EE382V: Embedded System Design and Modeling

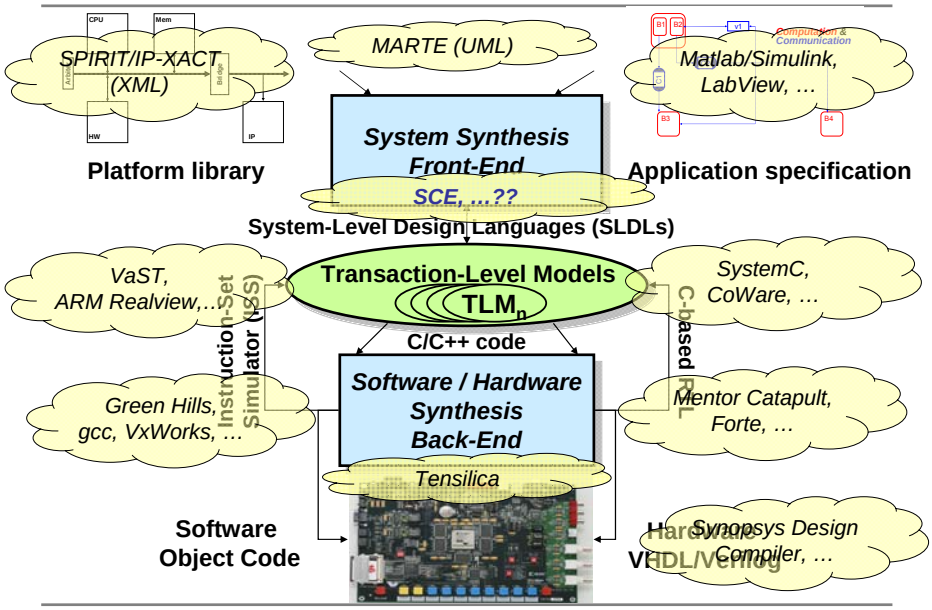
Lecture 8 – System-Level Design Tools

Andreas Gerstlauer
Electrical and Computer Engineering
University of Texas at Austin
gerstl@ece.utexas.edu

Lecture 8: Outline

- **System-level design tools**
  - Commercial tools
  - Academic tools
- **SCE design examples**
  - Baseband platform
  - MP3 decoder
  - Cellphone
- **SCE commercialization**
  - Specify-Explore-Refine (SER) environment

Electronic System-Level (ESL) Landscape



ESL Tools

- **Electronic System-Level (ESL) terminology misused**
  - Often single hardware unit only (high-level HW synthesis)
- **System-level has to span across hardware and software**
  - System-level frontend
  - Hardware and software synthesis backend

➢ **Commercial tools for modeling and simulation**
- Algorithmic modeling (MoC) [UML, Matlab/Simulink, Labview]
- Virtual system prototyping (TLM) [Coware, VaST, Virtutech]
  - ➢ *Only horizontal integration across models / components*

➢ **Academic tools for synthesis and verification**
- MPSoC synthesis [SCE, Metropolis, SCD, PeaCE, Deadalus]
  - ➢ *Vertical integration for path to implementation*

Commercial Tools (1)

- **CoFluent**
  - SystemC-based modeling and simulation
    - Networks of timed processes
    - Communication through queues, events, variables
  - Early, high-level interactive design space exploration
    - Graphical application, architecture and mapping capture
    - Fast TLM simulation with estimated timing
- **Space Codesign**
  - Graphical application, architecture and mapping capture (Eclipse)
    - Process network with message-passing or shared-memory channels
  - SystemC TLM simulation
    - Annotated, host-compiled or cycle-accurate ISS models
  - FPGA-based prototyping
    - Cross-compilation and third-party hardware synthesis (Forte/Catapult)

Commercial Tools (2)

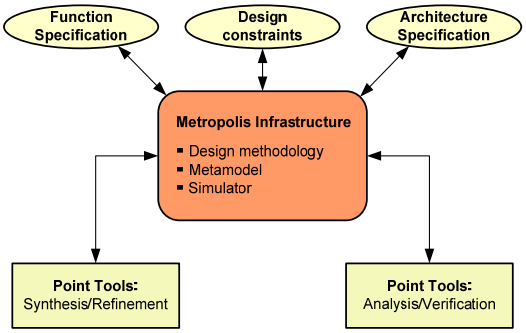
- **CoWare**
  - Virtual system platforms
    - SystemC TLM capture, modeling and simulation
    - Extensive library of IP, processor and bus models
    - Application-specific processor ISS models (LISAtek acquisition)
  - Proprietary SystemC simulation framework
    - Optimized SystemC kernel
    - Graphical debugging, visualization and analysis capabilities
- **Soc Designer**
  - Proprietary, C++ based modeling and simulation
    - Fast, statically scheduled cycle-accurate simulation
    - Special cycle-callable component models
- **VaST and Virtutech**
  - Proprietary SW-centric virtual platform modeling and simulation
    - Fast, cycle-approximate binary translated or compiled ISS + peripherals

Academic Tools

- **Metropolis**
  - Platform-based design (PBD)
- **SystemCoDesigner**
  - Dynamic dataflow MoC
  - Automated design space exploration
- **Daedalus**
  - KPN MoC for streaming, multi-media applications
  - IP-based MPSoC assembly
- **PeaCE**
  - "Ptolemy extension as a Codesign Environment"
  - Recent extensions for software development (HoPES)
- **SCE**
  - SpecC-based "System-on-Chip Environment"
  - Successive, stepwise Specify-Explore-Refine methodology

Academic Tools: Metropolis

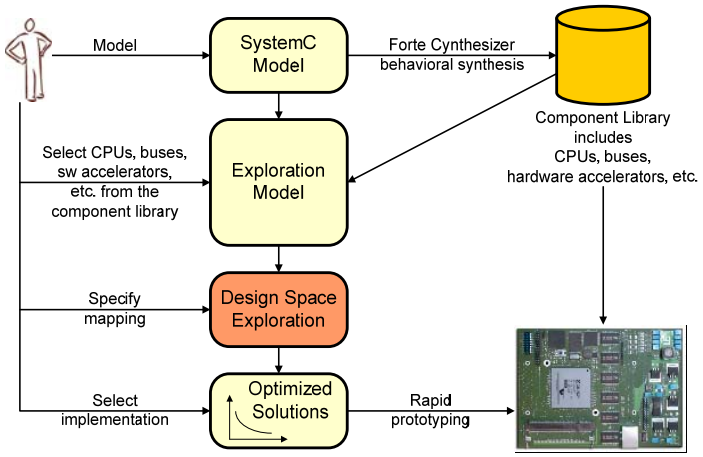
- **Platform-based**
  - Pre-defined target architecture
  - ➢ Reuse
- **Meet-in-the-middle**
  - Platform mapping and configuration



- **General, proprietary meta-modeling language**
  - Capture function, architecture and mapping
- **Modeling framework**
  - Built-in parsing and simulation
  - Back-end point tool integration

Academic Tools: SystemCoDesigner

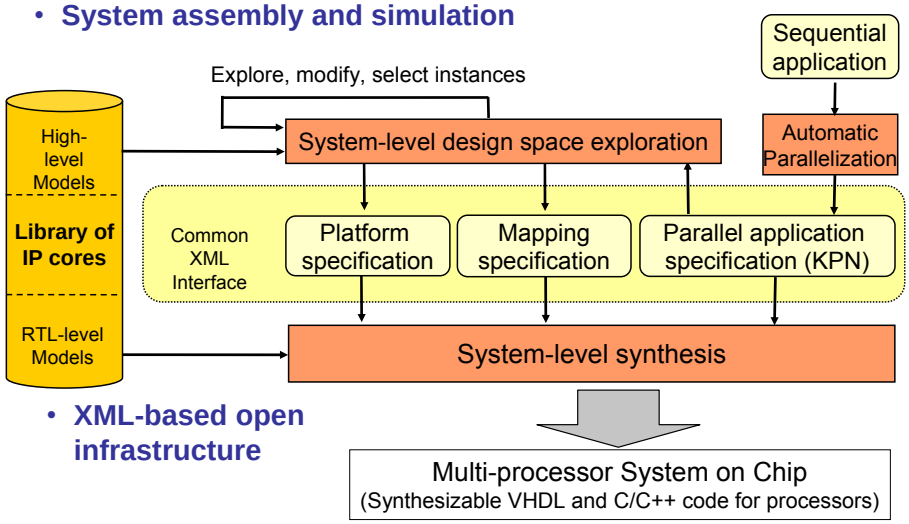
- **SysteMoC input model**
  - Dynamic dataflow MoC (actors + FSMDs) in SystemC
- **Fully automatic, multi-objective design space exploration**
  - Multi-objective evolutionary algorithms (MOEAs)



EE382V: Embedded Sys Dsgn and Modeling, Lecture 8 © 2009 A. Gerstlauer 9

Academic Tools: Daedalus

- **KPN input model**
- **System assembly and simulation**

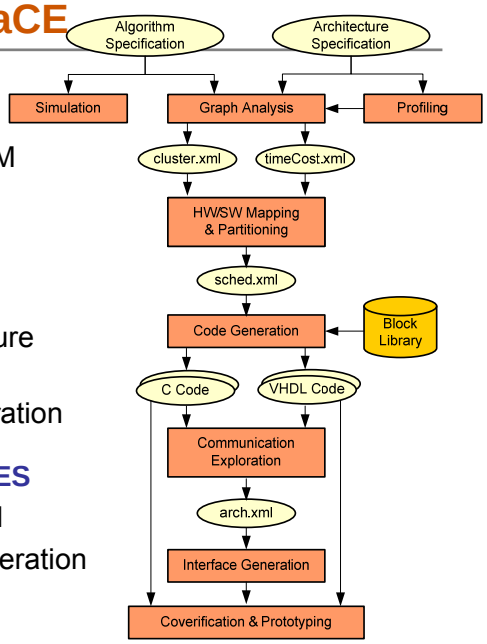


- **XML-based open infrastructure**

EE382V: Embedded Sys Dsgn and Modeling, Lecture 8 © 2009 A. Gerstlauer 10

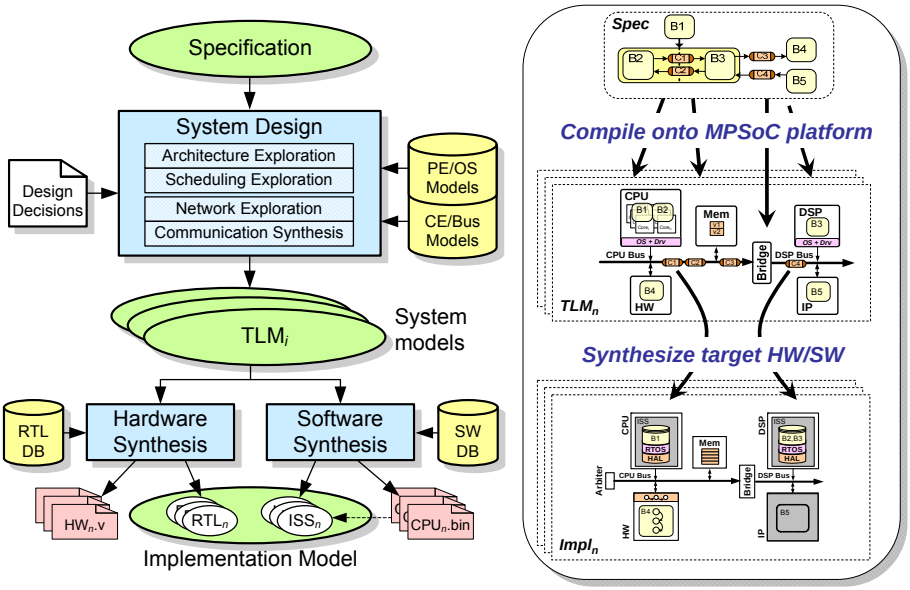
Academic Tools: PeaCE

- **Ptolemy-based**
  - Heterogeneous SDF+FSM application MoC
- **Stepwise flow**
  - Application partitioning
  - Communication architecture exploration
  - Code and interface generation
- **Software extensions: HOPES**
  - Parallel programming API
  - Multi-processor code generation



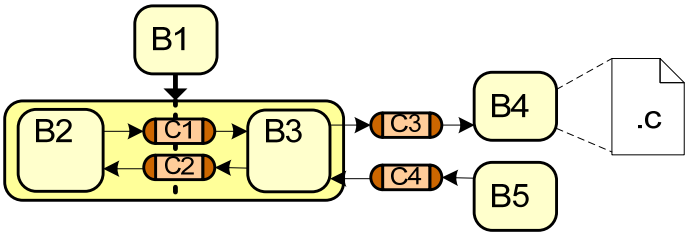
EE382V: Embedded Sys Dsgn and Modeling, Lecture 8 © 2009 A. Gerstlauer 11

System-On-Chip Environment (SCE)



EE382V: Embedded Sys Dsgn and Modeling, Lecture 8 © 2009 A. Gerstlauer 12

Abstract Programming Model

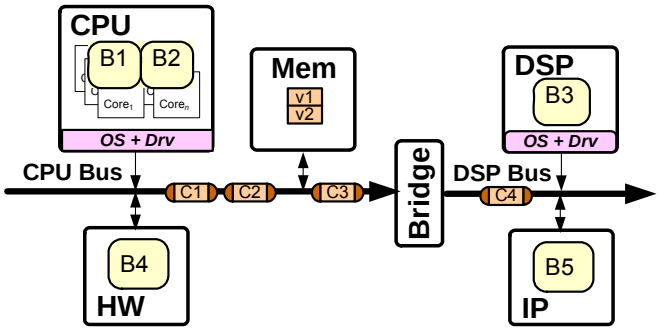


- **Hierarchical process graph**
  - Sequential processes
    - ANSI C code
  - Parallel-serial composition
    - Dependencies, Fork-join
- **Abstract inter-process communication**
  - Communication channels
    - Message-passing, queues, etc.
  - Shared variables

EE382V: Embedded Sys Dsgn and Modeling, Lecture 8 © 2009 A. Gerstlauer 13

System Transaction-Level Model (TLM)

- **Compile onto multi-core/multi-processor platform**
  - Implement computation on processors/cores and busses
  - Generate code for communication over bus network

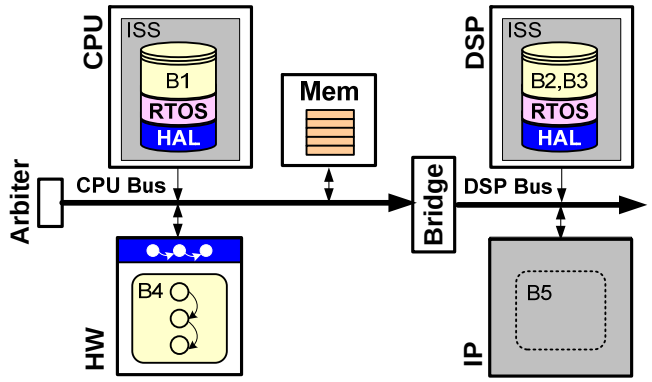


- **Generate MPSoC TLM simulation model**
  - Fast and accurate for exploration and verification

EE382V: Embedded Sys Dsgn and Modeling, Lecture 8 © 2009 A. Gerstlauer 14

System Implementation

- **Synthesize hardware and software for each processor**
  - High-level/behavioral RTL and interface synthesis
    - Allocation, scheduling, binding
  - Software synthesis and RTOS targeting
    - Code generation, firmware synthesis, cross-compile & link



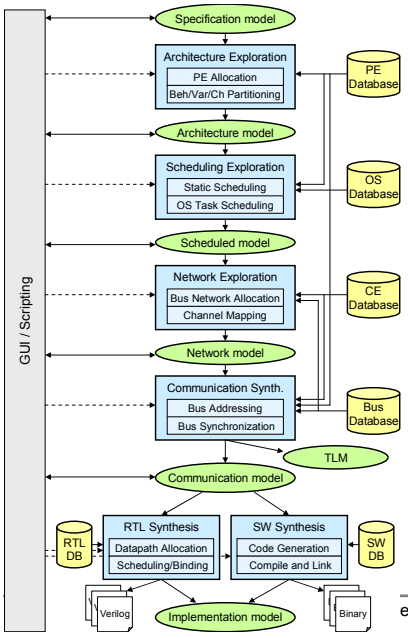
Academic MPSoC Design Tools

Approach	DSE	Comp. decision	Comm. decision	Comp. refine	Comm. refine
Daedalus	●	●	○	●	○
Koski	●	●	○	●	○
Metropolis		○		○	
PeaCE/HoPES	○	○		●	○
SCE				●	●
SystemCoDesigner	●	●	●	●	

Lecture 8: Outline

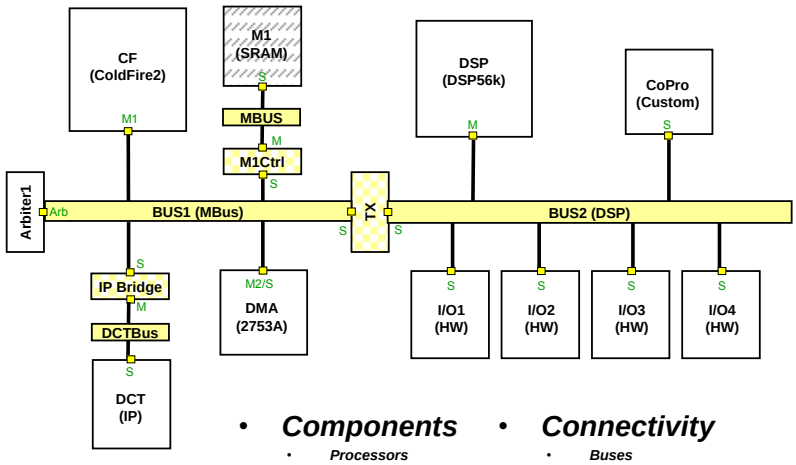
- ✓ System-level design tools
  - ✓ Academic tools
  - ✓ Commercial tools
- **SCE design examples**
  - Baseband platform
  - MP3 decoder
  - Cellphone
- SCE commercialization
  - Specify-Explore-Refine (SER) environment

System-On-Chip Environment (SCE)



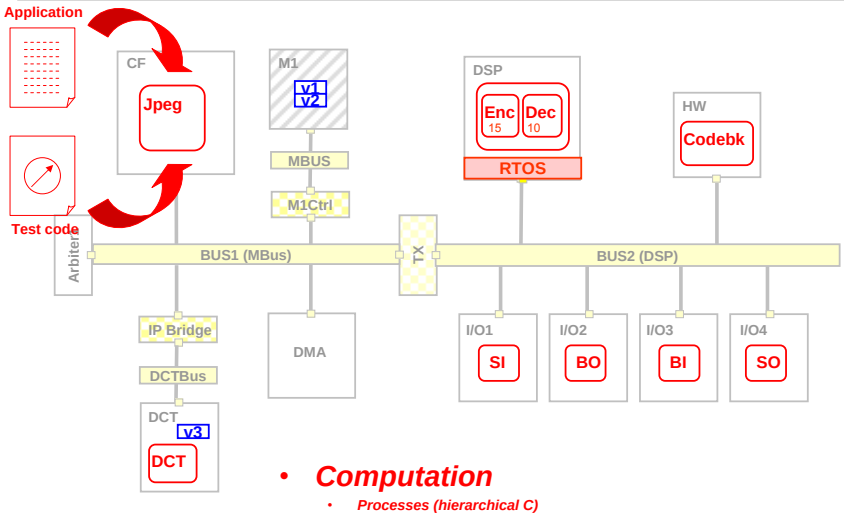
- **SCE Components:**
  - Graphical frontend (`sce, scchart`)
  - Editor (`sced`)
  - Compiler and simulator (`scc`)
  - Profiling and analysis (`scprof`)
  - Architecture refinement (`scar`)
  - RTOS refinement (`scos`)
  - Network refinement (`scnr`)
  - Communication refinement (`sccr`)
  - RTL refinement (`scrtl`)
  - Software refinement (`sc2c`)
  - Scripting interface (`scsh`)
  - Tools and utilities ...

Baseband Example: Platform Architecture



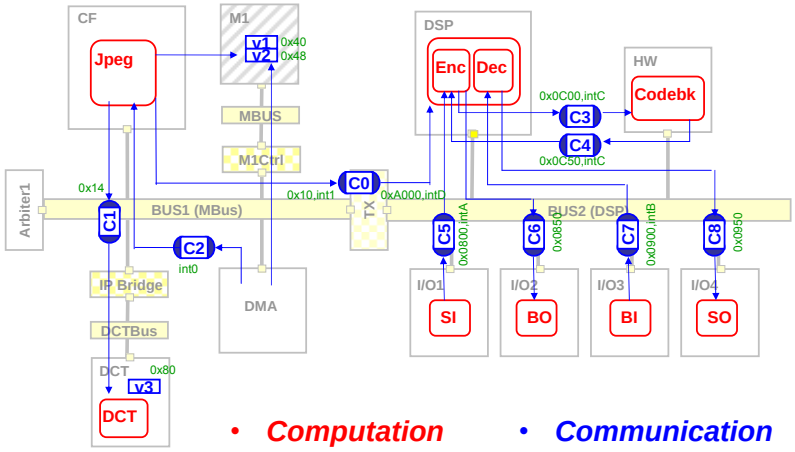
- ***Components***
  - *Processors*
  - *Memories*
  - *IPs*
  - *Custom HW*
- ***Connectivity***
  - *Buses*
  - *Bridges*
  - *Ports*

Baseband Example: Application, Services



- ***Computation***
  - *Processes (hierarchical C)*
  - *Scheduling (static/OS)*

Baseband Example: Application, Services

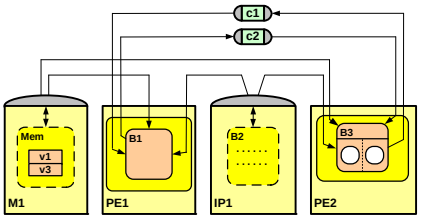


- **Computation**
  - Processes (hierarchical C)
  - Scheduling (static/OS)
- **Communication**
  - High-level IPC (channels)
  - Storage (variables)

Computation Design (1)

- **Architecture exploration**
  - Allocation of Processing Elements (PE)
    - Type and number of processors
    - Type and number of custom hardware blocks
    - Type and number of system memories
  - Mapping to PEs
    - Map each behavior to a PE
    - Map each complex channel to a PE
    - Map each variable to a PE

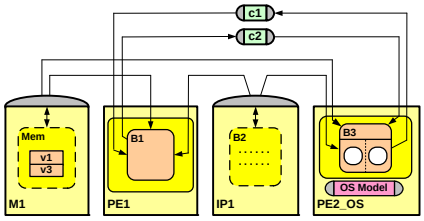
➢ **Architecture model**
  - ➢ Concurrent PEs
  - ➢ Abstract channels and memory interfaces



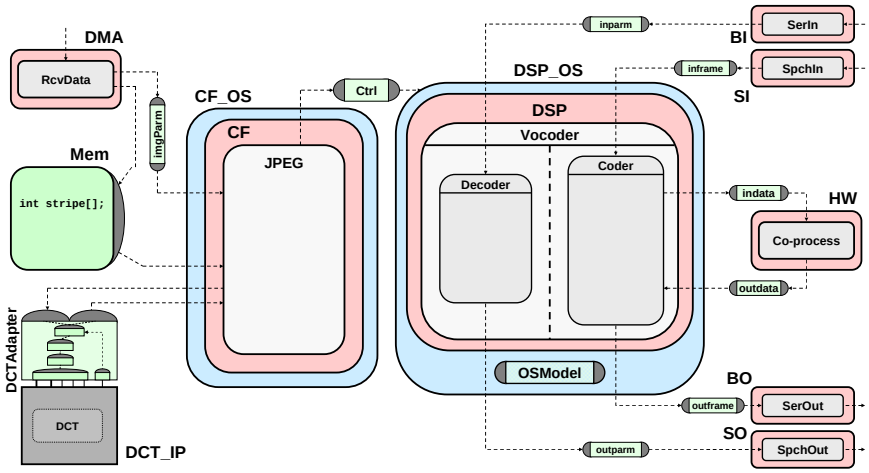
Computation Design (2)

- **Scheduling exploration**
  - Static scheduling of behaviors into sequential tasks
    - Group (flatten) behaviors into tasks
    - Determine fixed execution order of behaviors in each task
  - Dynamic scheduling of concurrent tasks by RTOS
    - Choose scheduling policy, i.e. round-robin or priority-based
    - For each set of tasks, determine task priorities

➢ **Scheduled model**
  ➢ Abstract OS model in software PEs



Baseband Example: Scheduled Model

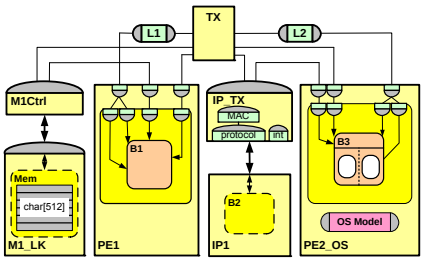


➢ **Partitioning, synchronization, message-passing, RPC**
➢ **Scheduling, task refinement, OS model insertion**

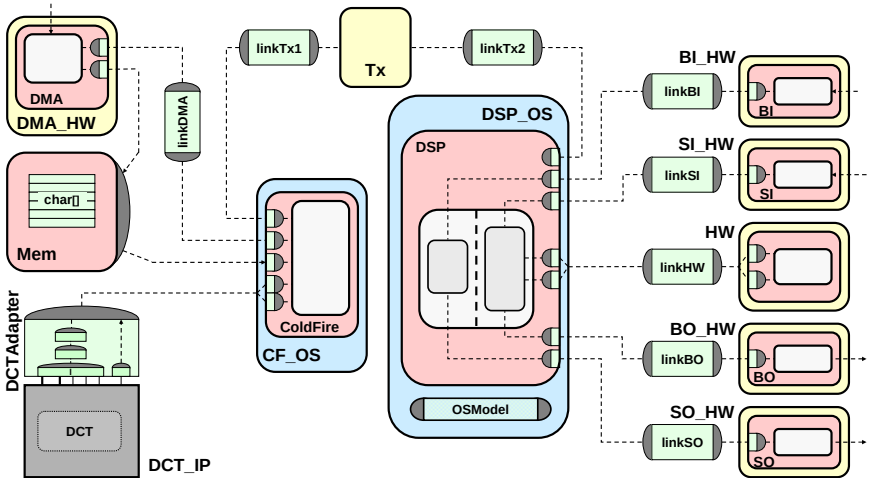
Communication Design (1)

- **Network exploration**
  - Allocation of system network
    - Type (protocols) and number of system busses
    - Type and number of CEs (bridges and transducers, if applicable)
    - System connectivity
  - Routing of channels over busses
    - Map each communication channel to a system bus (or an ordered list of busses, if applicable)

➢ **Network model**
  - ➢ PEs + CEs
    - ➢ Middleware stacks
  - ➢ Point-to-point links
    - ➢ Untyped packet transfers
    - ➢ Untyped memory interfaces



Baseband Example: Network Model

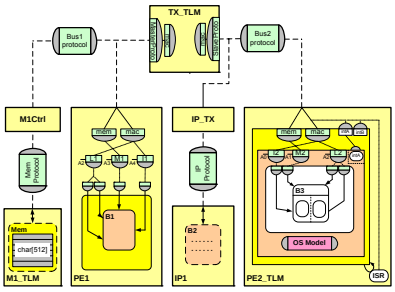


➢ **Data conversion, channel merging**
➢ **CE insertion, packeting, routing**

Communication Design (2)

- **Communication synthesis**
  - Assignment of bus parameters
    - Address mapping for each channel and memory interface
    - Synchronization mechanism for each channel (dedicated or shared interrupts, polling)
    - Transfer mode for each channel/interface (regular, burst, DMA)

➢ **Transaction-level model (TLM)**
- ➢ PEs + CEs + Busses
  - ➢ Protocol stacks
- ➢ Abstract bus channels
  - ➢ Bus transactions + interrupts

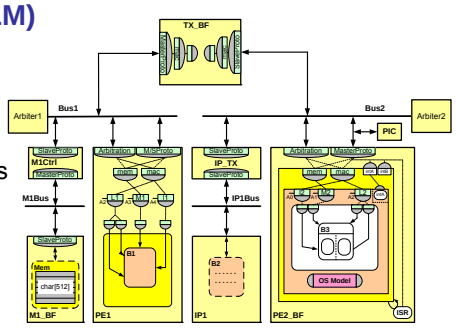


Communication Design (2)

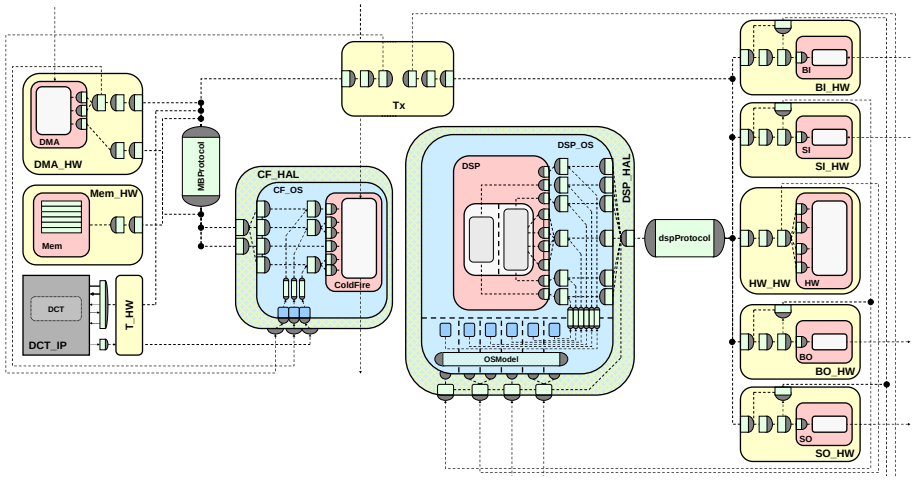
- **Communication synthesis**
  - Assignment of bus parameters
    - Address mapping for each channel and memory interface
    - Synchronization mechanism for each channel (dedicated or shared interrupts, polling)
    - Transfer mode for each channel/interface (regular, burst, DMA)

➢ **Transaction-level model (TLM)**
- ➢ PEs + CEs + Busses
  - ➢ Protocol stacks
- ➢ Abstract bus channels
  - ➢ Bus transactions + interrupts

➢ **Pin-accurate model (PAM)**
- ➢ Physical bus structure
  - ➢ Bit-accurate pins and wires

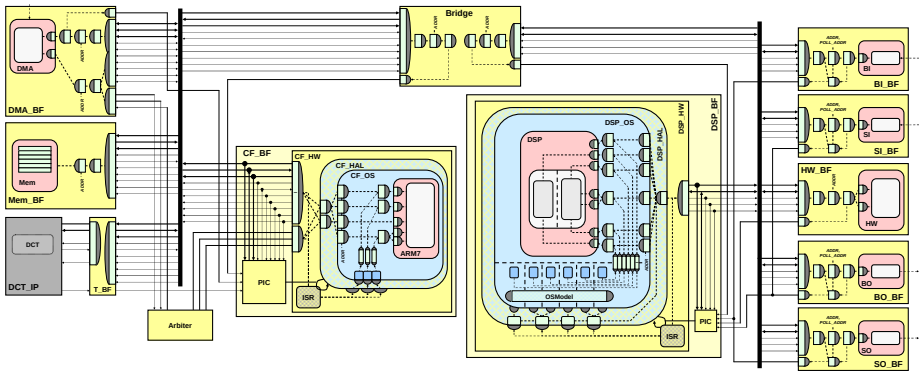


Baseband Example: TLM



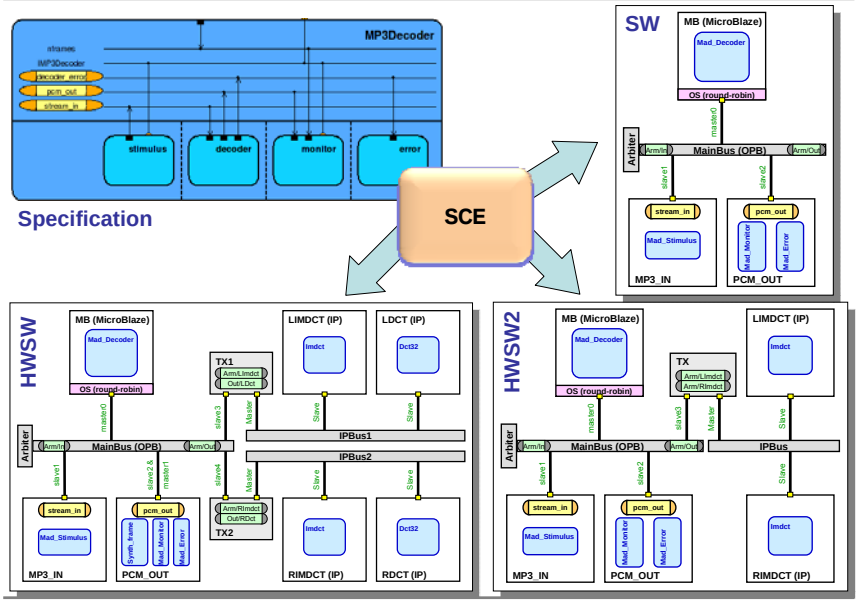
- Synchronization, addressing, media acces
- Arbitration, data slicing, interrupt handling

Baseband Example: Pin-Accurate Model



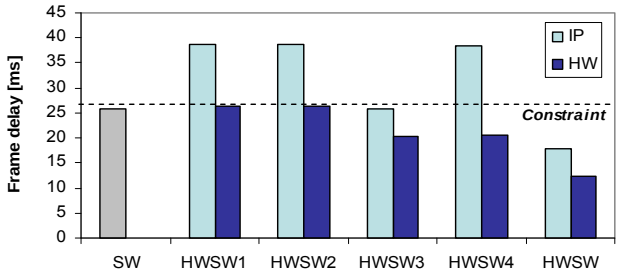
- Implementation synthesis in backend tools
  - Interface and high-level synthesis on hardware side
  - Firmware, RTOS and C synthesis on the software side

MP3 Decoder Example



MP3 Decoder Exploration Results

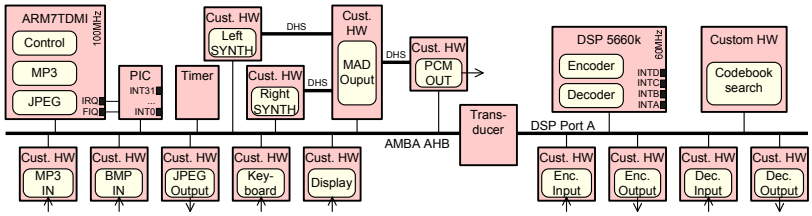
- **Single specification of application algorithm**
  - Fixed-point MP3 decoding algorithm [libmad]
- **Explored several architectural alternatives**
  - Pure SW solution
  - Custom HW or IPs for DCT/IMDCT acceleration
  - Parallelizing and pipelining of SW and HW/IP



- **Automatically generate models and simulate each**
  - Explore and validate all alternatives in < 1 hour
  - Ready to synthesize to FPGA prototyping board

Cellphone Example

- **2 Subsystems**
  - ARM7TDMI
    - MP3 Decoding
    - Jpeg Encoding
  - Motorola DSP 56600k
    - GSM Transcoding
- **4 Accelerator HW blocks**
- **10 I/O HW blocks**
- **5 Busses**
  - AMBA AHB
  - DSP Port A bus
  - 3 Custom busses



SCE Exploration Results

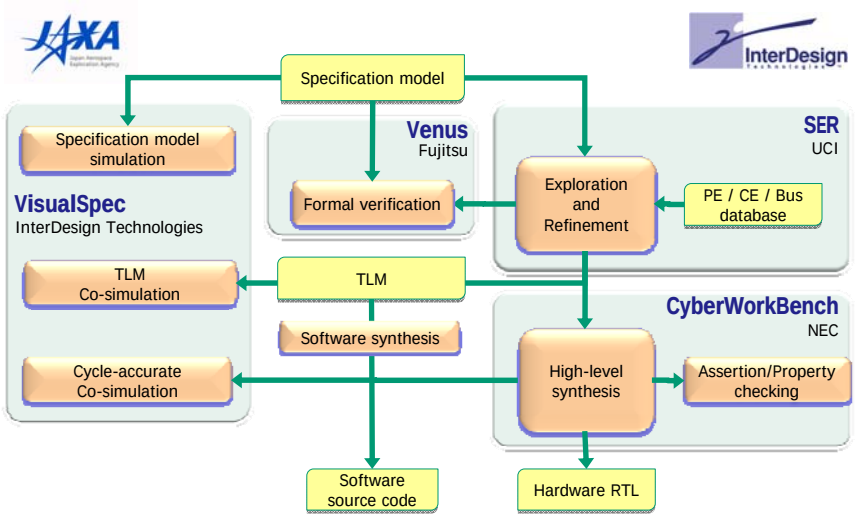
- **Suite of industrial size examples**

Example		System Architecture (masters → slaves)	Model size (LOC)				Generation time
			Spec	Arch	Net	PAM	
JPEG	A1	CF→HW	1806	2732	2780	4642	1.01 s
Vocoder	A1	DSP→HW	7385	9594	9775	10679	4.77s
	A2	DSP→HW1,HW2		9632	9913	10989	5.21s
	A3	DSP→HW1,HW2,HW3		9659	9949	11041	5.76s
Mp3float	A1	CF→HW1	6900	28190	28204	29807	5.86s
	A2	CF→HW1,HW2,HW3 HW1↔HW3 HW2↔HW3		28275	28633	31172	6.18s
	A3	CF →HW1, HW2, HW3, HW4 HW1↔HW3↔HW5 HW2↔HW4↔HW5		28736	30202	32795	17.69s
Mp3fix	A1	ARM→2 I/O	13363	17131	17270	21593	3.85s
	A2	ARM→2 I/O, LDCT, RDCT		18300	18564	23228	7.01s
	A3	ARM→2 I/O, LDCT, RDCT LDCT↔I/O RDCT↔I/O		18748	19079	24471	7.40s
Baseband	A1	DSP→HW, 4 I/O, T CF, DMA→Mem, BR, T, DMA BR→DCT_IP	11481	17020	17685	21711	8.99s
Cellphone	A1	ARM→4 I/O, 2 DCT, T LDCT, RDCT→I/O DSP→HW, 4 I/O, T	16441	21936	22570	30072	9.49s

Lecture 8: Outline

- ✓ System-level design tools
  - ✓ Academic tools
  - ✓ Commercial tools
- ✓ SCE design examples
  - ✓ Baseband platform
  - ✓ MP3 decoder
  - ✓ Cellphone
- **SCE commercialization**
  - Specify-Explore-Refine (SER) environment

ELEGANT Environment

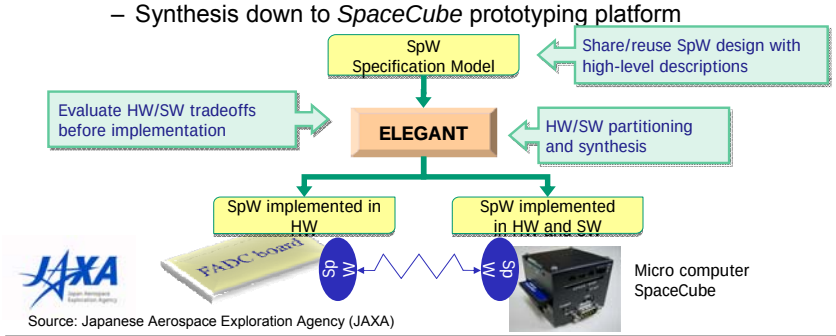


ELEGANT : Electronic Design Guidance Tool for Space Use

Source: InterDesign Technologies, Inc. / Japanese Aerospace Exploration Agency (JAXA)

ELEGANT *SpaceWire* Evaluation

- ***SpaceWire*: aerospace communication protocol standard**
  - High-speed and high-reliability interconnection network
    - Asynchronous, fault-tolerance, topology agnostic
- **Automated *SpaceWire* design with ELEGANT tool set**
  - From top-level specification model down to HW/SW
    - HW/SW partitioning and exploration of the architecture with SER
    - Synthesis down to *SpaceCube* prototyping platform



Source: Japanese Aerospace Exploration Agency (JAXA)

Lecture 8: Summary

- **System-level design tools**
  - Commercial focus still only on modeling and simulation
  - Academic approaches towards true system-level design
  - Emerging commercial backend HW/SW synthesis
  - ➢ Complete, automated system design flow
    - ➢ From specification to implementation
- **SCE design examples**
  - Industrial-size examples
- **SER commercialization**
  - Derivative of SCE
  - Integrated with tools for simulation, HW synthesis
  - Deployed in, e.g. NEC Toshiba Space Systems