OCP-IP Building Blocks for Space SoC Final Presentation

Aurélien LEFEVRE
Astrium Satellites Elancourt

Microelectronics Presentation Days
ESA/ESTEC, April 1st 2010

All the space you need

“OCP-IP Building Blocks for Space-SoC”

- ESA contract #21393/08/NL/JK
- ESA Technical Officer: Kostas Marinis

- Contractor: EADS Astrium Satellites (Elancourt, France)
  - Aurélien Lefèvre, Marc Souyri
- Magillem Design Services (Paris, France)
  - Cyril Spasevski, Serge Amougou
- Aeroflex Gaisler (Gothenburg, Sweden)
  - Sandi Habinc, Jan Andersson

- July 2008 – Feb 2010

Background

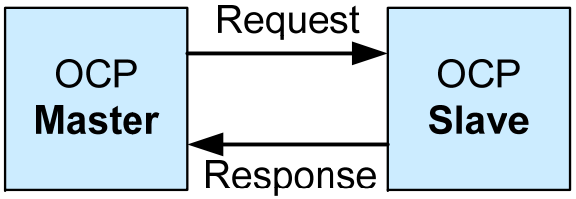
- Systems-on-Chip provide several benefits
  (Mass, Volume & Power ↘ => Eqpt Cost ↘; Processing speed ↗)
- But they require **re-using** blocks previously developed and already validated
- And IP re-use brings an issue of **heterogeneity**
  - IP Cores developed in different contexts by different teams
  - The interfaces can differ (physical, protocol)
  - Integration is not straightforward
    - requires a know-how transfer
    - merge of synthesis scripts, ...

=> The aim of this contract is to investigate the use of OCP and SPIRIT IP-XACT standards

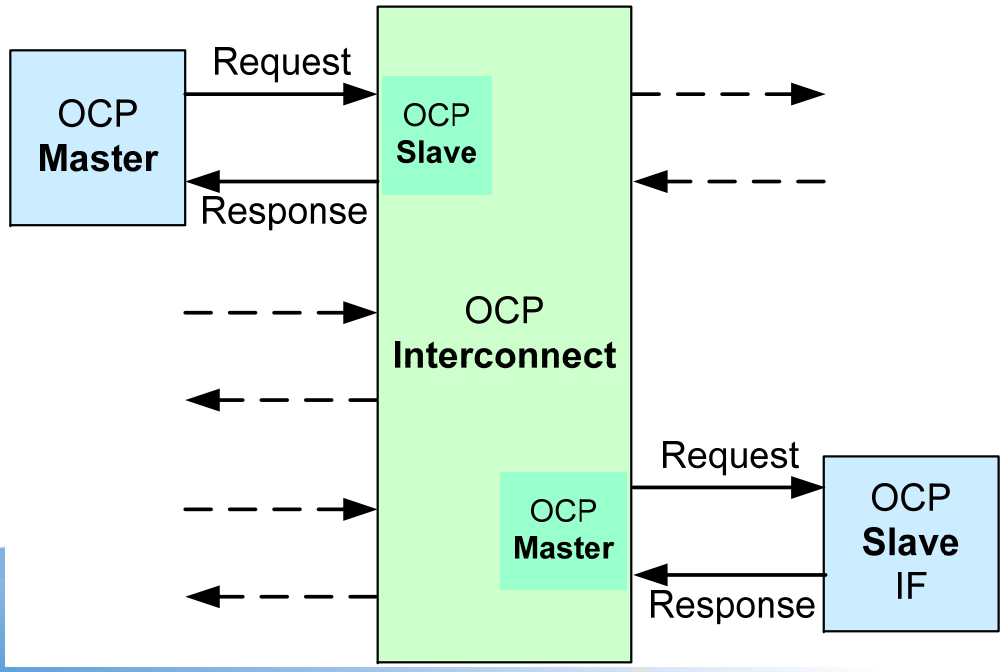
Open Core Protocol (OCP)

- OCP is a protocol for on-chip communication
- It defines core-centric interfaces: “Sockets”
  - Point-to-point connection
  - Master/Slave interfaces are symmetric, Request/Response signals

Direct connection



Connection through OCP interconnect



Open Core Protocol (OCP)

- OCP defines core-centric interfaces: “Sockets”
  - The Sockets are very configurable
    - Basic OCP is very simple / low complexity
    - Basic extensions = bursts, sub-word accesses, ...
    - Complex extensions = security levels, 2D burst types, out-of-order operation, threads for concurrency...
  - **The configuration shall be tailored to match the particular IP Core needs** (this is the “core-centric” concept)
    - Incompatible sockets can be connected with an OCP interconnect
    - In practice, restriction to “OCP Profiles” (widespread configurations) and to configurations supported by the OCP interconnects
  - OCP is non-proprietary and openly licensed

SPIRIT IP-XACT

- IP-XACT is an XML schema to describe IP Cores
  - List of Ports, Registers, Parameters
  - Source files, scripts, etc.
- Standardized mechanism to express and exchange information about an IP Core
  - from the IP designer to the SoC integration team, through tools or company borders
- Also defines a standard API to access the descriptions (the “TGI” API). It is possible to write tool-independent configurators, generators of top-levels / of register map headers / of documentation, ...
- Industry and now IEEE standard (IEEE 1685-2009)

Activity Overview

- Evaluation of Open Core Protocol & SPIRIT IP-XACT

- Spacewire-OCP Design, Verification & Synthesis (Astrium)
- Spacewire IP-XACT packaging (Astrium)
- LEON2FT-OCP Design, Verification & Synthesis (Magillem)
- AMBA-OCP Bridges Design, Verification & Synthesis (Magillem)
- AMBA-OCP Bridges IP-XACT packaging (Magillem)
- LEON2FT IP-XACT packaging (AE/Gaisler)

Selection of OCP Sockets

- Review of OCP specification wrt communication needs
- Two OCP sockets defined, close to AHB / APB ("Tailor the sockets to match the particular IP Core needs, without overdesign")
- Main differences:
  - H-Socket can send multiple requests before receiving the first response
  - P-Socket can send error response
  - No notion of bus arbitration in OCP
  - No SPLIT/RETRY in H-Socket
  - The OCP sockets chosen are more simple than AMBA. The arbitration logic **centralized** in the interconnect instead of being **distributed** over all the IP Cores, what reduces bug risk and design cost

Spacewire-OCP

- Based on Spacewire-AMBA IP Core: AMBA interfaces have been replaced by “native” OCP Sockets
- Verification: re-used SpW-AMBA testbench, with AMBA-OCP bridges
  (test sequences extended; coverage focused on OCP parts)
- Same functionality / register map
- AMBA interface FSM simplified for OCP (no arbitration, no SPLIT&RETRY)
  - => Functional performance: SpW-OCP better for TX master interface
  - => Synthesis: SpW-OCP is a little faster and smaller

AMBA-OCP Bridges

- Bridges are protocol converters between AMBA and OCP
  - Enable integration of OCP cores in AMBA AHB/APB systems, and vice-versa
  - Four bridges developed: AHB2OCP, OCP2AHB, APB2OCP and OCP2APB
- Bridges are based on existing cores already available and verified at Magillem
- Full AMBA support (SPLIT, RETRY, burst types)
- Verification using random traffic generation (full functional and statement code coverage) and Verification Components based on the *e* language
- Synthesized in ACTEL RTAX2000 (FPGA) and ATMEL MH1RT (ASIC)

LEON2FT-OCP

- Based on LEON2FT ESA processor (SPARC V8 Integer Unit with caches and AMBA interfaces) : AMBA interfaces have been replaced by “native” OCP sockets
- **Two** master OCP sockets (one per cache) instead of **one** master AHB for both caches
  - To investigate **multi-layered** interconnects
- Verification: re-used LEON2FT-AMBA testbench
  (test sequences extended; coverage focused on OCP parts)
- A fair performance comparison requires a commercial OCP interconnect
- A platform with an interconnect from SONICS has been set up
- Synthesis: LEON2FT-OCP is a little faster and bigger
  (because of the 2 master sockets)

SPIRIT IP-XACT Packaging

- IP-XACT packaging of SpW (AMBA & OCP), LEON2FT distribution (incl. LEON2FT-OCP) and AMBA-OCP bridges
  - IP-XACT v1.4 – Use of Magillem Environment
- The IP-XACT descriptions contain:
  - the ports & bus interfaces
    - bus interfaces to group ports together (ex: AHB, APB, OCP, ...)
  - the parameters (generics, VHDL constants)
  - the registers
  - views and file sets
- The IP-XACT packages also contain “TGI Generators”

Views and File sets (part of IP-XACT description)

- One ***view*** per use (Documentation, Simulation, ASIC_synthesis, FPGA_synthesis)
- A ***file set*** is a consitent set of files (VHDL_source, Simulation_scripts, Documentation, ASIC_synthesis_reports...)
- A view contains one or multiple file sets
- A file set can be included in multiple views (VHDL_Source is used in Simulation and Synthesis)

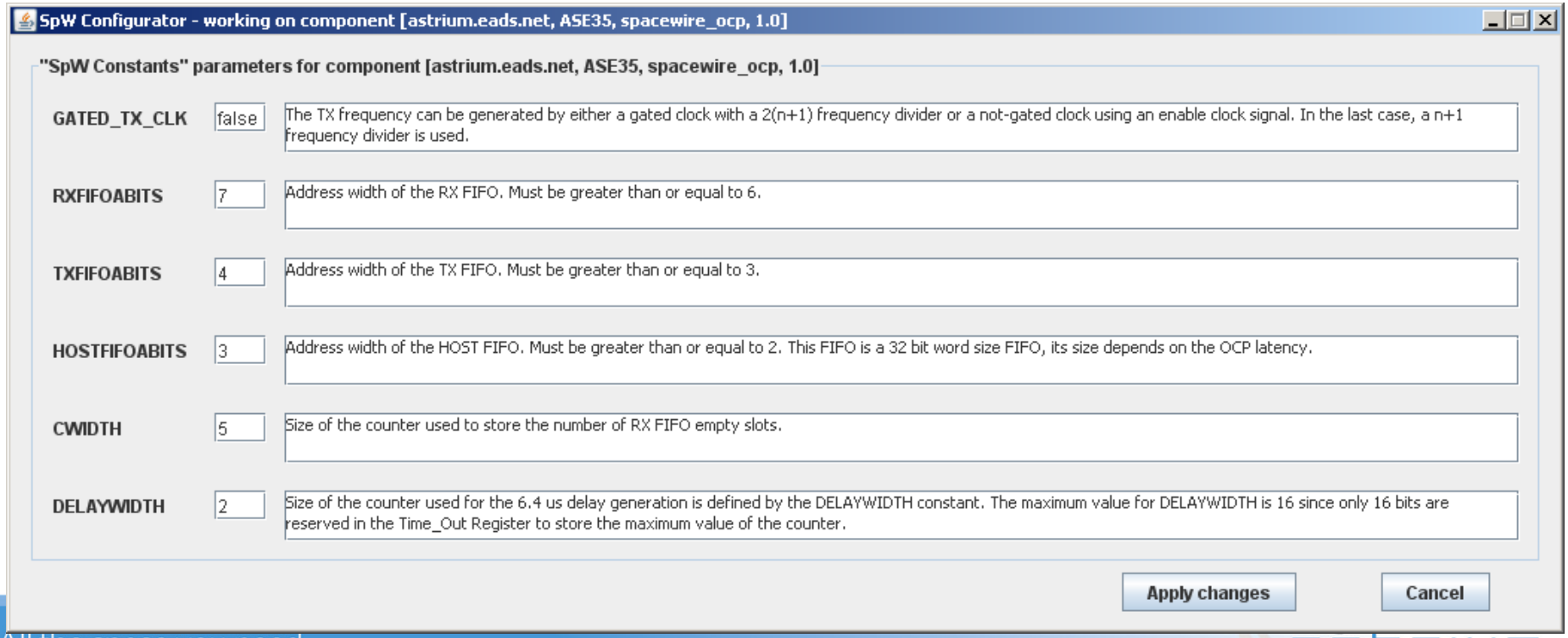
TGI Generators

- Programs that use the standard “TGI” API (access functions to the IP-XACT descriptions)
- The TGI API is implemented by the SPIRIT environment
- Can be written in various languages – JAVA in our case

- LEON2FT TGI generator
  - can replace the existing TCL/TK configuration scripts
  - it reads parameters from the IP-XACT description of the design (through the TGI API) and creates the VHDL LEON2FT configuration file

TGI Generators

- Spacewire TGI generator: configuration of parameters which are VHDL constants (keeps VHDL and IP-XACT consistent, checks validity)



SPIRIT IP-XACT Packaging Experience

- The different views of a component must be pin-consistent in IP-XACT
  - not possible to have a single IP-XACT for SpW with a parameter to select AMBA or OCP => a TGI Generator has been written to swap between the two IP-XACTs
- VHDL records are not supported in IP-XACT 1.4.
  Two workarounds:
  - use of “vendor extensions”
  - writing wrappers to expand the records
- Verification
  - XML well-formedness and validity with XML tools
  - Semantic checks with the SPIRIT Environments
  - Functional tests for the TGI generators

Conclusions (1)

- Outcomes:
  - OCP Sockets for Spacewire and LEON2FT
    - In particular, original implementation of LEON2FT with 2 master sockets
  - AMBA-OCP Protocol converters
  - IP-XACT packages for SpW (AMBA and OCP), LEON2FT (AMBA and OCP) and AMBA-OCP Bridges

Conclusions (2)

- Open Core Protocol
  - Less complex than AMBA for equivalent functionalities
  - But many different OCP configurations possible, and AMBA is more widespread
- SPIRIT IP-XACT
  - Seems to be a promising technology for packaging, exchanging and integrating IP Cores (now IEEE standard)
  - The activity focused on the packaging stage. The utility of IP-XACT can be demonstrated later by assembling, configuring, generating the top-level / documentation / SW register description, ...