OSSS: An Approach for Modeling, Seamless Refinement, and Synthesis of HW/SW SoC

Frank Oppenheimer
Kim Gruettner

OFFIS - Institute for Information Technology
What can we do about it?

- Components
- IP components (Re-use)
- High-level Modeling
- Communication
- TLM-like Communication (Abstraction)
- Architectures
- Standard Platforms (Re-use)
- Performance Analysis (Simulation)
- Systems
- Hardware/Software Co-design (Simulation)
- Synthesis (Automation)
OVERVIEW

- Introduction and Motivation
- "Moore remains (at least for now) valid"
- OSSS Methodology
- "From Application via Virtual Platform to Implementation"
- Avoiding conflicts
- Occam's Razor for Software modeling
- "From Methods to Signals"
- OSSS Synthesis Tool chain
- Conclusion
OVERALL GOAL

APPLICATION LAYER

IMPLEMENTATION

BUS $\mu$P / $\mu$C

RAM

BUS IF

BUS IF

ARBITER

MEMORY CONTROLLER

BUS IF

EFICIENT DESIGN FLOW FOR EMBEDDED HW/SW SYSTEMS
OSSS 2.0 - Oldenburg System Synthesis Subset

C++ System C Software Hardware Virtual Architecture

OSSS 2.0 OSSS Software Tasks
new / delete
Pointers
C++ Polymorphism
Classes & Structures
Templates
Inheritance
bool
Synth. subset of System C data types
OSSS data types (oss::enum, oss::array, oss::serializable_object)
Processes (sc::method, sc::cthread)
Shared Objects
Processors
OSSS Channels
Object Sockets
Memories
Modules
Signal Ports (sc::in, sc::out, sc::inout)
Polymorphic Objects cross compiler & target dependent
Integer types
References
Enumerations
Arrays
Mutex
Semaphore
Events
Report Handling
Tracing
Signals
Shared Objects

- Method-based Interface for Objects and Modules
- Mutual Exclusive Access (preserves data consistency)
- Guards for Access Control
- Similar to Hier. Channel but precise Synthesis Semantics

```
MyBufferClass

get(): ITYPE
put(ITYPE)
is_empty(): bool
is_full(): bool
...
BufferP->put(...);
...
BufferP->get();
...
```

MySharedBuffer

MyBuffer, Scheduler

ESCU 16

Frank Oppenheimer
Kim Grütter

Shared Objects in System C/OS-XX

**osss_port**

**< oss_s_shared < FIFO < Packet , 10 > , oss_s_round_robin >**

**template < class ItemType >**

```cpp
class FIFO_put_if : public sc_interface {
public:
    virtual void put(ItemType item) = 0;
    virtual bool is_empty() = 0;
    virtual bool is_full() = 0;
    ...
};
```

```cpp
< < implementations > >
```

```cpp
template < class ItemType >
class FIFO_get_if : public sc_interface {
public:
    virtual ItemType get() = 0;
    virtual bool is_empty() = 0;
    virtual bool is_full() = 0;
    ...
};
```

```cpp
< < implementations > >
```
So far we have seen…

- OSSS (Oldenburg System Synthesis Subset) defines a rich subset of System C/C++ augmented with some extra concepts

- Refinement Layers
  - Application Layer
  - Virtual Target Architecture Layer

Now:
- Communication refinement with OSSS-Channels
- Channel-Synthesis
OSS Refinement of Communication Links

Application: User-defined Transactions

Virtual Target Architecture

Bus Software Processor

Object Socket

Hardware Block

Shared Bus Topology

P2P Topology

Port Interface Module with Process(es)

Shared Object Software Task

Producer My Shared Buffer

Consumer
OSCS Communication Alternatives (1)

Virtual Target Architecture

Software Processor

Object Socket

Hardware Block

Hardware Block

Shared Bus Topology

Point-to-Point Topology

OPB Application Port Interface
OSSS - Channel Synthesis Layer

Software Processor

Hardware Block

Hardware Block

Hardware Block

Hardware Block

Shared Bus

Point-to-Point Topology

Master trans.
Slave trans.
Slave
Arbitrator
Master trans.

Interconnect signals

e.g.: OPB, AMBA, Wishbone RTL Models

Custom-built Point-to-Point Channel
Conclusions: OSSS Methodology

- OSSS Modeling and Refinement
- Application Model
- Refinement to Virtual Target Architecture Model
- OSSS Channels
- Modeling and encapsulation of arbitrary protocols and buses
- Automated generation of internal channel structure
- Separation of communication and behavior enables evaluation of design alternatives
- Automated synthesis completes seamless design flow for embedded HW/SW systems
Thank you for your attention!

Further Information http://icode офис.de
http://andre офис.de
http://www.офис.de/his/

Funded by the EC
OSSS Remote Method Invocation
template<class ItemType, unsigned int Size>
class FIFO {
public:
  OSSS_GUARDED_METHOD_VOID(put, OSSS_PARAMS(1, ItemType, item));
  OSSS_IMPORTED_METHOD(issfull());
  OSSS_GUARDED_METHOD(ItemType, get, OSSS_PARAMS(0));
  OSSS_IMPORTED_METHOD(issempty());
  OSSS_GUARDED_METHOD(bool, issempty, OSSS_PARAMS(0), true);
  OSSS_GUARDED_METHOD(bool, issfull, OSSS_PARAMS(0), true);
};