A Novel Interconnection Approach for Globally Asynchronous Locally Synchronous Circuits

Wolfgang Forster
Vienna University of Technology,
Distributed Systems Group E184-1
w.forster@infosys.tuwien.ac.at

Eric Armengaud
Vienna University of Technology,
Embedded Computer Systems Group E182-2
armengaud@ecs.tuwien.ac.at

Abstract

This paper introduces a new methodology to solve the interfacing problem in Globally Asynchronous Locally Synchronous *(GALS) design approaches. We present a generic high-speed and delay-insensitive connector based on asynchronous* four state logic *(FSL). The advantages of this approach are as follows: First, it provides flexibility in the time domain since the data transfer is based on local handshakes and does not depend anymore on a global clock signal. Consequently, it removes timing constraints and even enables local optimizations. Second, this approach only requires a small number of interfacing signals, thus reducing the routing resources needed between data source and sink. Furthermore, the proposed architecture does not require customized delay lines and thus suits well for both ASIC and FPGA platforms. A modified FlexRay bus analyzer tool has been used to illustrate the advantages of our approach: High-speed and delay-insensitive data communication between different clock domains.*

Context

❑ **Integration trend in digital design technologies**
* Large chips with several IPs
* Different (proprietary) interfaces
* Different clock domains

❑ **Globally Asynchronous Locally Synchronous (GALS) approach**
* Several synchronous islands use asynchronous connectors for communication
* Two-stage synchronizer or *Clock Stretching* techniques for synchronization

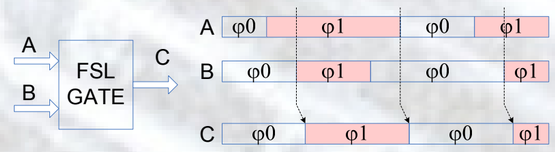
❑**Limitations**
* Low data rate in case of two-stage synchronizer
* Complex gates for clock stretching techniques are not applicable for FPGA based platforms

Async. logic

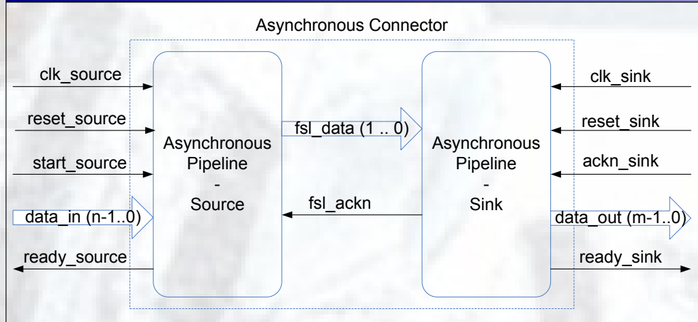
❑**Four State Logic (FSL)**
* Dual rail encoding: two signals required to represent a logical ‚0' or ‚1'
* Logical state (‚0' or ‚1') has two representations: one in phase φ_0 and one in phase φ_1
* Alternating phases (φ_0 and φ_1) signal the availability of new data
* Exactly one transition required from one phase to the other (independent from data information)

FSL logical state	φ0	φ1
LOW ('0')	(0,0)	(0,1)
HIGH ('1')	(1,1)	(1,0)

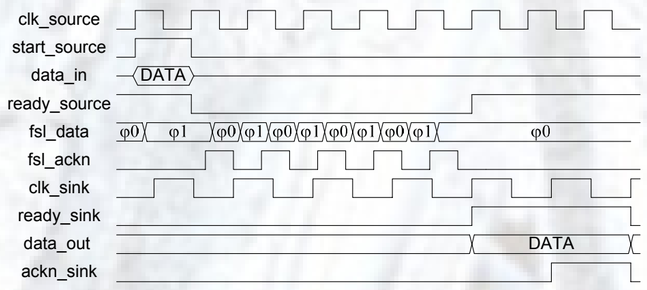
❑**Advantages**
* No global clock required anymore (local handshake used instead)
* Enables local synthesis optimizations



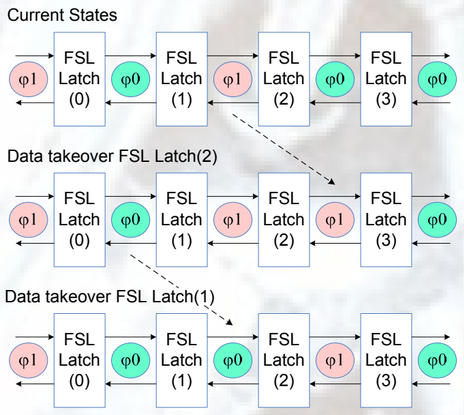
Connector



❑ **Simple interface**
* Simple interface: Handshake mechanism (start - acknowledge - ready) synchronized to each local clock domain
* No metastability problems
* Delay-insensitive interconnection
* Minimizes the number of interconnect signals (reduced routing resources)

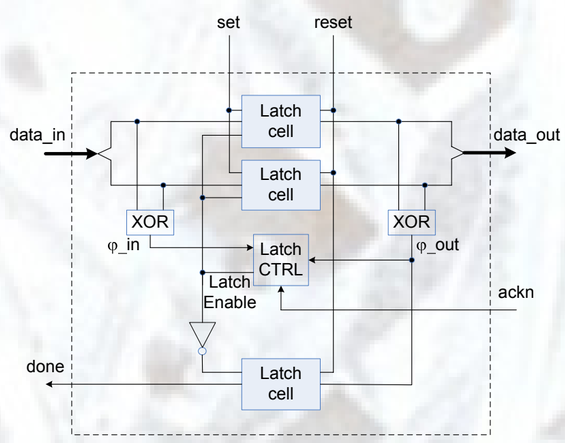


❑ **Asynchronous serial data transmission**
* Data vector serialized into a FSL register
* Single bits sent asynchronously
* Received data stored into a vector during reception



❑ **Elastic pipeline**
* Bits are transmitted as soon as possible
* Local handshakes regulate the pipeline sequences
* Independent from global clock signal

❑ **Advantages**
* Enables local synthesis optimizations (short paths are faster than clock period)
* Enables inter-chip communication (long paths are supported, too)
* Eliminates clock domain crossing problems



❑ **FSL latch**
* Consists of three latch cells to store the current state until the next is available
* Updates its state as soon as the next FSL latch is ready (*ackn*) and the previous FSL latch provides new data
* Signals operation completed to the previous FSL latch (*done*)

Evaluation

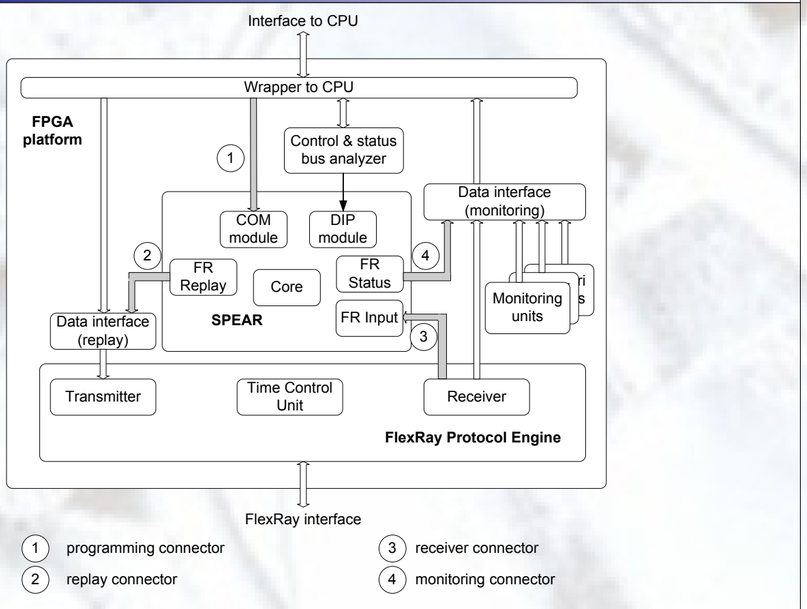
Connector	Asynchronous Serial				Synchronous Serial				Synchronous Parallel			
Data Width	LC	Duration	Signals	TR	LC	Duration	Signals	TR	LC	Duration	Signals	TR
[bit]	[]	[ns]	[]	[MBit/s]	[]	[ns]	[]	[MBit/s]	[]	[ns]	[]	[MBit/s]
8	131	177	3	**15.07**	49	1200	3	**2.22**	27	230	10	**3.48**
16	261	285	3	**18.71**	87	2300	3	**2.32**	43	230	18	**3.86**
32	519	505	3	**21.12**	162	4480	3	**2.38**	75	230	34	**4.09**
64	1036	880	3	**24.24**	312	9050	3	**2.36**	139	230	64	**4.35**

❑**Synchronous serial connector**
* $T=N.[2.T_{Csource} + 2.T_{Csink} + 2.T_{ic} + 2.T_{preset} + 2.T_{readout}]$
* Low interconnect resources while low transmission performances

❑**Synchronous parallel connector (dual-clock FIFO)**
* $T=2._{TCsource} + 2.T_{Csink} + 2.T_{ic} + T_{preset} + T_{readout}$
* High transmission performances while high interconnect resources

❑**Asynchronous serial connector**
* $T=N.[T_{cell} + T_{mic}] + T_{preset} + T_{readout}$
* Low interconnect resources **and** high transmission performances
* Does not depend on global clock signal!

Application



❑**Integration of the SPEAR µC into our bus analyzer tool**
* Increases computation resources
* Platform independent
* Real-time guaranties
* Flexible, extendable

❑ **Different interface implementations**
* Programming connector: dual clock FIFO
* Replay connector: async. connector
* Receiver connector: async. connector
* Monitoring connector: async. connector

http://www.infosys.tuwien.ac.at/compass
http://ti.tuwien.ac.at/armengaud/extract

Vienna University of Technology
Distributed Systems Group
Information Systems Institute
A-1040 Vienna, Argentinierstrasse 8
http://www.infosys.tuwien.ac.at/

This work has been partially funded by the FIT-IT [embedded systems initiative of the Austrian Federal Ministry of Transport, Innovation, and Technology] and managed by Eutema and the Austrian Research Agency FFG within the projects COMPASS (grant 809444) and ExTraCT (grant 810834)