GALS Baseband Processor for WLAN

Miloš Krstić, Eckhard Grass

IHP
Im Technologiepark 25
15236 Frankfurt (Oder)
Germany
Overview

• Motivation

• Concept of the request-driven GALS technique

• Baseband processing compliant to IEEE 802.11a standard

• GALSification of the baseband processor

• Testing strategy

• Design-flow and implementation

• Conclusion
Motivation

- Practical application of the request driven GALS technique in complex communication system

- First goal was to decouple complex digital blocks from the common clock source in order to avoid clock-skew problems

- Aim is to achieve high data throughput with low latency

- Another issue is avoiding of unnecessary transitions (clock cycles) during data transfer and minimization of the power consumption

- Lowering of the EMI and crosstalk in the mixed-signal environment has also great importance
Request-Driven GALS

• The target is to cover point-to-point communication with very intensive and bursty data transfer

• For a quasi-synchronous operation, GALS blocks can operate in a request-driven mode

• When there is no input activity, the data stored inside the locally synchronous pipeline has to be flushed out

• Then a local clock generator should drive the GALS blocks

• A Time-out function is proposed to control the transition from request driven operation to local clock generation mode
**Request-driven Asynchronous Wrapper**

Locally synchronous block is decoupled from the other blocks with the asynchronous wrapper.

Asynchronous wrapper consists of input & output port, time-out detector, and local-clock generation.

- **Locally Synchronous Block**
  - Input Port
  - Output Port
  - Time-out Detection
  - Local Clock Generation

- **Asynchronous Wrapper**
  - Input Port
  - Output Port
  - Handshake signals
  - Request driven clock
  - Locally generated clock

Data flows through the locally synchronous block, which is decoupled by the asynchronous wrapper. The wrapper consists of input/output ports, time-out detector, and local-clock generation modules.
Potential Gain from Proposed GALS Technique

• The circuit immediately responds to input requests and unnecessary delays are avoided

• Reliable and fast transfer of large bursts of data is achieved. Data transfer is possible at every clock cycle of synchronous block

• This proposed architecture offers an efficient power-saving mechanism, similar to clock gating

• Additionally, EMI (and crosstalk) should be reduced due to varying delays and frequencies in different asynchronous wrappers
Validation of the GALS concept

- Experimental GALS chip has been fabricated and tested

- Purpose of the chip was to confirm the feasibility of the proposed GALS strategy and BIST testing
Baseband Processor for IEEE 802.11a Standard

• The goal of our project is to develop a single-chip wireless broadband communication system in the 5 GHz band, compliant with the IEEE802.11a standard.

• This standard specifies a system using Orthogonal Frequency Division Multiplexing (OFDM) with data rates ranging from 6 to 54 Mbit/s.

• The baseband processor was initially implemented as an ASIC (complexity around 700k gates) working synchronously.

• Design of the baseband processor involves a number of challenges as:
  global clock tree generation, power consumption, testability, EMI and crosstalk.

• Our request-driven GALS architecture was developed as a possible solution for those problems.
Structure of the Baseband Processor

- Baseband processor includes receiver and transmitter datapath structure
- Very complex blocks are implemented such as Viterbi decoder, FFT, IFFT, CORDIC processors, ...
GALS Partitioning

- Partitioning is separately performed in the receiver and the transmitter.
- This process has taken into account possible power saving.
Additional Special Blocks Needed

- **Token rate adaptation blocks are developed**
  
  20 Msp → 80 Msp (in forward data transfer Rx_2 → Rx_3),
  and 80 Msp → 20 Msp (in backward data transfer Rx_3 → Rx_2)

- **Control of the asynchronous feedback loop is needed**
  
  Join circuit for the alignment of tokens entering block Rx_2 is crucial

- **Power saving mechanism in the baseband uses activation properties of the system and minimizes the switching**
Asynchronous-synchronous Communication

- Connection between a synchronous producer and an asynchronous consumer is achieved without additional circuitry.

- To connect the asynchronous producer with a synchronous consumer we have used pipeline synchronization.
Testing Strategy

• For testing the additional units based on Built-In Self-Test (BIST) were developed

• They allow test of individual blocks as well as whole system
Test Procedure

- For control of the testing a Central BIST Controller is used

- Test results are acquired over the single bit Test_OK
Design Flow

• We have used our in-house 0.25µ technology

• Asynchronous wrapper is equivalent to about 1.6 k inverter gates
  Only tunable clock generation is 1,1 k gates

• Asynchronous wrapper is operational up to about 110 Msps
  The current need is 80 Msps
### Area and Power Distribution

<table>
<thead>
<tr>
<th>Component</th>
<th>Cell area [mm², %]</th>
<th>Power [mW, %]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseband chip</td>
<td>45,68</td>
<td>151</td>
</tr>
<tr>
<td>Sync. blocks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rx_1 block</td>
<td>4.85</td>
<td>17.7</td>
</tr>
<tr>
<td>Rx_2 block</td>
<td>14.11</td>
<td>40.4</td>
</tr>
<tr>
<td>Rx_3 block</td>
<td>10.12</td>
<td>17.7</td>
</tr>
<tr>
<td>Rx_TRA</td>
<td>1.10</td>
<td>0.9</td>
</tr>
<tr>
<td>Tx_1 block</td>
<td>1.06</td>
<td>23.1</td>
</tr>
<tr>
<td>Tx_2 block</td>
<td>1.17</td>
<td>1.6</td>
</tr>
<tr>
<td>Tx_3 block</td>
<td>9.19</td>
<td>23.6</td>
</tr>
<tr>
<td>Asyn. blocks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW_Rx1</td>
<td>0.11</td>
<td>1.2</td>
</tr>
<tr>
<td>AW_RxTRA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AW_Rx2</td>
<td>0.13</td>
<td>1.2</td>
</tr>
<tr>
<td>AW_Rx3</td>
<td>0.13</td>
<td>0.6</td>
</tr>
<tr>
<td>AW.Tx2</td>
<td>0.13</td>
<td>0.3</td>
</tr>
<tr>
<td>AW.Tx3</td>
<td>0.28</td>
<td>0.7</td>
</tr>
<tr>
<td>Asyn. interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Join</td>
<td>0.03</td>
<td>0.2</td>
</tr>
<tr>
<td>FIFO_TA</td>
<td>0.70</td>
<td>0.0</td>
</tr>
<tr>
<td>As-Sy interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rx_int</td>
<td>0.60</td>
<td>6.4</td>
</tr>
<tr>
<td>Tx_int</td>
<td>0.36</td>
<td>7.1</td>
</tr>
<tr>
<td>BIST</td>
<td>1.63</td>
<td>6.6</td>
</tr>
<tr>
<td>CBC</td>
<td>0.11</td>
<td>1.5</td>
</tr>
<tr>
<td>TDE</td>
<td>1.23</td>
<td>2.1</td>
</tr>
<tr>
<td>TPG</td>
<td>0.29</td>
<td>3</td>
</tr>
</tbody>
</table>

- Area and power statistics are based on the synthesized netlist data. Synchronous blocks occupy more than 90% of the total area. The BIST circuitry requires around 3.6%, interface blocks 2,9% and asynchronous wrappers 2%.

- Based on the switching activities, in the realistic transceiver scenario, power estimation with Prime Power has been performed. Synchronous blocks are using most of the power (around 83%). Other important power consumers are async-to-sync interfaces with 13,5%, BIST 6,6% and asynchronous wrappers with 3%.
Final Results

- Our GALS baseband processor is taped-out and it is currently under fabrication.

- The total number of pins is 120 and the silicon area including pads is 45.1 mm\(^2\).

- After layout, the estimated power consumption is 324.6 mW.

The increase compared with the after-synthesis power estimation is due to the clock-tree and pad insertions.

Our synchronous baseband is consuming around 393 mW in the same transceiver application.
Conclusions

- GALSification of a complex IEEE 802.11a compliant WLAN baseband processor was presented

- GALS technique used here is based on a request-driven operation with additional local clock generation

- BIST is used as an effective testing technique

- It is shown that the GALS implementation consumes less power than a functionally equivalent synchronous version with clock gating