Multi-synchronous systems

George Taylor, Robert Mullins, Simon Moore

- multi-synchronous / GALS systems
  - potential for energy reduction
  - mixed IP blocks, fixed I/O interface frequencies
- synchronisation
  - when, where, cost: latency, MTBF
  - time safe or value safe, different failure modes
- communication
  - fixed wiring (e.g. point-to-point, bus)
  - on-chip network (virtual channels, packet switching)

<table>
<thead>
<tr>
<th>name</th>
<th>frequency</th>
<th>phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>mesochronous</td>
<td>same</td>
<td>constant</td>
</tr>
<tr>
<td>plesiochronous</td>
<td>small difference</td>
<td>slowly varying</td>
</tr>
<tr>
<td>derived (rational)</td>
<td>fixed simple ratio</td>
<td>constant</td>
</tr>
<tr>
<td>heterochronous</td>
<td>different</td>
<td>constant/varying</td>
</tr>
<tr>
<td>asynchronous</td>
<td>non-periodic</td>
<td>unknown</td>
</tr>
</tbody>
</table>

- desirable to hide synchronisation latency
  - straightforward for meso- & plesiochronous
    (similar to dynamic clock de-skewing)
  - non-trivial or impractical for heterochronous
  - impossible for asynchronous
Synchronising asynchronous events

- **time safe** (metastability MTBF)
  - use latency to reduce MTBF
    - pipeline flow control to keep bandwidth
  - 2xDFF’s
    - every bit of every data word individually synchronised
  - FIFO synchroniser
    - synchronisation needed when full/empty flags change
    - steady state avoids synchronisation failure and latency
  - pipeline synchroniser (Seizovic)
    - ‘synchronicity’ of data increased as it traverses pipeline
    - synchronisation only on bundled control signals
- **value safe** (bounded time MTBF)
  - clock stretching (on-chip local clock oscillator)
    - potential low latency synchronisation for first data item
    - possibility of FIFO sync. behaviour for subsequent items

On-chip network

- **virtual circuits**
  - meso-, plesio-, hetero-chronous schemes require repeatable communication latency
    - perform synchronisation when not communicating
    - schedule communications to use same route
    - problem: need distributed time reference to deploy schedule
- **asynchronous packet switch network**
  - synchronisation latency unavoidable
  - use well designed synchroniser
  - reduce number of synchronisations
    - e.g. high bandwidth packet ‘burst’:
      - receiver only needs to synchronise on ‘newpacket’ signal
      - transmitter must be able to dump whole packet into network in one burst