivation

2. Asynchronous NoC

3. Calibrating asynchronous links

4. Calibrating upstream logic

5. Summary
3 Calibrating asynchronous links

- Performance properties are not clearly known at design time
- Links consume 40% to 50% of total energy
- Delay variations affects performance
- Autonomous selection of operating point i.e. Voltage – Frequency pair
**Worst-case Vs Self-calibrating design**

**Worst-case Design**: \( O_W(V_{DD}, d_1) \)

**After Post-silicon testing**

**Chip 1:**
Operating Point: \( O_1(V_1, d_1) \)

**Energy waste**

**Chip 2:**
Operating point: \( O_2(V_2, d_2) \)

**Timing errors, Yield loss**
Delay variations

After design

- A
- B
- C
- $T_{CP} = 2\text{ns}$

After post-silicon testing

- A
- B
- C
- $T_{CP} = 3\text{ns}$

Stage becomes slower
Stage becomes faster

Calibrating delay variations using cycle time stealing

- A
- B
- C
- $T_{CP} = 2\text{ns}$

B steals time from faster stage

Self-calibrating Asynchronous NoC Links
ReCycle Technique

Total Skew of a stage

\[ T_{\text{skew}} = \delta_f - \delta_i \]
Self-Calibrating interconnect

Classical asynchronous link

Self-calibrating asynchronous link

Self-calibrating Asynchronous NoC Links
**Synchronizing Coding**

**Encoder and Decoder**

- Locally generated flipping bit
- CRC-8 encoding
Self-calibrating Asynchronous NoC Links

Operating Point Controller (OPC)

- Chooses the adaptive operation point \((V_{ch}, F_{ch})\)
- Implemented in hardware
- Modes
  - *Normal* mode
  - *Explore* mode
- Memory table contains operation points
- Register *counter* for calibration periodicity
### Operating Point Controller Algorithm

**Examples**

- **Case 1**
  - Reset, no error
- **Case 2**
  - Explore, no error
- **Case 3**
  - Explore, error
- **Case 4**
  - Normal, no error

---

**Self-calibrating Asynchronous NoC Links**
Overview

1. Motivation
2. Asynchronous NoC
3. Calibrating asynchronous links

4. Calibrating upstream logic

5. Summary
4 Calibrating upstream unit

- What happens if input to asynchronous link already suffers from delay variation?
- Last latch element of upstream unit is replaced with a special register called „Self-correcting flip-flop“ (SCFF)
- Pipeline model with upstream logic and link as two stages
Basic idea: **Delayed sampling**  
**Dynamic voltage-swing scaling**

### Construction

**Operating Point Control table**

<table>
<thead>
<tr>
<th>S2</th>
<th>S1</th>
<th>MAIN</th>
<th>Selected FF</th>
<th>Skew Selector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>MAIN</td>
<td>0 0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>S1</td>
<td>0 1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>S2</td>
<td>1 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>S3</td>
<td>1 1</td>
</tr>
</tbody>
</table>
Operating Point Controller

Metastability and Error detector

Skew selector generation

Selection Table

<table>
<thead>
<tr>
<th>OUT1 (MAIN)</th>
<th>OUT2 (S1)</th>
<th>OUT3 (S2)</th>
<th>SEL 1</th>
<th>SEL2</th>
<th>Selected FF</th>
<th>Swing voltage (example)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>MAIN</td>
<td>200 mV</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>S1</td>
<td>400 mV</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>S2</td>
<td>700 mV</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>S3</td>
<td>1000 mV</td>
</tr>
</tbody>
</table>
Modes of operation

- **Calibration Mode**
  - All flip-flops are operational
  - Entered after post-silicon testing to calibrate the link to process variations
  - Can be entered periodically to test and tolerate variations due to environmental factor and ageing.

- **Normal Mode**
  - All flip-flops except the one selected during calibration mode are turned off to save power
Overview

1. Motivation
2. asynchronous NoC
3. Calibrating asynchronous links
4. Calibrating upstream logic

5. Summary
Summary

- Need for asynchronous NoCs
- Worst-case design is waste of resources
- Self-calibrating design improves yield
- ANoC Link and upstream calibration applies
  - Dynamic voltage swing scaling
- The gain attained in reliability and energy consumption
- Two techniques presented can be used in combination
Thank you!