Re-configurable SoC Design Platform

Chih-Wei Liu

VLSI Signal Processing Group
Department of Electronics Engineering
National Chiao Tung University, Taiwan, R.O.C.
E-mail: cwliu@twins.ee.nctu.edu.tw
Outline

• Introduction
• Configurable Platform
• Example
• Conclusion
Silicon Pendulum

• **standardization**
  - flexibility
  - time to market
  - cost effectiveness

• **customization**
  - performance
  - differentiation
  - value addition
**SoC: System on Chip**

- **System**
  - A collection of all kinds of components and/or subsystems that are appropriately interconnected to perform the specified functions for end users.

- **SoC**
  - **Hardware:**
    - Analog: ADC/DAC, RF, …
    - Digital: Processor, Interface, Accelerator, …
    - Storage: SRAM, DRAM, Flash, ROM, …
  - **Software:**
    - RTOS, Driver, Application, …
**Example: Mobile Phone**

**Yesterday**
- Voice only; 2 processors
- 4 year product life cycle
- Short talk time

**Today**
- Voice, data, video, SMS
- <12 month product life cycle
- Lower power; longer talk time

*Source: EI-SONICS*
SOC Complexity / Abstraction

Yesterday

- Processor-centric (1 or 2)
- Simple I/O
- Manageable Complexity

Today

- Many processing units
- Large amount of I/O
- Overwhelming Complexity!
Conquer the SoC Complexity

• Use a known real entity
  – A pre-designed component (IP, VC reuse)
  – A platform (architecture reuse)

• Partition
  – Based on functionality
  – Hardware and software

• Modeling
  – At different level
  – Consistent and accurate
**IP, VC, PE, ...,**

- Memory controller
- Interrupt controller
- Power management controller
- Internal memories
- Bridges
- Caches
- Other functions
Why Platform-based Design?

• Market needs
  – Diverse products in various application domains
  – Short development cycle
  – Producing a feasible design
  – Cost down
  – Reusing architecture and components of hardware and software designs
A platform is a suite of reusable parts (IP) of many system designs in a limited spectrum of applications.
Platform-based Methodology

- Significant element of Plug-and-Play
- VCs with Standard interface
- Well-defined backbone, interconnection network
- Configurable VCs, or IPs
- High-level system verification
- Hardware/software codesign
- System-level Virtual prototyping for performance evaluation
Outline

• Introduction
• Configurable Platform
• Example
• Conclusion
(Re-)Configurable Hardware

• Configurable hardware

  some functions can adapt to precisely fit a problem, either
  
  – statically (off-line or just at the design time with the support of a compiler a/o a hardware generator)
  
  – dynamically (at run-time involving a local controller or a host processor with a memory subsystem, even with the OS support)

• Reconfigurable hardware

  the functions can be changed after fabrication or shipment (i.e. field configurable)
Configurability Metrics

• Structural
  – PE only
  – interconnection only
  – both

• Temporal
  – once (design time)
  – statically scheduled (run time)
  – dynamically scheduled (run time)
  – evolutionary (run time)

• Spatial (minimum addressable configuration point)
  – full
  – partial
Why Configurable?

• All IPs are typically customized to meet specific SoC specification
• Software upgradability
• Short product cycles
• E.g. External memory controller supports
  – memory types (sync. Or async.)
  – Sizes,
  – Widths,
  – Banks,
  – Etc.
Benefit of configurable platform

General-Purpose Configurable Platform

- Design-time configurable
  - Xtensa, Tensilica
  - PICO, HP LAB
  - Excalibur, Altera

- Run-time Configurable
  - SA-1110, Intel
  - M-Core
Domain-Specific Platform

- Radio Processor, outer receiver for radio access
- Channel Processor, for radio channel-related signal processing
- Intel IXA, network Processor
- Trimedia for multimedia processor
- …
- Cost to develop new platform from one domain to another is substantial.
Dynamic Platform Management (DPM)

• Components:
  – Configurable processors,
  – Parameterized caches,
  – Configurable memory hierarchies,
  – Configurable bus architecture,
  – Programmable logic,
  – Parameterized co-processors,
  – ...

• DPM:
  – a customized software to characterize each component within the platform in order to optimally manage and configure platform resources
Dynamic platform management
General-Purpose Metamer

- PE granularity
  (usually imply # of functionalities)
- Interconnection routability
  - neighbor (1-D) / mesh (2-D)
  - crossbar
  - bus
- Initialization mechanism
- …
**PE Granularity**

- Smallest unit of the reconfigurable fabric that can be reprogrammed
- Tradeoffs between flexibility and reconfiguration overhead
Outline

• Introduction
• Configurable Platform
• Example
• Conclusion
SystemC-Based Computing Platform for Configurable Baseband SoC

Radio Processor

Advanced Signal Processing Dept.
SoC Technology Center
Industrial Technology Research Institute
Outline

• Introduction
• A Configurable Architecture
• A Configurable Coprocessing Datapath
• Virtual Prototyping
• Summary
Trend

- Platform-based Configurable Computing Platform for Multi-Standards Baseband SoC
  - IP reuse
  - Hardware/Software Co-design
  - A Quantitative approach
Alternative Computing Platform

- Control-dominated subsystem
  - controls & coordinates system tasks
  - performs reactive tasks (e.g. user interface)
- Data-dominated subsystem
  - regular & predictable transformational tasks
  - well-defined DSP kernels with high parallelism
Radio Processor -
A Re-Configurable OFDM-Based PHY Processor

- A re-configurable OFDM-based PHY processor suitable for several systems is proposed:
  - DAB/DVB
  - XDSL
  - WLAN
Proposed Configurable Architecture

Radio Processor Subsystem

Command Interpreter

Global Control Unit

Test Register

Data Interconnection Network Controller

Switch Box

Local Control Unit

I/F Control Unit

Function unit kernel

# n

I/F Control Unit

Local Control Unit

Function unit kernel

# 1

I/F Control Unit

Local Control Unit

Function unit kernel

# 2

I/F Control Unit

Local Control Unit

Function unit kernel

(# of functions, data grain, time limit.....)

Main Processor

Task Command

AHB Bus Interface

Task

2003/12/12 Platform-based SoC Design 30
**Different Types of FUs**

- **Configurable data processing FU**
  - The flexibility is limited, so is the configuration overhead.
  - The performance is closed to ASIC
  - Configurable outer receiver

- **Programmable data processing FU**
  - Bit/Stream-based DSP-lite co-processor
  - Programmable inner receiver

- **Configurable data routing FU**
  - Interconnection data network

- **Programmable control FU**
  - Used to control the simple procedure between FUs
Length-Scalable, Latency-Specified $64 \sim 2^n$ Point FFT/IFFT Function Unit

![Block Diagram of FFT/IFFT Function Unit](image)
Constraint Length-Scalable Soft Input Soft Output Viterbi Decoder Function Unit

Address generator

Address rotator

RAM-A

RAM-B

RAM-C

RAM-D

Data Left Rotator

Reg

Reg

Reg

Data Right Rotator

Reg

Reg

Reg

Reg

Data

Left

Rotator

Data

Right

Rotator

Radix-2 butterfly

Radix-2 butterfly

PE

Branch Metric Calculation

Hybrid Trace-Back Unit

decoded data
Scalable M-Input / N-Output Data Interface

Bit Counter: Indicate How Many Valid bits Within the Interface
Data In: Data Input of Interface
Data Out: Data Output of Interface
Out Pointer: Indicate Where to Start to Transfer the Output Data
In Pointer: Indicate Where to Start to Store the Input Data
Features of Radio Processor

- Heterogeneous Configurable Architecture
- Scalable Infrastructure with Unified Design Environment
- Customizable Design
- Flexible Instruction Set
- Limited Main-Processor Resource Requirement
Proposed Channel Processor

- Controller
- Task Interpreter
- Configurable Interconnection Network
- Storage Block
- Functional Units

Channel Processor

- Configurable DSP Datapaths
  - Clock recovery
  - Diversity Selection
  - Timing Sync.
  - Initial cell search
  - Channel estimation
  - Multi-user interface canceller
  - Rake receiver
  - ...

FY92
Channel Processor

- **Task Interpreter (controller)**
  - The **controller** coordinates the system tasks and all interfaces.
  - **Data-driven datapaths** (Adders, MACs, Complex multipliers...) as slave accelerators attached to the controller.
  - **Task Interpreter** translates the original control-originate task into dataflow and drives the scalable function units.
  - The **configurable stream interface unit** is an embedded memory together with configurable interconnection routing that interacts with the task interpreter.
  - The computation time of the undertaken task is predictable such that it is easy scheduling for the subsystem.

- **Configurable Stream Interface Unit (CSIU)**

- **Scalable Function Units (data-driven datapaths)**
Scalable Performance

- The number of configurable coprocessing datapaths are *scalable* in order to meet the performance requirement.
- Each configurable coprocessing datapath can be *reconfigured* for different DSP application.
- The performance boost is achieved by parallel processing via *SIMD-like* function units.
The Design Flow

- The data-driven acceleration is well synchronized with the micro-controller.
- The task ideally suited for coprocessing share the following characteristics:
  - Computationally intensive
  - Very little use of input, output, or internal registers
- The HW/SW interfacing could be automatic generated.
- Data movement is completely controlled by MCU (i.e. no explicit data coherence mechanism is required)

Channel Processor

Configurable DSP Datapaths
- Clock recovery
- Diversity Selection
- Timing Sync.
- Initial cell search
- Channel estimation
- Multi-user interface canceller
- Rake receiver
- ....

Micro-Controller

C/C++ Code (Application-specific)
Pre-processing (Compilation/Profiling)
Kernel Loop Extraction (Task Dispatch)
Source Code Replacement (HW/SW Interfacing)
Post-processing (machine code on CPU)

Data-Driven Acceleration

Task Pre-Analysis/Pre-Transformation (Control/Data Dependent Graph)
Primitive Function Unit Determination
Data Scheduling/Datapath Partitioning (Load Balancing)
Dataflow Control Optimization
Optimal Task Datapath Mapping
Coprocessing Datapath Generation

HW/SW Co-Simulation/Co-Verification
A Unified Wireless Open Platform

Analog RF Circuits

Communication Algorithms

Protocols

Reconfigurable Transceiver - RF Front-end + Programmable ADC/DAC

Channel Processor

Configurable DSP Datapaths

- Clock recovery
- Diversity Selection
- Timing Sync.
- Initial cell search
- Channel estimation
- Multi-user interface canceller
- Rake receiver
- ...

Radio Processor

Dedicated H/W Accelerators

- Spreading
- Scrambling
- Interleaving
- Channel CODEC
- Radio MoDem
- FFT
- ...

Network Processor

Framing

μP core (ARM)

RTOS
Keypad, Display
API, Control
SIG

DSP cores

ALU
FSM
Coders
Proposed Architecture
Proposed Design Flow (draft)
HW & SW Co-Verification

ARMulator

Debugger

DSM

Performance
observer

Test
Bench

Virtual Prototyping

Fast Rapid Prototyping

AMBA (systemC)

AMBA (Verilog)

AMBA (FPGA)

ARM core

Debugger

RDI

JTAG
Virtual Prototyping

• A software platform to support **HW/SW co-verification environment**.
  – Synthesizable AMBA SystemC model
    • Bus function model
    • Generic AMBA platform
    • Instruction Set Simulator
  – Performance Observer
The Platform

Bus Function Model:
Create sequence of transactions on the bus.
Overview of Virtual Prototyping

- Components
  - AHB components
  - APB Bridge
  - APB components
  - EMC
  - TIC
  - Performance Observer
- Behavioral Modules
  - Memory Modules
  - TUBE
  - Tic box
  - Test bench transfer tools
Features

• **Bus/Processor independent**
  – Low requirement on processor computation power. (move data only)
  – Bus substitution can be achieved by modifying the AHB slave interface only.

• **Flexible architecture**
  – Expandable architecture simplify the mounting of new function units.
  – Distributed command decoding scheme.
  – Numerous well built configurable function units are provided.
  – A highly flexible function unit dedicated for inner transceiver algorithm will be available this year.

• **Complete simulation and verification environment**
  – SystemC environment is well built except the ISS.
  – Verilog environment is complete including ARM.
  – Bus function model (TIC box) is available in both SystemC and Verilog version.

• **Silicon proofed**
  – Test chip @ TSMC 0.18 CMOS technology, with 750k gate count (not including SRAM module)
**Features (Cont.)**

- **A transceiver PHY platform:**
  - Available function units currently:
    - 1. Scrambler/De-scrambler.
    - 2. FEC / Viterbi decoder: 1/2, 2/3, 9/16, 3/4 coding rates with 64 states Viterbi decoder.
    - 4. Modulator: BPSK/QPSK/16QAM/64QAM.
    - 5. FFT: 64/256/2048/4096 points FFT.
    - 6. I/O Interface.
  - An array based processing unit targeted on inner transceiver of communication system will be built this year.

- **An easy algorithm to hardware design platform:**
  - A design and verification methodology from algorithm to RTL via SystemC is provided.
  - Co-simulation can be performed in SystemC and Verilog using ARM ISS or bus function model (**Test-Interface-Controller**).
Benefits

- **Simplify backbone design:**
  - A platform provides an architecture reference which is proved to be a applicable architecture.
  - Simple modification is enough to be suitable to similar systems.

- **Save repetitive design time:**
  - Existing IPs for the platform can be adopted to accelerate the build up time.
  - Based on existing platform ease the replace of custom design.

- **Ease the verification:**
  - The environment provided by a platform helps to verify the custom modification in each step.

- **Early evaluation:**
  - A virtual prototyping provides early system performance data and H/S partitioning information.
Conclusion

• Platform-based design facilitates complex system designs in SOC
  – Reuse a platform architecture, hardware IPs, interconnection, and software IPs
  – Simplify design method and techniques
  – Focus on system design, analysis, optimization, and verification
  – Reduce design time, cost, and risks