# Exploring Pausible Clocking Based GALS Design for 40-nm System Integration

Xin Fan[1], Miloš Krstić[1], Eckhard Grass[1], Birgit Sanders[2], and Christoph Heer[2]
[1]IHP Microelectronics, Im Technologiepark 25, Frankfurt (Oder), 15236, Germany
[2]Intel Mobile Communications, Neubiberg, Germany
{fan, krstic, grass}@ihp-microelectronics.com, {birgit.sanders, christoph.heer}@intel.com

*Abstract* — **Globally asynchronous locally synchronous (GALS) design has attracted intensive research attention during the last decade. Among the existing GALS design solutions, the pausible clocking scheme presents an elegant solution to address the cross-clock synchronization issues with low hardware overhead. This work explored the applications of pausible clocking scheme for area/power efficient GALS design. To alleviate the challenge of timing convergence at the system level, area and power balanced system partitioning was applied for GALS design. An optimized GALS design flow based on the pausible clocking scheme was further proposed. As a practical example, a synchronous/GALS OFDM baseband transmitter chip, named *Moonrake*, was then designed and fabricated using the 40-nm CMOS process. It is shown that, compared to the synchronous baseline design, 5% reduction in area and 6% saving in power can be achieved in the GALS counterpart.**

***Keywords - SoC, GALS, pausible clocking, OFDM***

## I. INTRODUCTION

Globally asynchronous locally synchronous (GALS) design presents an alternative to the traditional synchronous design for digital system integration [1]. In GALS design, data is processed in synchronous functional modules while transferred through asynchronous interfacing circuits. Instead of the global clock reference, each GALS block is timed by its local clock signal. As a result, the on-chip clock trees can be simplified, leading to power and area reduction. GALS design also provides a platform for system-level power optimization by the dynamic voltage and frequency scaling (DVFS) [2]. Switching activity in the GALS islands is intrinsically randomized, which contributes to smooth supply currents over time. By introducing particular modulation on local clock signals, the attenuation of switching noise can be further improved [3].

One of the most challenging tasks in GALS design is the implementation of robust asynchronous interface circuits with low overhead. In principle, three asynchronous schemes have been explored to address the synchronization issues for cross-clock domains communication. The most straightforward way is to deploy cascaded flip-flops at the clock boundaries. By this means, extra clock cycles are provided to solve the metastability due to asynchronous data sampling, with the latency penalty. Dual-clock FIFOs are also adopted for transferring data over two clock regions in GALS design. Traditionally, Gray encoding is applied in read/write addressing for FIFO status detection. In [4] Johnson encoding was first exploited to simplify the control logic of dual-clock FIFOs. However, to minimize the throughput drop caused by synchronization latency, the FIFOs have to be sufficiently large. Remarkable overheads in area and power could be introduced as a result. In recent years, an alternative method to GALS design, named pausible clocking scheme, has been developed [5] [6] [7]. Data transfer across clock domains is scheduled by local handshaking signals. The clocks both on transmitter and on receiver can be paused to avoid metastability at data sampling. Also, dynamic frequency scaling on each local clock is facilitated by the pausible clocking scheme. As a latest example, this solution has been employed in the design of a DVFS NoC platform for 4G wireless Telecommunication [8].

This paper reports the implementation and measurements of an OFDM baseband transmitter (BB TX) chip, named *Moonrake*, in a state-of-the-art 40-nm CMOS technology for 60-GHz WLAN applications. To evaluate GALS vs. synchronous technology in a homogeneous experimental setting, two function-identical TX cores - a synchronous baseline design and a pausible clocking based GALS counterpart, were fabricated on the same die in a common package. In Section II, the optimized GALS design methodology based on the pausible clocking is presented. The design details of *Moonrake* TX chip are shown in section III. Measurements and comparison in area and power between the synchronous and GALS designs are reported in section IV. Section V draws a short conclusion of the work.

## II. OPTIMAL GALS DESIGN METHODOLOGY BASED ON PAUSIBLE CLOCKING SCHEME

### *A. System Partitioning Strategy*

GALS partitioning can be done according to the connectivity and dataflow of the functional modules as reported in [6] [7], to lower the number of asynchronous channels and the activity of cross-clock domain communication. This method benefits GALS design by avoiding system throughput drop caused by frequent synchronization. However, it contributes little to global timing optimization as well as chip power/area saving. Saving power consumption in clock networks by optimal GALS partitioning was addressed in [9]. The author suggested dividing the system to *N* equal sized clock islands, and significant power reduction (up to 70%) on the on-chip clock networks was shown. Similar studies on the GALS processors indicate that eliminating the global clock only introduces marginal power reduction at the system level [2]. Up to now, no measured result is reported on the system power saving by GALS design.

978-3-9810801-8-6/DATE12/©2012 EDAA

The GALS system partitioning strategy applied in this work has taken into account all above functional and physical criteria: functional modules with loose control/data dependency are ungrouped into different GALS clock islands, and all GALS blocks are further balanced in terms of area and power dissipation. Our goal is twofold: first to maximize the power and area benefits offered by the GALS design; second to minimize the throughput and latency penalties caused by cross-clock arbitration.

### B. GALS Design Flow

Input and output port controllers are utilized in the pausible clocking scheme to schedule the communication across clock domains. To obtain high throughput, port controllers are often designed as asynchronous finite state machines triggered by the input and feedback output transitions. Academic tools such as *Petrify* can be applied for the behavioral description (STG for example) and synthesis of asynchronous port controllers [10].

However, an open question is how to verify the behavior of asynchronous controllers at the system level in a synchronous environment. Invalid timing sequences of input transitions can lead to false switching on I/O ports, which is known as the most error-prone issue in the pausible clocking scheme. Normally, this is verified by the Verilog netlist simulation after synthesis, causing additional iteration in asynchronous wrapper design if an error occurs. In order to address this challenge, a behavior level simulator for mixed synchronous-asynchronous design, named *AsipIDE*, was developed [11]. Relying on *Asip* packaging format, the VHDL/Verilog design of synchronous modules and the STG design of asynchronous ports can be co-simulated at system behavior level in an early design stage.

The *Petrify* synthesized circuits are speed-independent, and have to be further implemented as soft/hard macros to ensure the robustness to interconnect delay. Therefore, the hierarchical layout is preferred in the pausible clocking based GALS design. In this work, each asynchronous port controller was separately designed at the transistor level and integrated as a hard macro at the system level. Path delay in the port controllers has to be characterized and back-annotated for the timing analysis of interface circuits. Due to the existence of combinational loops, special attention has to be drawn on the timing characterization of asynchronous state machines.

Therefore, aiming at robust and efficient GALS design, a design flow optimized for the pausible clocking scheme, along with the industry-standard CAD tools, is presented in Fig. 1. Above mentioned points advance our approach from the one developed by the ETHZ GALS group [7].

## III. Design of Moonrake Chip

### A. Chip-Level Design Consideration

An industry-relevant design, an OFDM BB TX with up to 1 Gbps datarate for 60 GHz WLAN applications was adopted to evaluate the feasibility of the pausible clocking based GALS technique. The synchronous baseline BB TX and the GALS counterpart were implemented in parallel on the same die, thus allowing an objective comparison in a homogeneous setting: identical in function and in process.

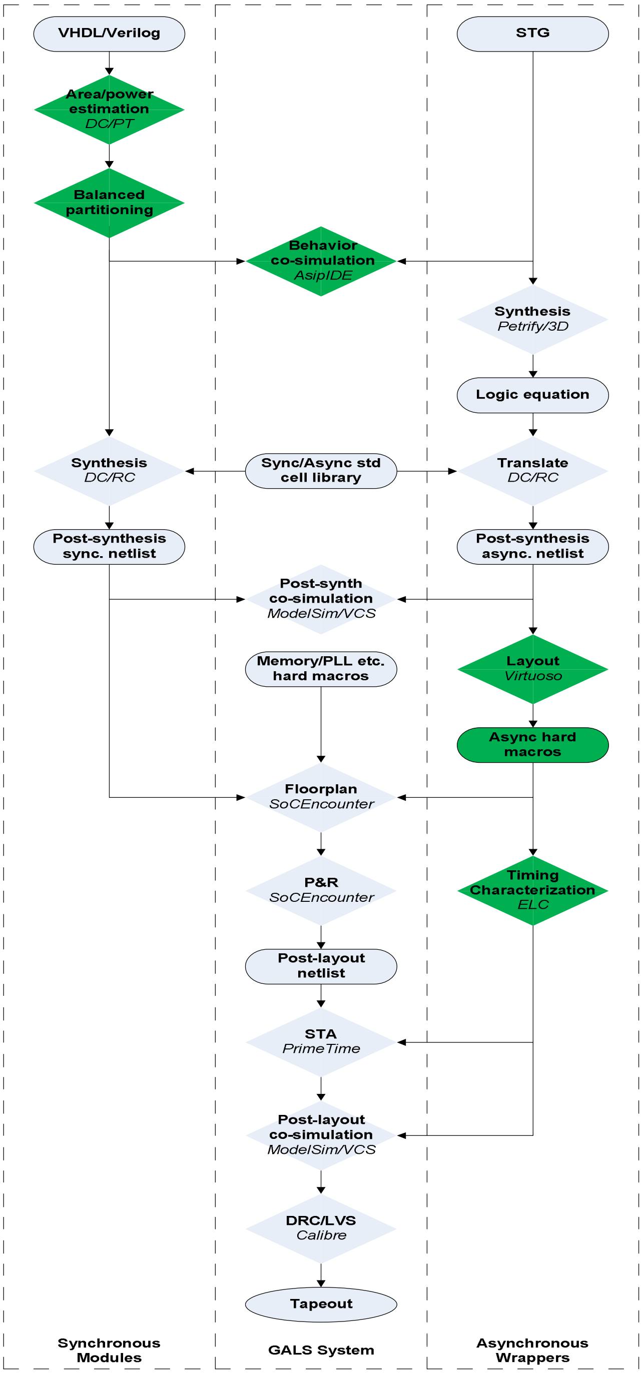

Figure 1. Pausible clocking based GALS design flow

To save the area all the data pads were shared by the two BB TX cores. The built-in self-test (BIST) logic based on the pseudo-random number generator (PRNG) and multiple-input signature register (MISR) was inserted for on-chip functional testing. A JTAG controller was applied for working mode configuration and GALS clock programming. A PLL hardcore was integrated to generate the clock for the synchronous BB TX. To support Giga-bit throughput, the synchronous BB TX was highly paralleled and pipelined in structure. It was fully validated on FPGA, with 7.8 M equivalent gates in complexity.

### *B. GALS OFDM BB TX Design*

The area and power estimation of the BB TX functional modules were performed based on the post-synthesis netlist using typical CAD tools (*DesignCompiler* and *PrimeTime*). According to the criteria proposed in II.B, a partition scheme was applied as presented in Table I. The most computation-intensive functional module, 256-point IFFT processor, was broke down into 2 GALS blocks. Due to the considerable area and power expenses, the 6 interleavers were grouped into 3 clock islands. To limit the number of data links, all the tightly coupled frond-end control and pre-processing modules were integrated into a single GALS clock region. As a result, the BB TX design was partitioned to 6 balanced GALS blocks, each at most 15% larger in area and 11% higher in power than average. A total of 16 point-to-point channels were used. Fig. 2 shows the architecture of the GALS BB TX.

### *C. Implementation of Moonrake Chip*

A hierarchical layout was performed. All asynchronous I/O port controllers were designed at the transistor level and then integrated as hard macros on the chip. Because the bundled-data protocol was utilized in the asynchronous data links, the primary consideration at top-level layout was to constrain the wire delays on handshake signals in the GALS BB TX. All the port macros were intentionally placed as close as possible to the corresponding GALS cores. Explicit timing constraints were further specified on the max/min interconnects delays on the asynchronous datalinks.

The chip is 9 mm$^2$ in size using the state-of-the-art 40-nm CMOS process, with 218 memory cells and 219 signal/power pads on the chip. Most of the area was actually occupied by the on-chip memory cells. A LBGA-345 package was adopted for the *Moonrake* chip.

TABLE I. POWER/AREA ESTIMATION AND SYSTEM PARTITIONING OF GALS OFDM BB TX DESIGN

| | **GALS Block 1** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Input FIFO | Input controller | Symbol mapping | Universal scrambler | Middle controller | FEC encoder [12:1] | Pilot inserter | Mapping [4:1] | ***Total*** |
| Power | 0.1% | 0.1% | 0.5% | 0.0% | 7.0% | 0.09% | 3.1% | 0.08% | ***10.97%*** |
| Area | 1.0% | 0.1% | 1.0% | 0.0% | 12.0% | 0.06% | 5.0% | 0.14% | ***19.3%*** |
| | | | | | | | | | |
| | **GALS Block 2** | | | **GALS Block 3** | | | **GALS Block 4** | | |
| | Interleaver 1 | Interleaver 2 | ***Total*** | Interleaver3 | Interleaver 4 | ***Total*** | Interleaver 5 | Interleaver 6 | ***Total*** |
| Power | 8.7% | 8.7% | ***17.4%*** | 8.7% | 8.7% | ***17.4%*** | 8.7% | 8.7% | ***17.4%*** |
| Area | 8.9% | 8.9% | ***17.8%*** | 8.9% | 8.9% | ***17.8%*** | 8.9% | 8.9% | ***17.8%*** |
| | | | | | | | | | |
| | **GALS Block 5** | | | | | **GALS Block 6** | | | **Post-synth BB TX** |
| | IFFT_64P 1 | IFFT_64P 2 | IFFT_64P 3 | IFFT_64P 4 | ***Total*** | IFFT_4P | Out Stage | ***Total*** | |
| Power | 4.9% | 4.3% | 4.3% | 4.3% | ***17.8%*** | 11.3% | 7.2% | ***18.5%*** | **230mW** |
| Area | 2.7% | 2.4% | 2.4% | 2.4% | ***9.9%*** | 10.3% | 6.7% | ***17%*** | **2.2mm$^2$** |

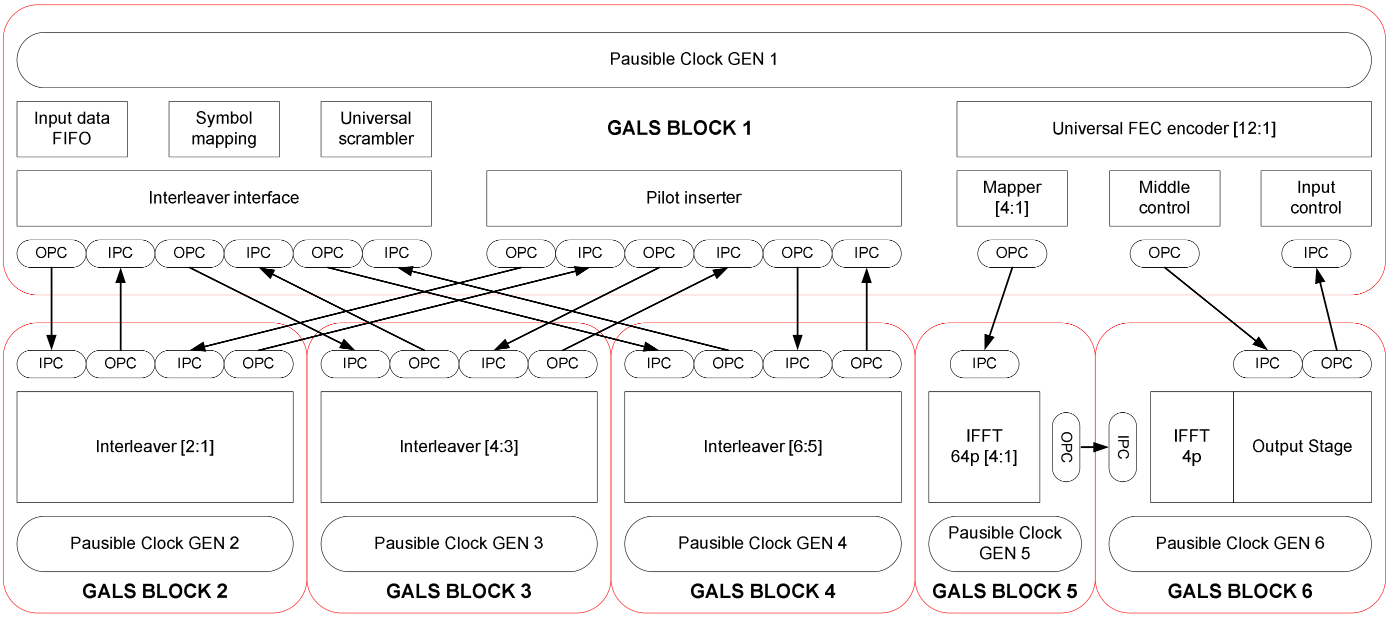

Figure 2. Architecture of pausible clocking based GALS OFDM BB TX design

## IV. Experimental Results of Moonrake Chip

### A. Overheads of the GALS Infrastructure

Firstly, the cost for introducing GALS system design by the pausible clocking scheme was addressed. The post-layout area and power of the GALS infrastructure, including the local clock generators, asynchronous I/O port controllers and input data registers, are shown in Table II. In general, pausible clocking specific circuits accounted for 2.7% in power and 1.6% in area of the entire GALS BB TX design. The majority of the GALS overhead came from the local clock generators. The I/O port controllers were negligible both in power and in area. In particular, around 740 extra *DFFs* were inserted as input data register on the GALS channels [12], which correspond to buffer each datalink only with a 1.5-word FIFO in average.

TABLE II. Power/area occupancy of GALS interface

| | *Local Clock Gen* | | *I/O Port Cntr* | | *Input Data Reg* | |
|---|---|---|---|---|---|---|
| Power (*mW*) | 4.05 | 1.77% | 0.19 | 0.08% | 1.80 | 0.79% |
| Area ($\mu m^2$) | 30K | 1.25% | 640 | 0.03% | 5.7K | 0.26% |

### B. GALS Optimization of High Fanout Nets

Table III shows the key features of SYNC/GALS clock trees after layout. It is observed that the complexity of clock trees was reduced in GALS BB TX, with an almost 3X drop of the number of clock buffer levels and a 6% saving in the total number of buffer cells. As a consequence, above 20% power reduction was achieved in the GALS clock networks. Also, GALS design resulted in a 30% drop in the number of buffers inserted for the reset network distribution.

TABLE III. Features of clock and reset buffer trees

| | *Clock Trees* | | | | *Reset Trees* | |
|---|---|---|---|---|---|---|
| | *Num.* | *Level* | *Buffer* | *Power* | *Num.* | *Buffer* |
| SYNC TX | 1 | 27 | 1645 | 42*mW* | 1 | 582 |
| GALS TX | 6 | ≤ 10 | 1549 | 32*mW* | 6 | 405 |

### C. Area and Power Comparisons

To compare the complexity between the synchronous and GALS BB TXs, the cell area was reported as shown in Table IV. The GALS design contributed to a slight overhead (0.57%) after synthesis over the synchronous baseline (w/o PLL). However, during the layout we observed a 1.26% increase of the area of synchronous BB TX and a 1.16% shrink of the GALS part. It led to a 0.67% area reduction in the GALS design after layout. Power dissipation was also measured as shown in Table V. The GALS design prevailed over the synchronous baseline (w/o PLL) by a 5.8% reduction in power when running at about 160 MHz. It was mainly caused by the simplification of on-chip clock networks.

A more objective comparison, or taking into account also the PLL overheads ($0.1mm^2$, < 1.5mW at 160MHz), revealed that the GALS BB TX out performed the synchronous one with 5% of area reduction and 6% of power saving.

TABLE IV. Cell Area Comparison

| | Post-synthesis netlist | Post-layout netlist | ***Difference*** |
|---|---|---|---|
| SYNC TX w/o PLL | 2206895$\mu m^2$ | 2234712$\mu m^2$ | ***+27817$\mu m^2$*** |
| GALS TX | 2225823$\mu m^2$ | 2220080$\mu m^2$ | ***-25743$\mu m^2$*** |
| ***Difference*** | ***+18928$\mu m^2$*** | ***-14632$\mu m^2$*** | |

TABLE V. Power Comparison at 160 MHz

| | Post-layout simulations | Chip measurement | ***Difference*** |
|---|---|---|---|
| SYNC TX w/o PLL | 234*mW* | 252*mW* | ***+18mW*** |
| GALS TX | 225*mW* | 237*mW* | ***+12mW*** |
| ***Difference*** | ***-9mW*** | ***-15mW*** | |

## V. Conclusion

For the first time, experimental results are reported on the system-level area and power benefits offered by GALS design. The cell area comparison clearly shows the GALS contribution for timing convergence in layout. The effort for timing closure in the synchronous TX was much more challenging due to the chip-wide cell location and global synchronization. In contrast, balanced partitioning engaged in the GALS design result in a set of compact locally-timed blocks, which can be optimized more aggressively and efficiently. The localized clock/reset signals also simplified the synthesis of on-chip high fanout nets, leading to lower power/area overhead. Attributed to the simple interface of the pausible clocking scheme, the marginal overhead due to the GALS infrastructure was compensated at the system level.

## Acknowledgement

This work has been supported by the European Project GALAXY under grant reference number FP7-ICT-214364 (www.galaxy-project.org).

## References

[1] D. M. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D thesis, Standford Univ., 1984.

[2] A. Iyer, and D. Marculescu, "Power and performance evaluation of globally asynchronous locally synchronous processors," in *Proc. 29th IEEE Symp. Computer Architecture (ISCA)*, 2002

[3] X. Fan, M. Krstić, C. Wolf, and E. Grass, "A GALS FFT processor with clock modulation for low-EMI applications," in *Proc. 21st IEEE Intl. Conf. Application-specific Systems, Architectures and Processors (ASAP)*, 2010.

[4] Y. Thonnart, E. Beigne, and P. Vivet, "Design and implementation of a GALS adapter for ANoC based architectures," in *Proc. 15th IEEE Intl. Symp. Asynchronous Circuits and Systems (ASYNC)*, 2009

[5] K. Y. Yun, and R. P. Donohue, "Pausible clocking: a first step toward heterogeneous systems," in *Proc. 15th IEEE Intl. Conf. Computer Design (ICCD)*, 1996.

[6] J. Muttersbach, T. Villiger, and W. Fichtner, "Practical design of globally-asynchronous locally-synchronous systems," In *Proc. 6th IEEE Intl. Symp. Advanced Research in Asynchronous Circuits and Systems (ASYNC),* 2000.

[7] F. K. Gurkaynak, S. Oetiker, T. Villiger, N. Felber, H. Kaeslin, and W. Fichtner, "On the GALS design methodology of ETH Zurich," in *Proc. 1st Intl. Workshop Formal Methods for Globally Asynchronous Locally Synchronous Architecture (FMGALS)*, 2003.

[8] E. Beigne, et al., "An asynchronous power aware and adaptive NoC based circuit," in *IEEE JSSC*, 2009.

[9] A. Hemani, et al, "Lowering power consumption in clock by using globally asynchronous locally synchronous design style," in *Proc.36th Design Automation Conference (DAC)*, 1999.

[10] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno and A. Yakovlev, "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers," in *IEICE Trans. Inf. and Syst.*, March, 1997.

[11] L. Janin, and D. Edwards, "AsipIDE: a graphical framework to design and debug GALS systems through simulation and prototyping," available at http://www.minatec.org/nocs2010.

[12] X. Fan, M. Krstić, and E. Grass, "Analysis and optimization of pausible clocking based GALS design," in *Proc. 27th IEEE Intl. Conf. Computer Design (ICCD)*, 2009.