Hardware-dependent Software Synthesis for Many-Core Embedded Systems

Samar Abdi, Gunar Schirner, Ines Viskic, Hansu Cho,
Yonghyun Hwang, Lochi Yu, Daniel Gajski

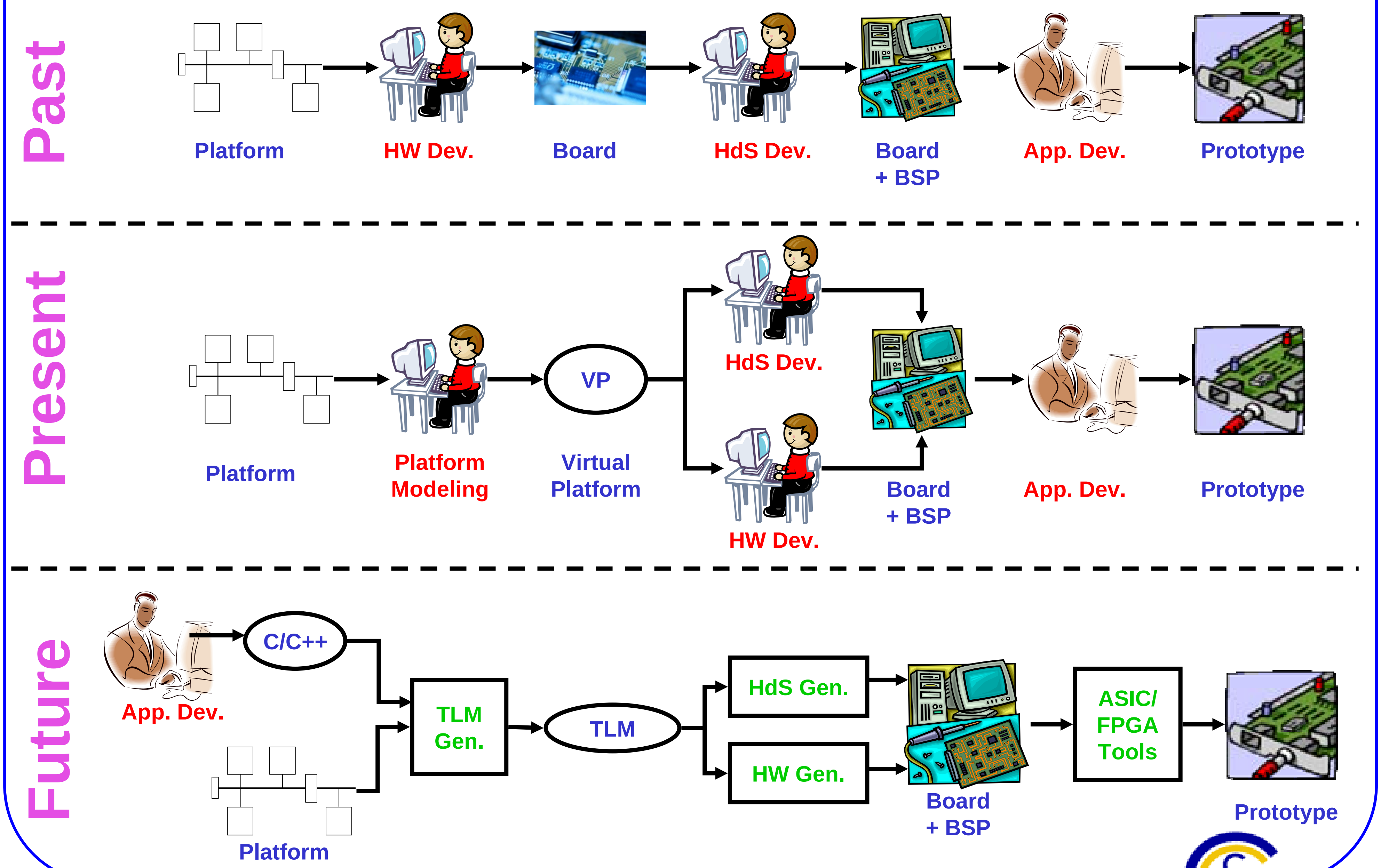
Presenter: Samar Abdi

Center for Embedded Computer Systems
University of California, Irvine
http://www.cecs.uci.edu

Outline

- Motivation for many-core HdS synthesis
- Model based approach to system synthesis
- HdS synthesis for TLM (Front-End)
- HdS synthesis for PCAM (Back-End)
- Embedded System Environment (ESE)
- Experimental results for MP3 decoder example
- Conclusion

System Design Trend



Past

Platform → HW Dev. → Board → HdS Dev. → Board + BSP → App. Dev. → Prototype

Present

Platform → Platform Modeling → VP (Virtual Platform) → HdS Dev. / HW Dev. → Board + BSP → App. Dev. → Prototype

Future

App. Dev. → C/C++ → TLM Gen.

Platform → TLM Gen.

TLM Gen. → TLM → HdS Gen. / HW Gen. → Board + BSP → ASIC/ FPGA Tools → Prototype

Motivation for Many-Core HdS Synthesis

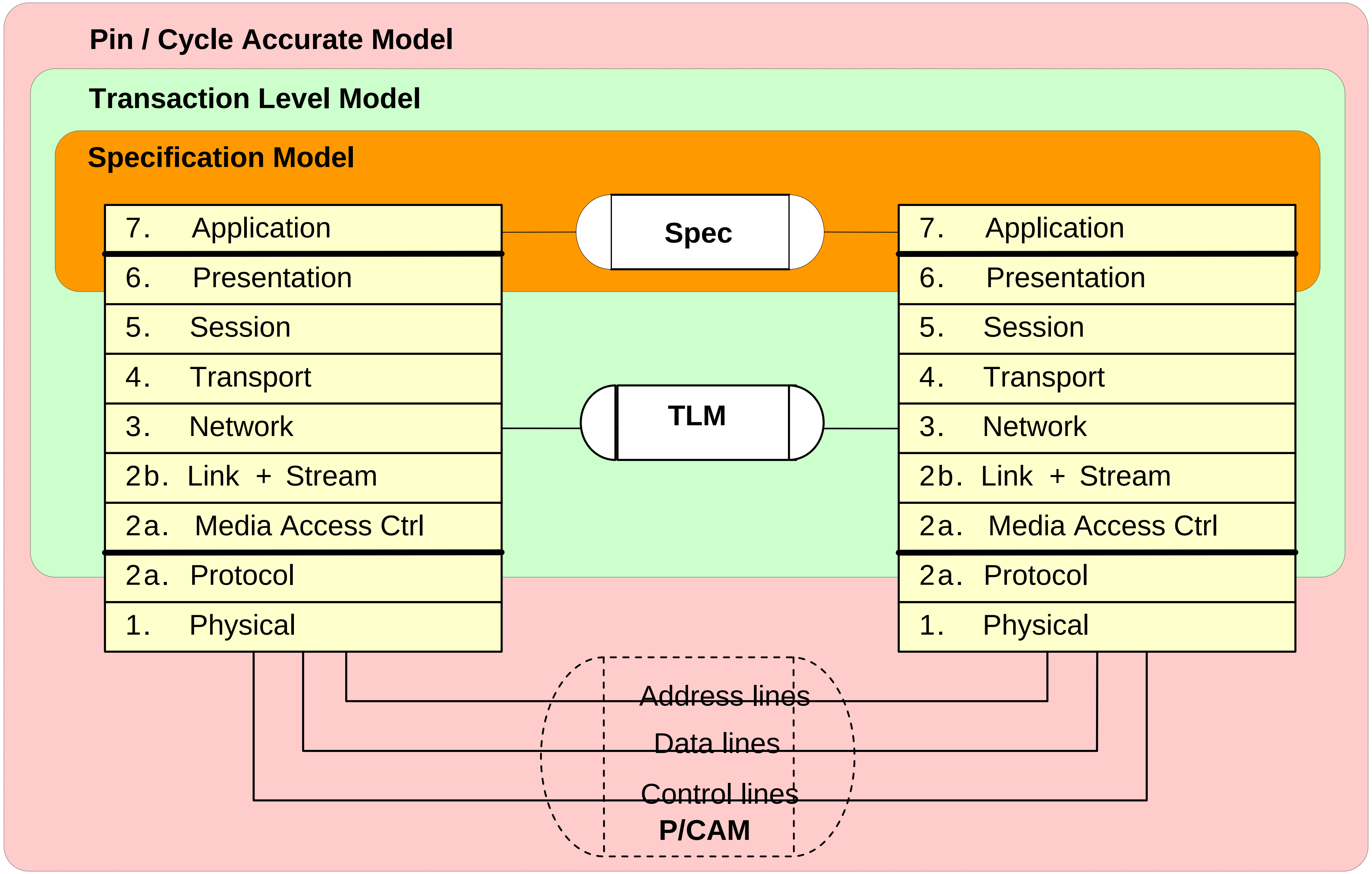
- Multi-core and many-core HW platforms
  - Technology scaling is limited
  - Better throughput, lower power with parallel execution
  - Heterogeneous systems required for efficient execution
- Automatic HdS synthesis
  - Multipurpose, complex applications
  - Flexible HW platforms that vary across application domains
  - Short time to market
- Approach: Model based synthesis
  - Executable models at various abstraction levels
  - Formalized model semantics
  - Layered HdS structure

Model Based Synthesis

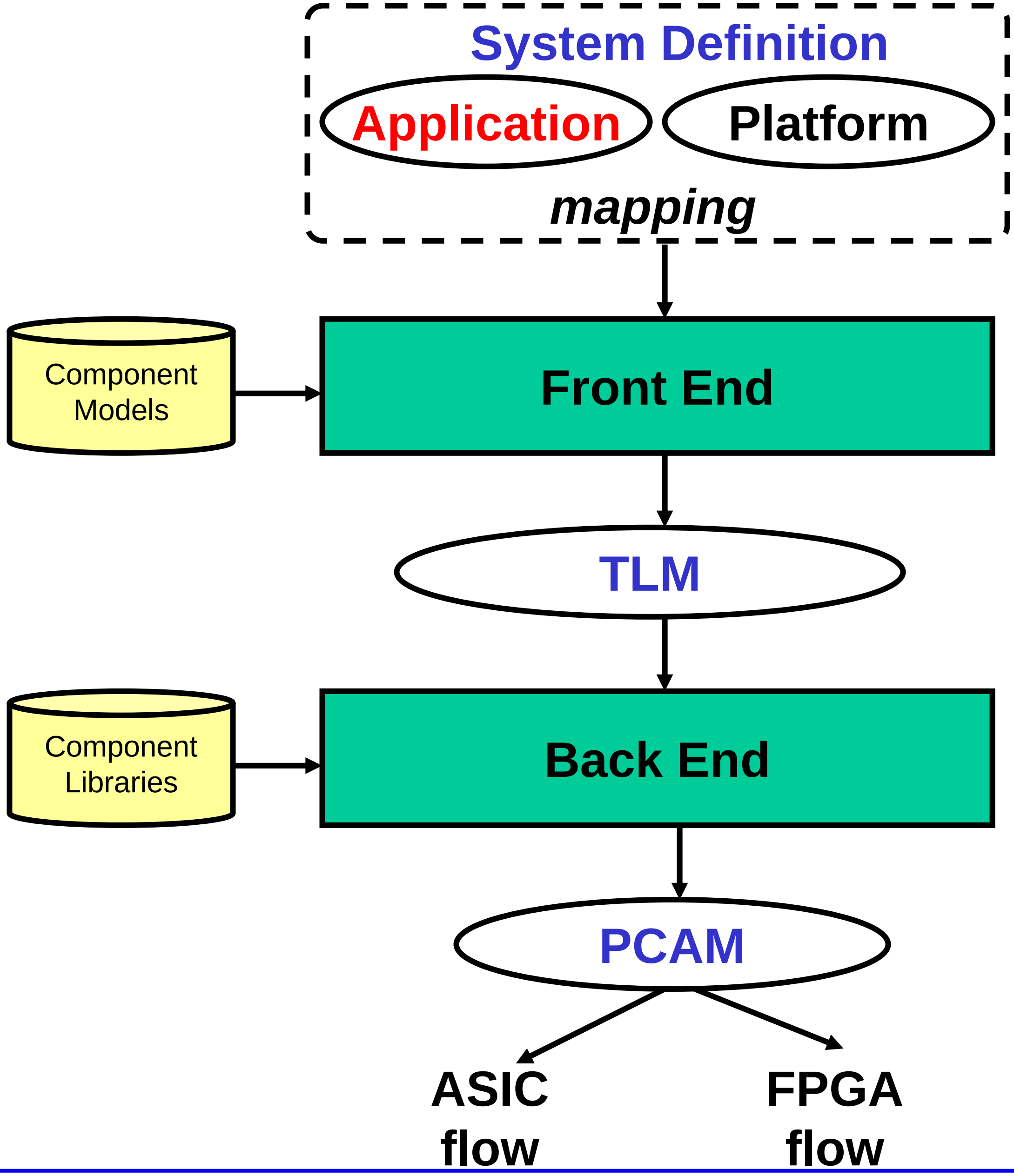
- Benefits
  - Faster design/simulation/validation at higher abstraction
  - Early application development
  - Rapid design space exploration
  - Automatic synthesis
    - reduces probability of error vs. manual implementation
    - Increases designer productivity

- Challenges
  - Identifying the “right” abstraction levels
  - Formal model semantics at each abstraction level
  - Methods and tools for
    - Application development
    - SW/HW synthesis
    - Model debugging and analysis

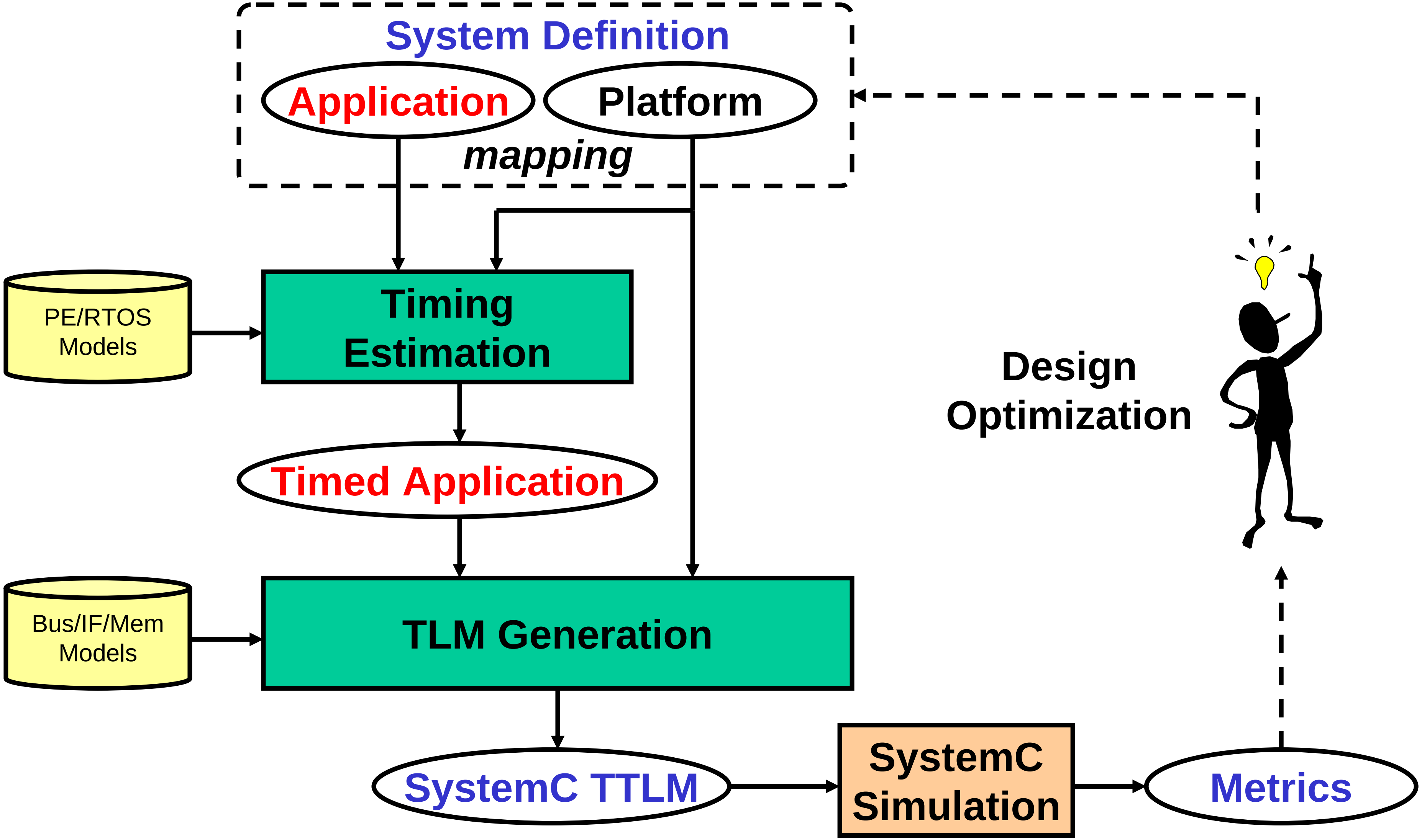
Model Abstractions (with Respect to OSI)



System Synthesis Flow



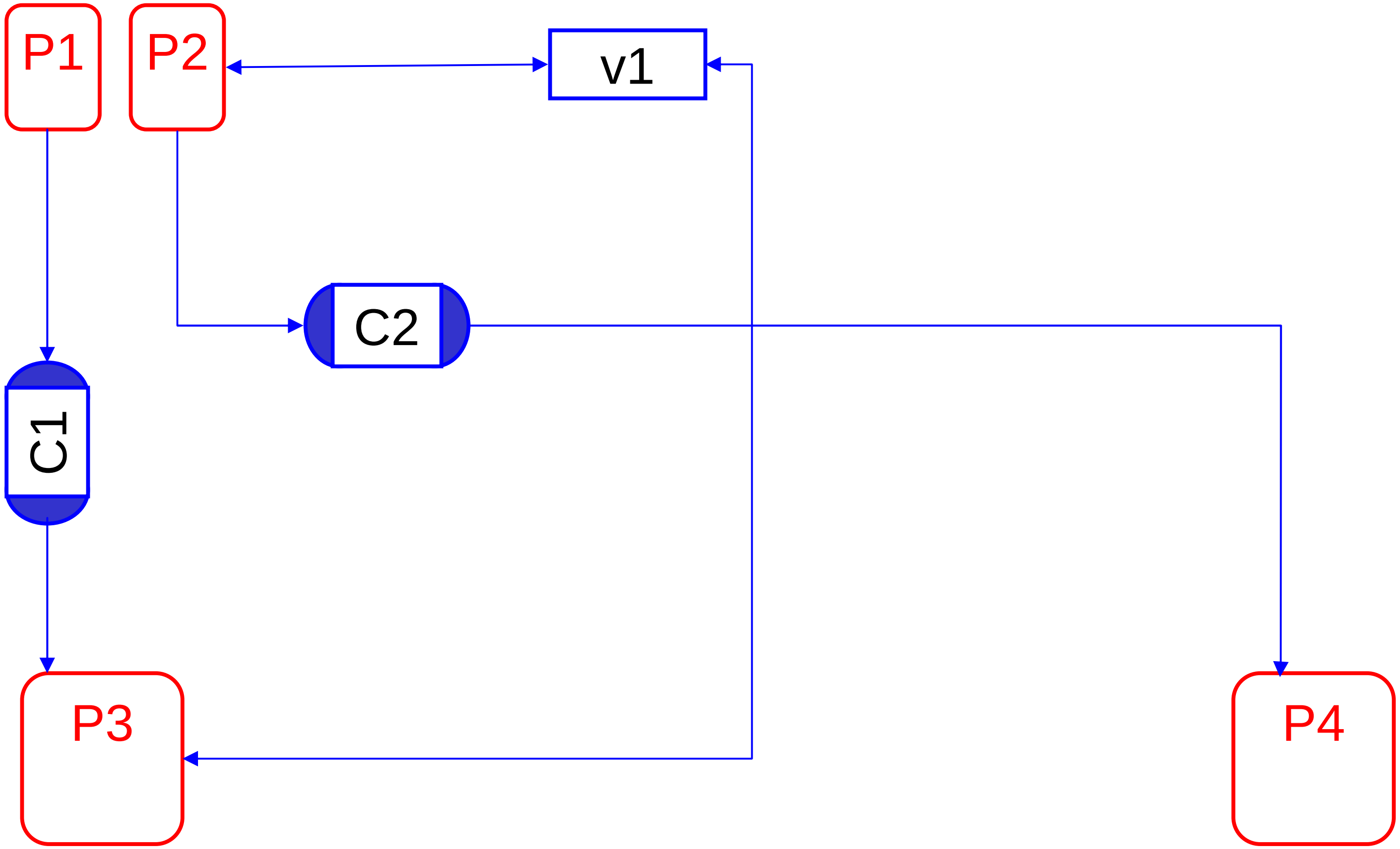
Front End Design Flow



Outline

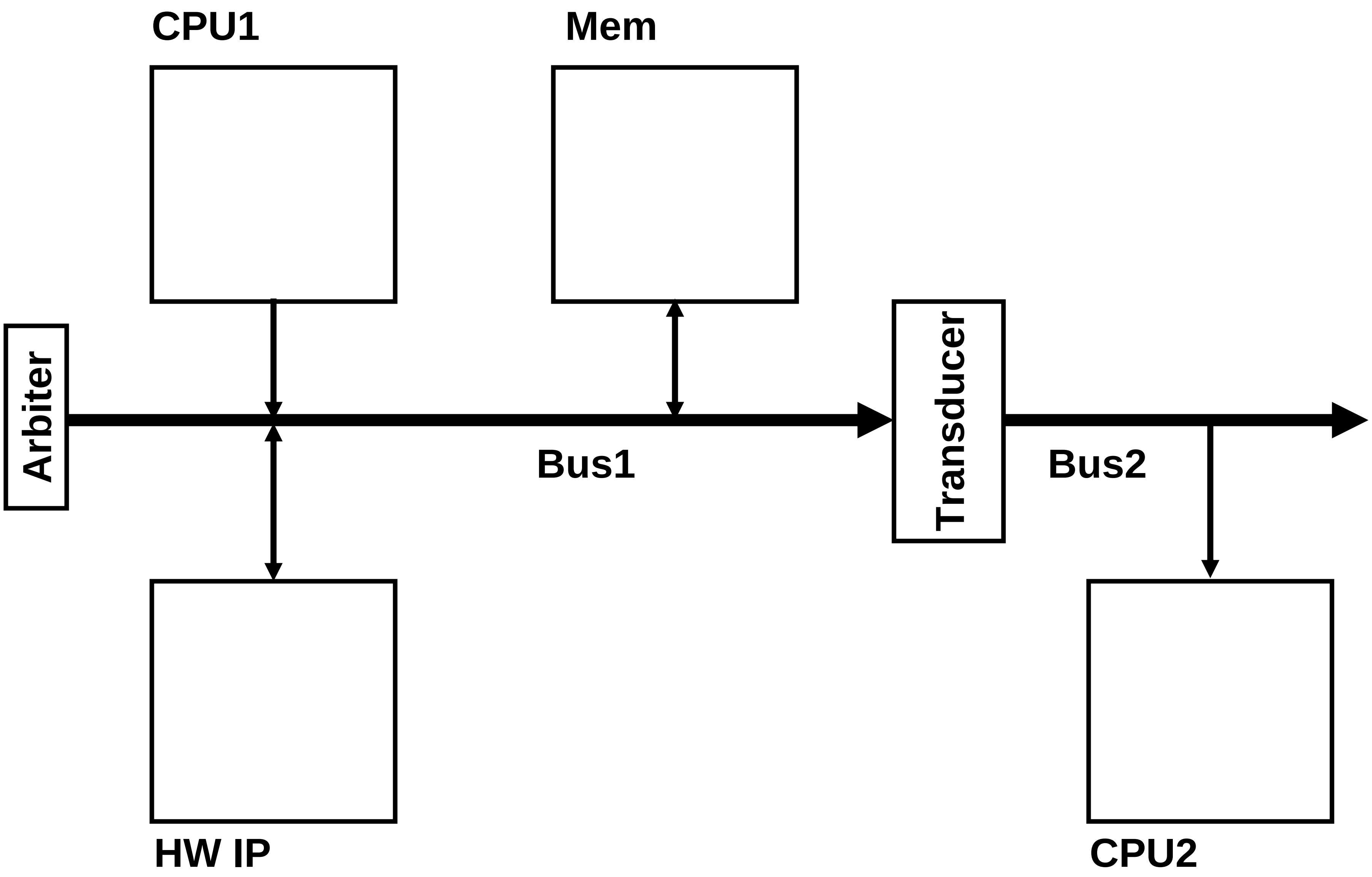
- Motivation for many-core HdS synthesis
- Model based approach to system synthesis
- HdS synthesis for TLM (Front-End)
- HdS synthesis for PCAM (Back-End)
- Embedded System Environment (ESE)
- Experimental results for MP3 decoder example
- Conclusion

Application Model



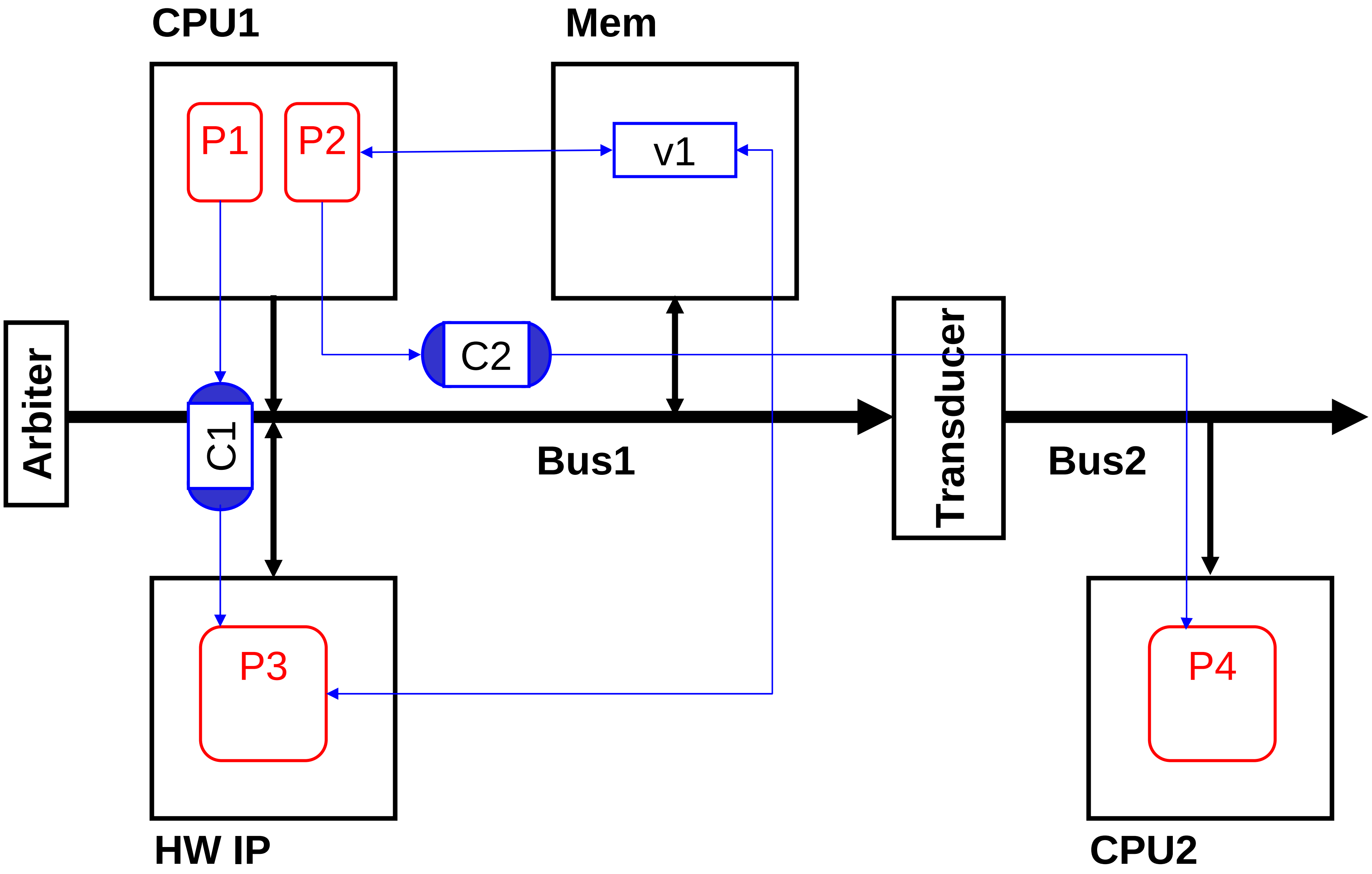
- **Application Spec Model: set of communicating processes**
  - ➢ Executes on a simulation kernel (eg: SystemC) – No HdS
  - ➢ Process functionality defined using C/C++
  - ➢ Blocking channels and non-blocking variables

Platform Architecture



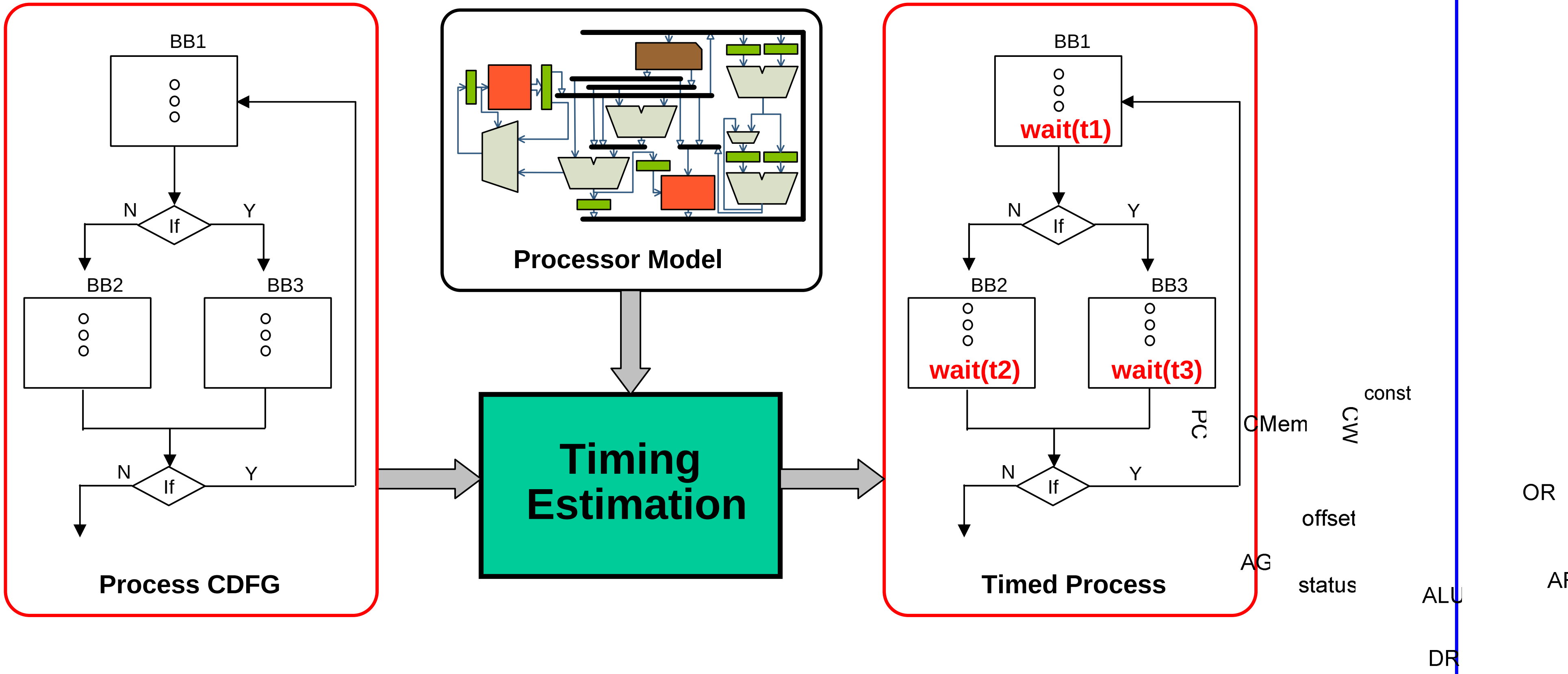
- Netlist of SW processors, HW, Buses and interfaces

Input: System Definition



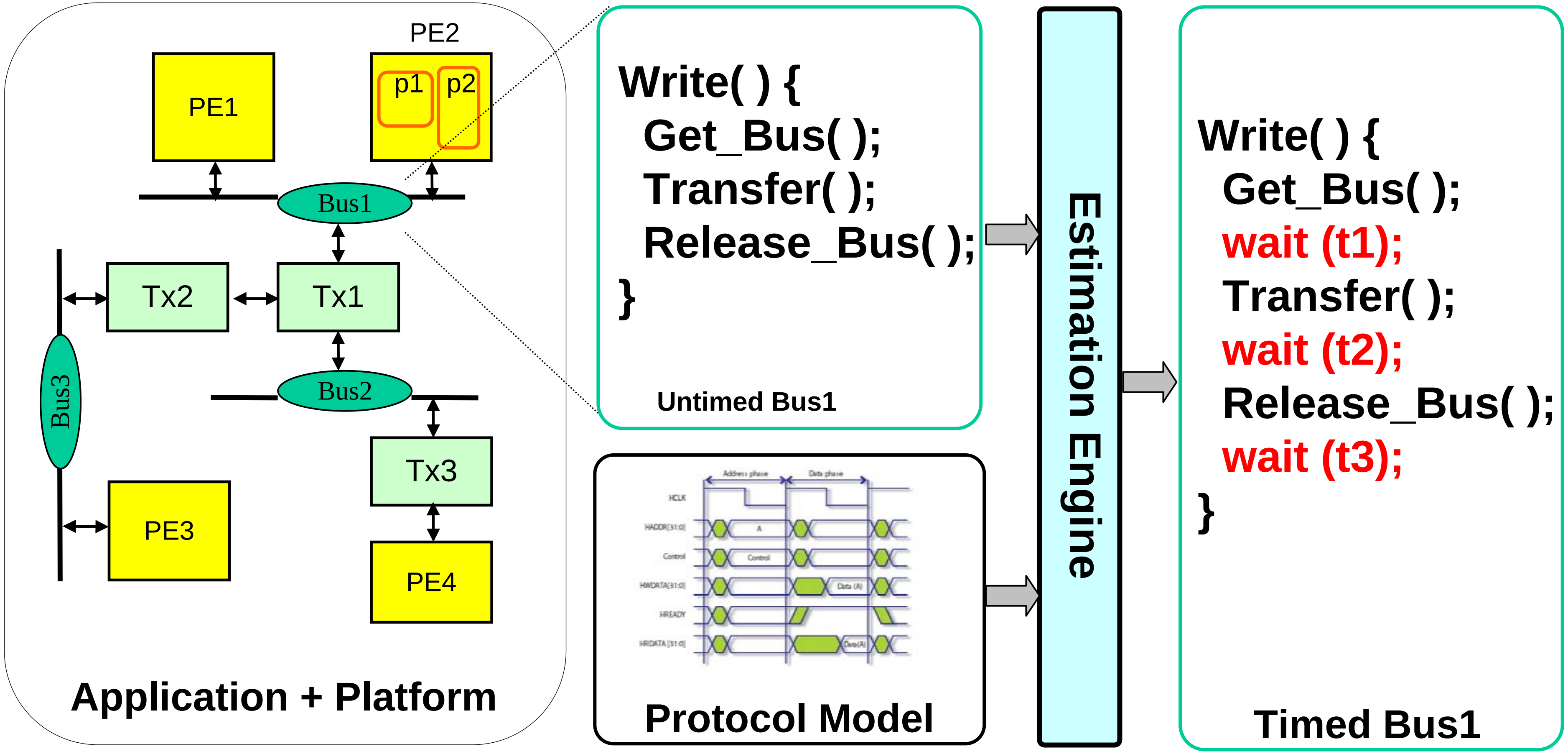
System Definition* = *Platform* + *Application + Mapping

Computation Timing Estimation



- DFG scheduling to compute basic block delay
- RTOS model added for PEs with multiple processes

Communication Timing Estimation



```
Write( ) {
  Get_Bus( );
  Transfer( );
  Release_Bus( );
}
```

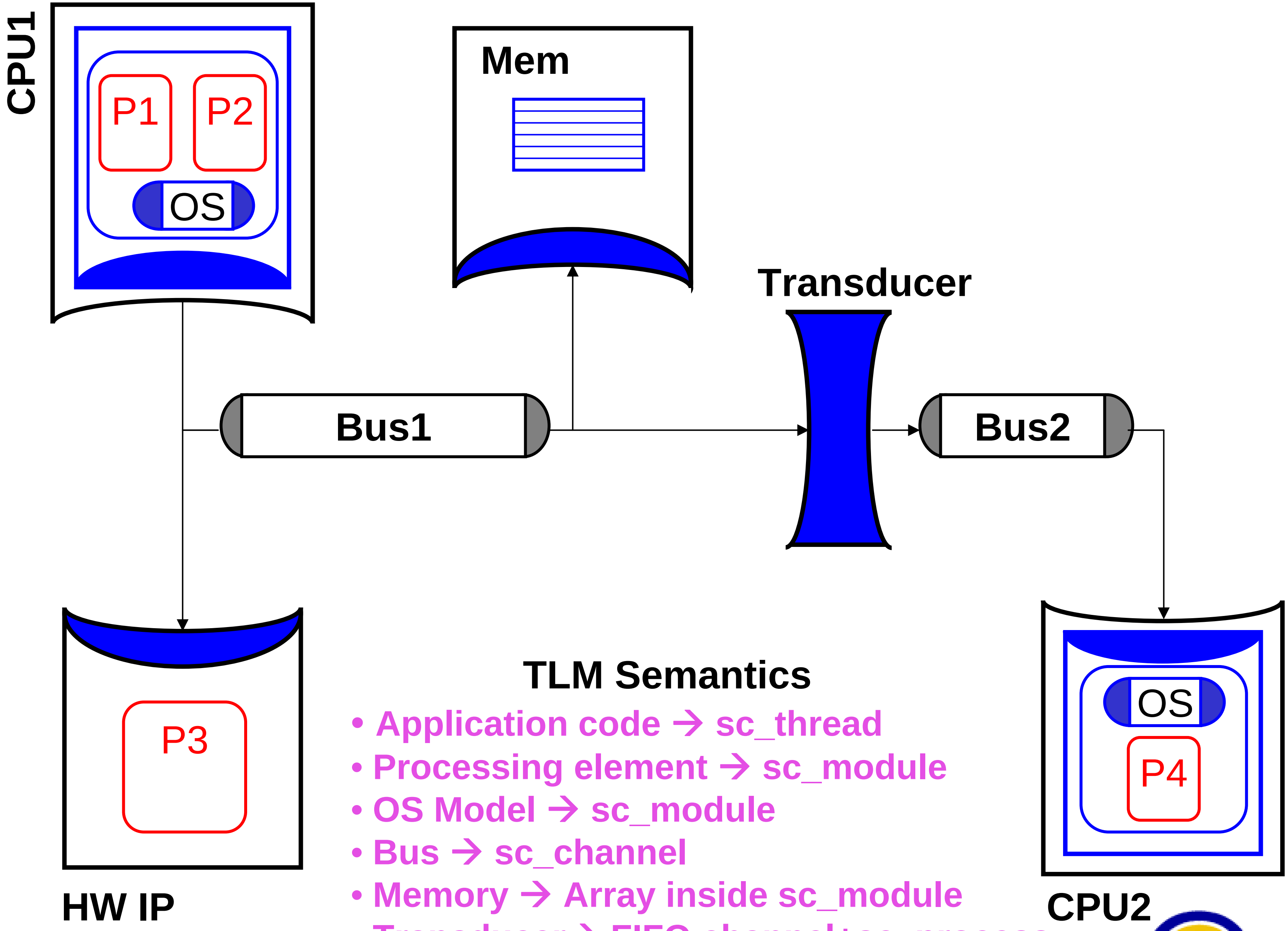
Untimed Bus1

```
Write( ) {
  Get_Bus( );
  wait (t1);
  Transfer( );
  wait (t2);
  Release_Bus( );
  wait (t3);
}
```

Timed Bus1

- Protocol model used to estimate synchronization, arbitration and transfer
- Timing is annotated in bus channel and HdS model

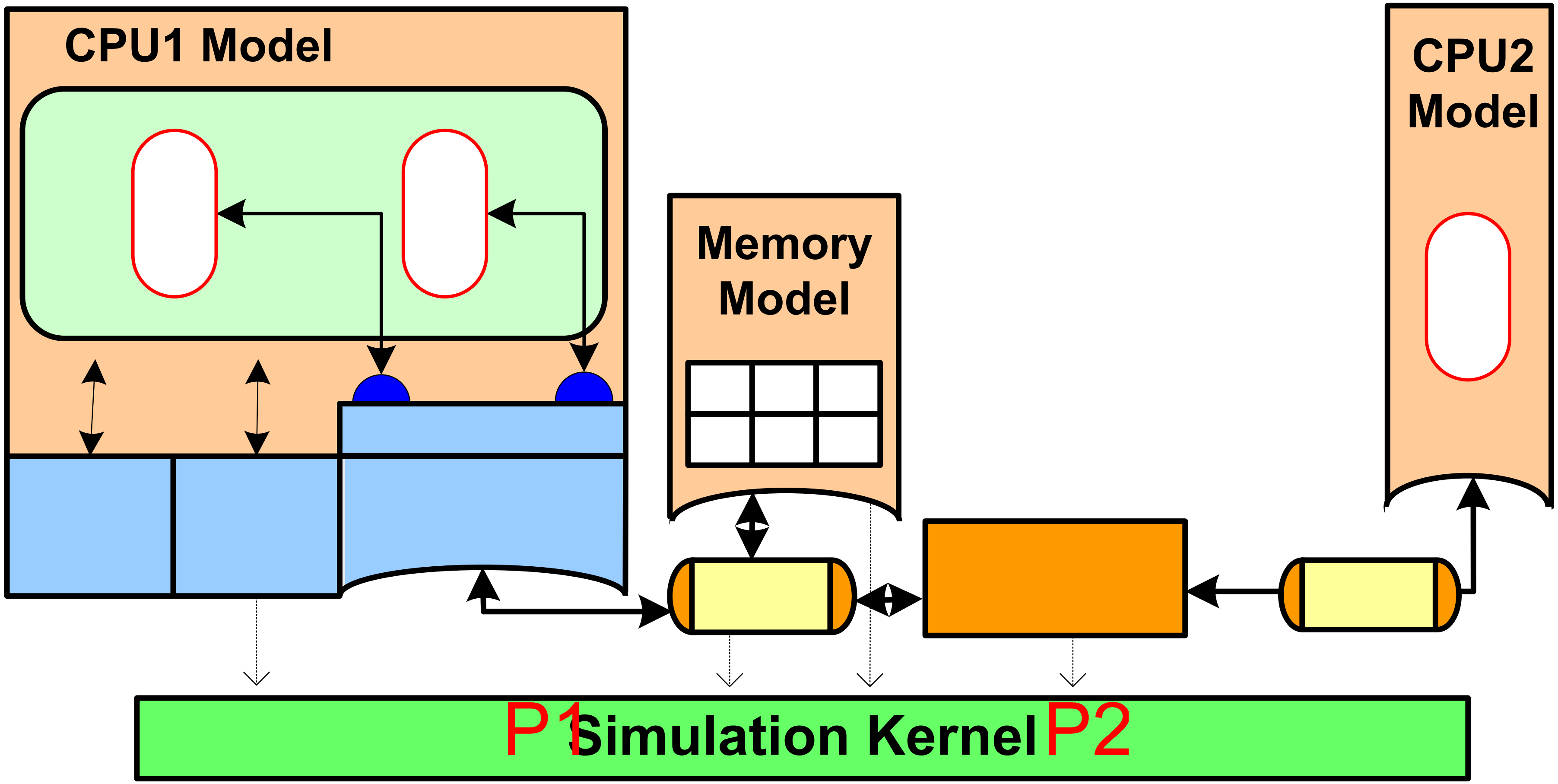
Output: SystemC Timed TLM



TLM Semantics

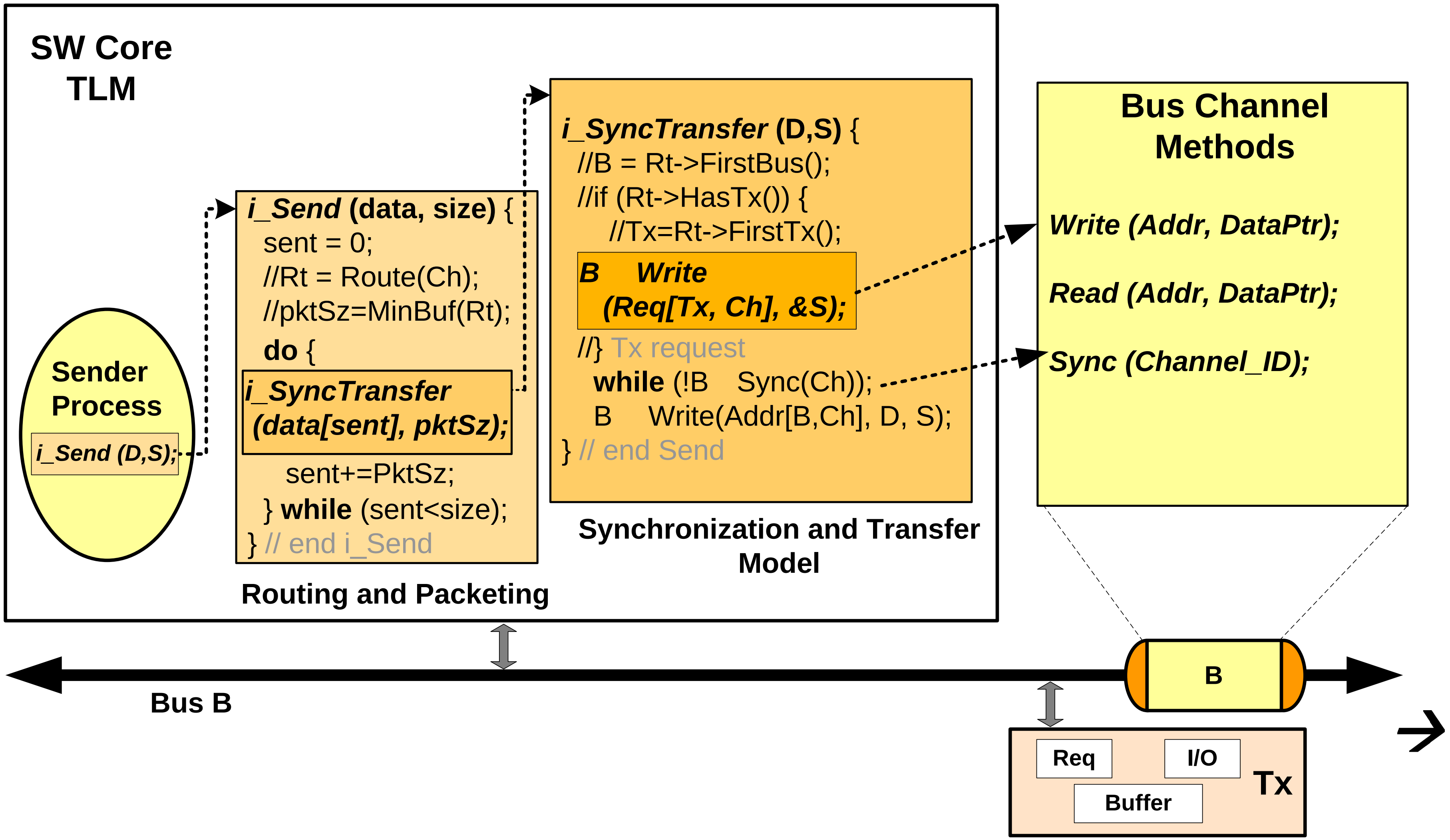
- Application code → sc_thread
- Processing element → sc_module
- OS Model → sc_module
- Bus → sc_channel
- Memory → Array inside sc_module
- Transducer→ FIFO channel+sc_process

Transaction Level HdS Model



- **Routing and packeting layers of HdS generated in TLM**

HdS Layers in TLM



```
i_Send (data, size) {
  sent = 0;
  //Rt = Route(Ch);
  //pktSz=MinBuf(Rt);
  do {
i_SyncTransfer
 (data[sent], pktSz);
    sent+=PktSz;
  } while (sent<size);
} // end i_Send
```

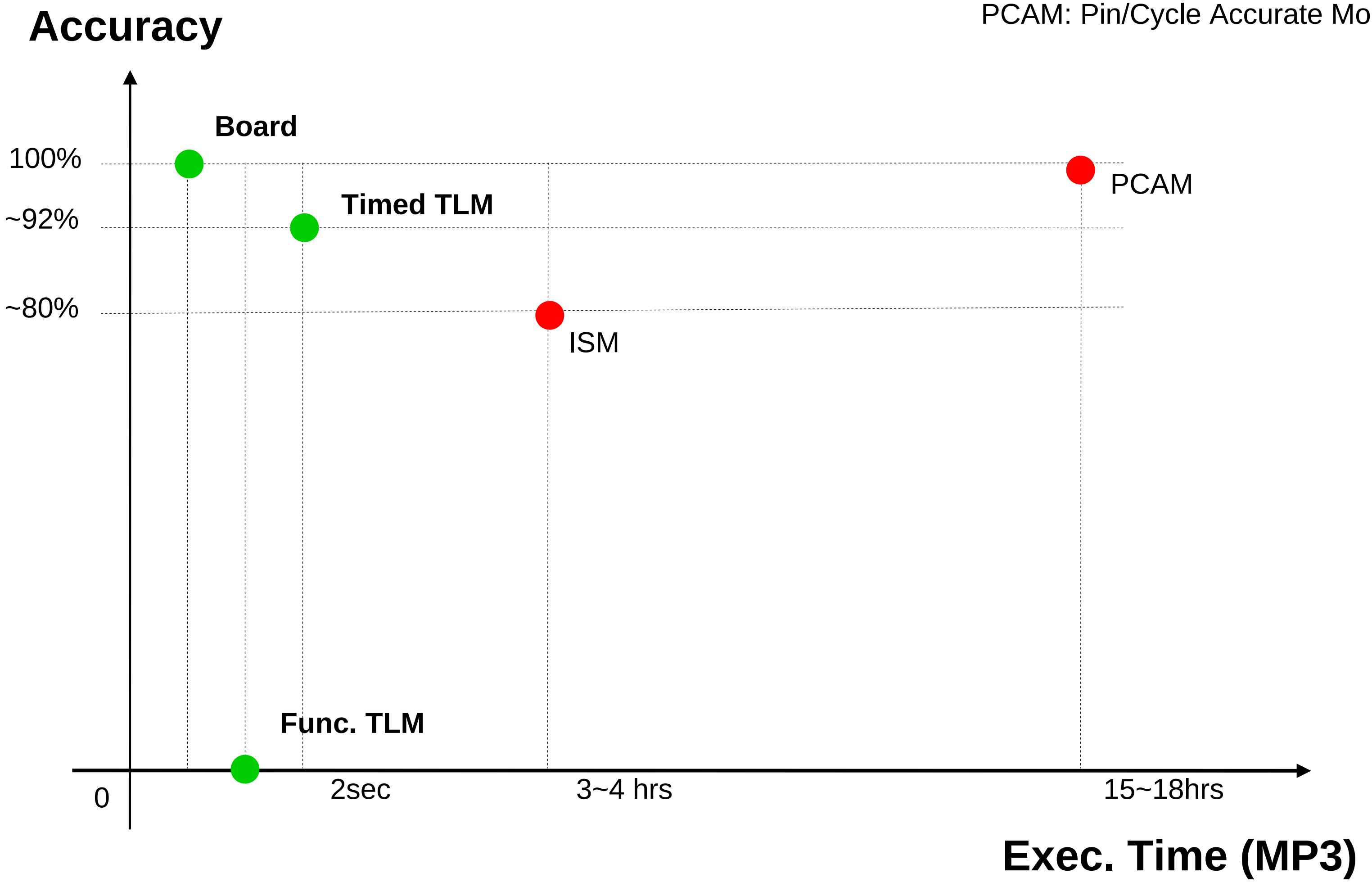
```
i_SyncTransfer (D,S) {
 //B = Rt->FirstBus();
 //if (Rt->HasTx()) {
    //Tx=Rt->FirstTx();
B    Write
  (Req[Tx, Ch], &S);
 //} Tx request
  while (!B   Sync(Ch));
  B    Write(Addr[B,Ch], D, S);
} // end Send
```

```
Write (Addr, DataPtr);

Read (Addr, DataPtr);

Sync (Channel_ID);
```

- Bus channel model provides basic transaction services
- Synchronization and transfer is modeled in SW core

TLM vs. Traditional Models

TLM: Transaction Level Model
ISM: Instruction Set Model
PCAM: Pin/Cycle Accurate Model

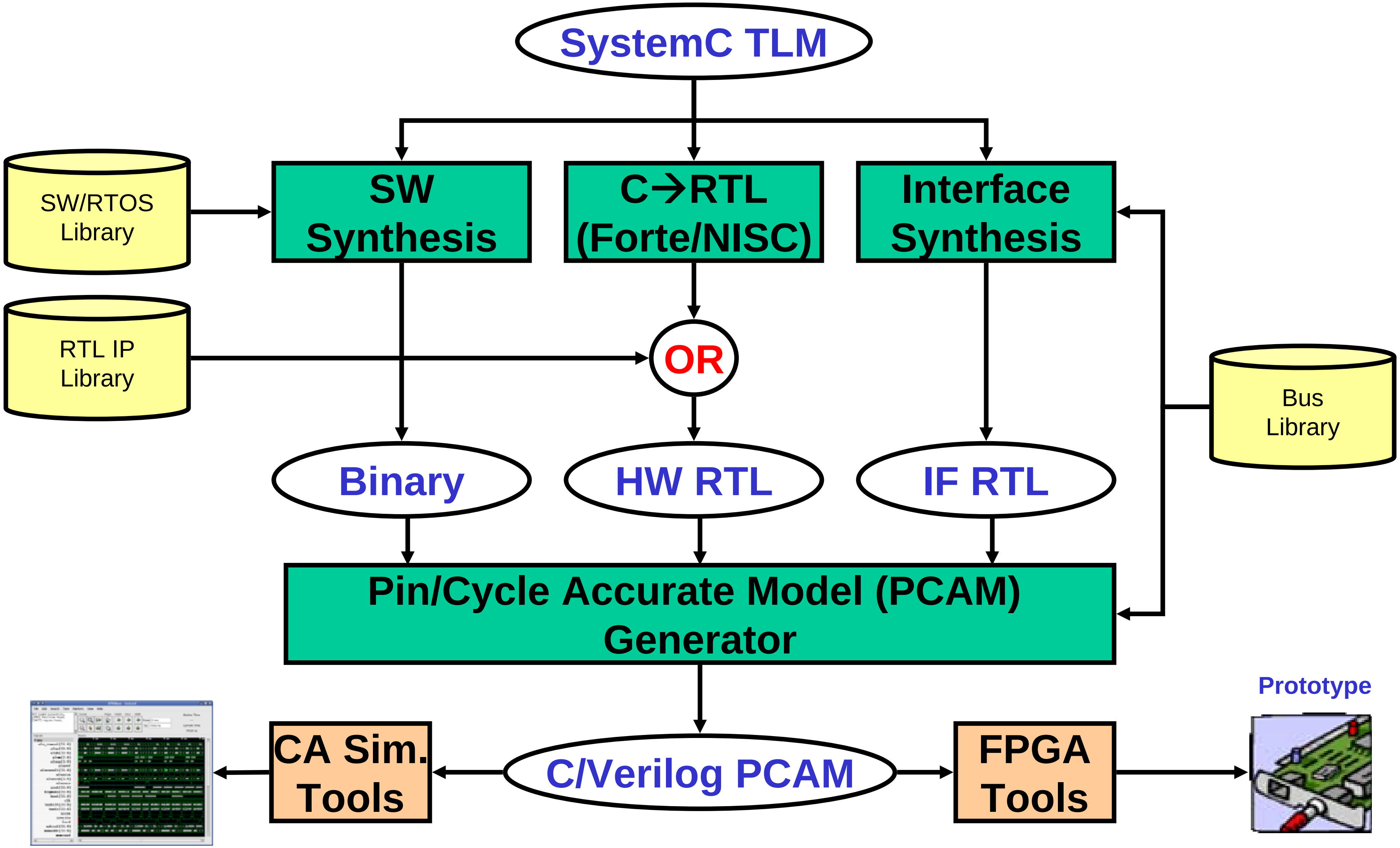


Time and accuracy trade off among different models

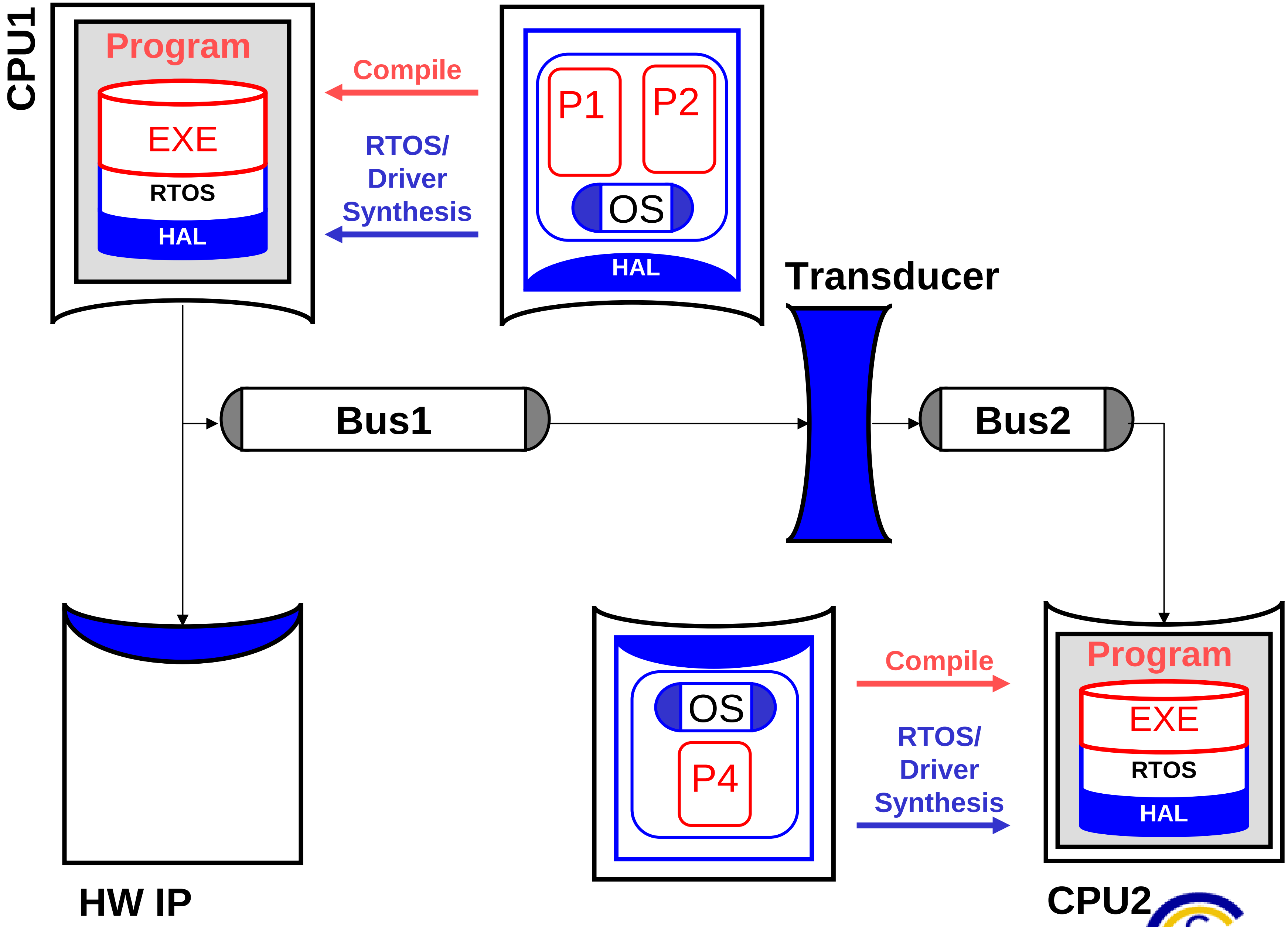
Outline

- Motivation for many-core HdS synthesis
- Model based approach to system synthesis
- HdS synthesis for TLM (Front-End)
- HdS synthesis for PCAM (Back-End)
- Embedded System Environment (ESE)
- Experimental results for MP3 decoder example
- Conclusion

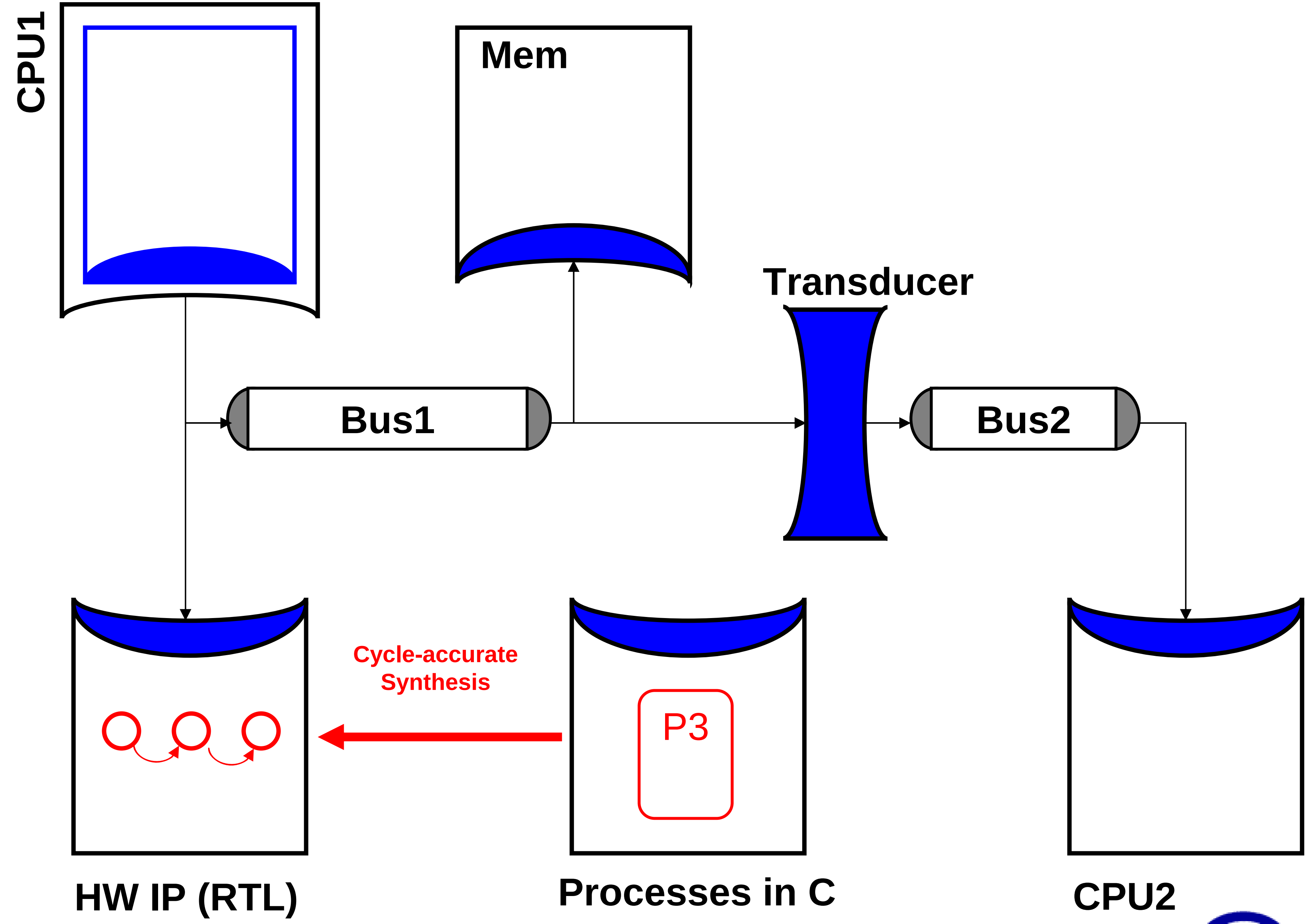
Back End Synthesis Flow



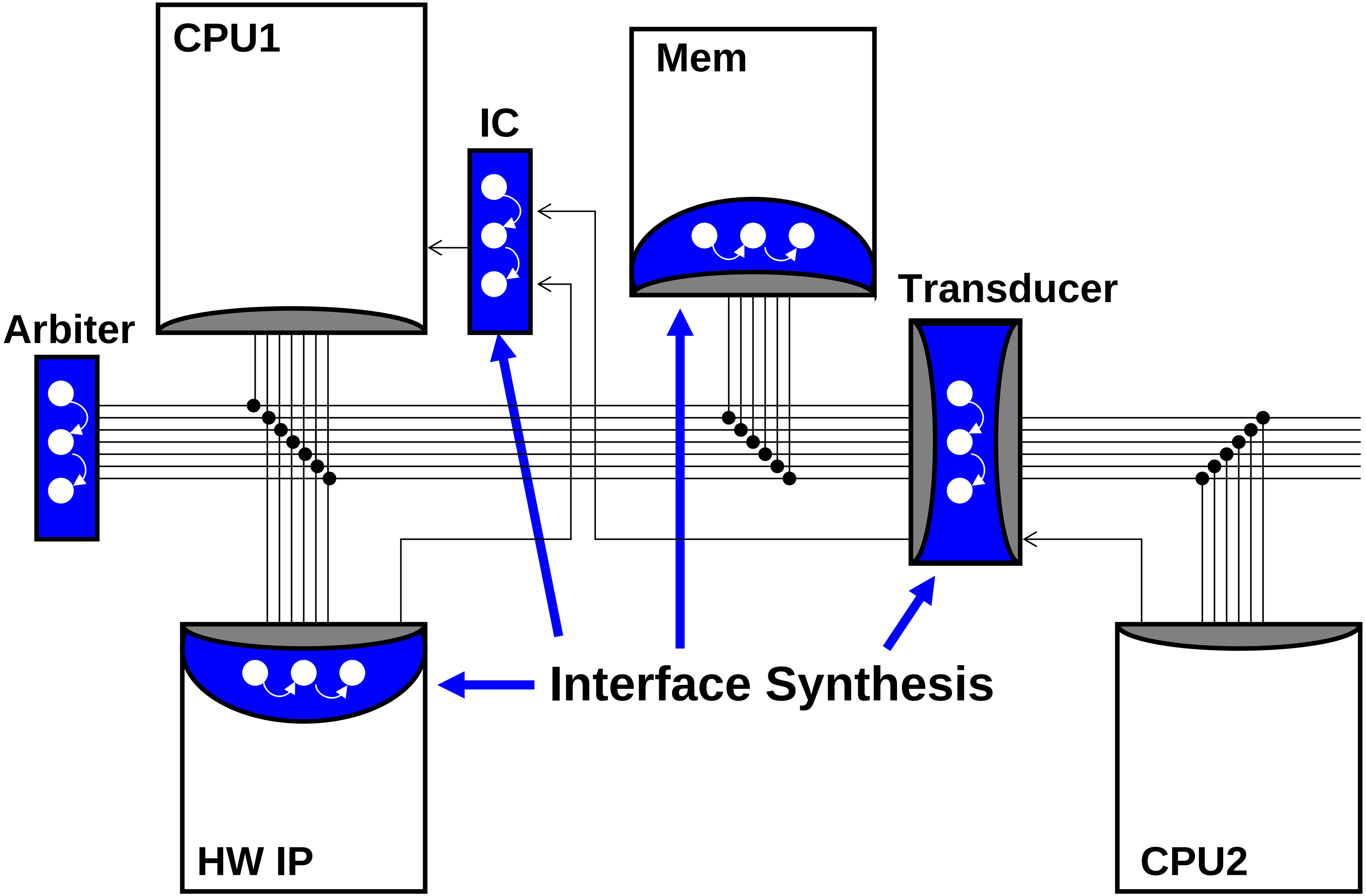
Cycle-Accurate Software Synthesis



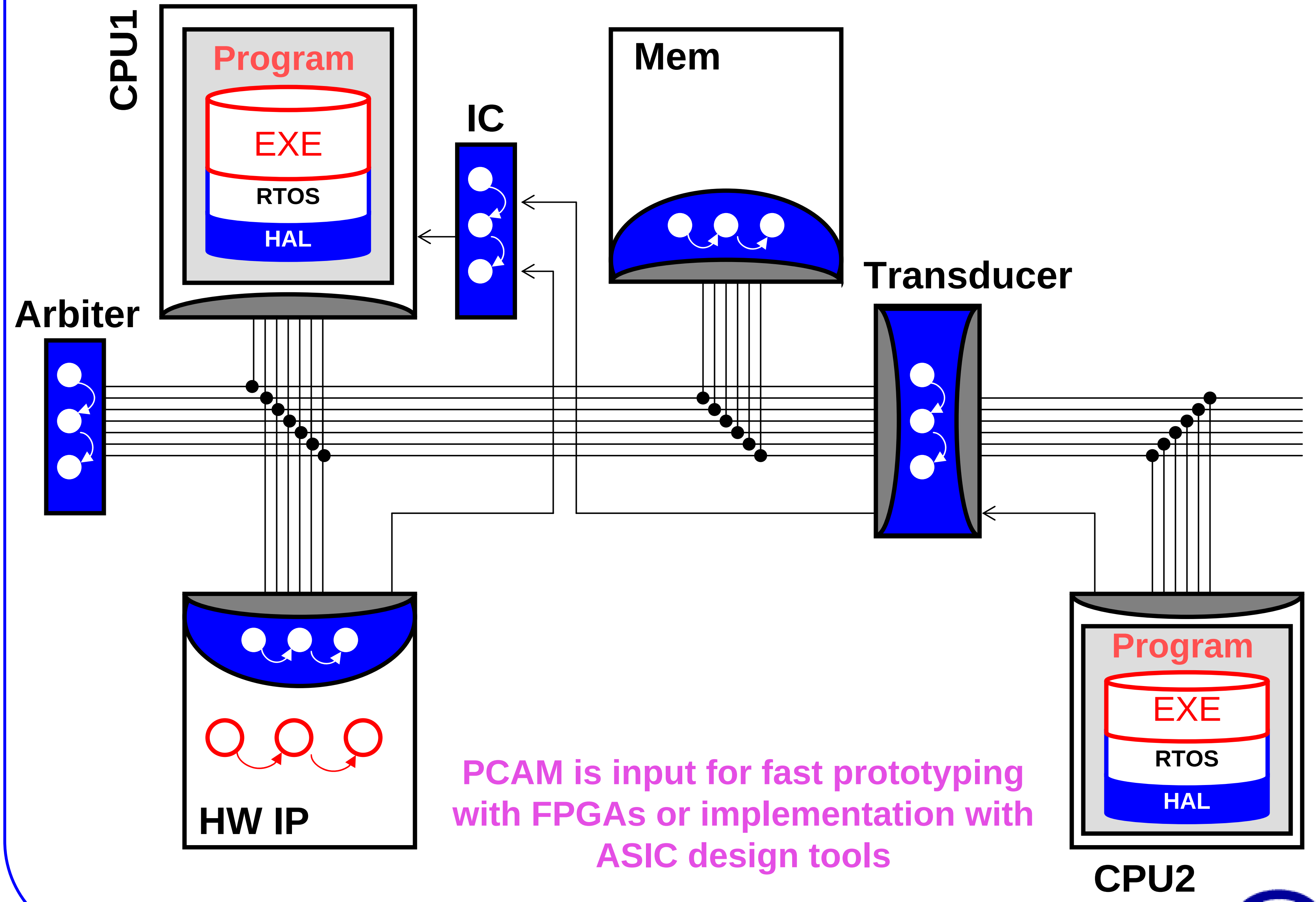
Cycle-Accurate Hardware Synthesis



Cycle-Accurate Interface Synthesis

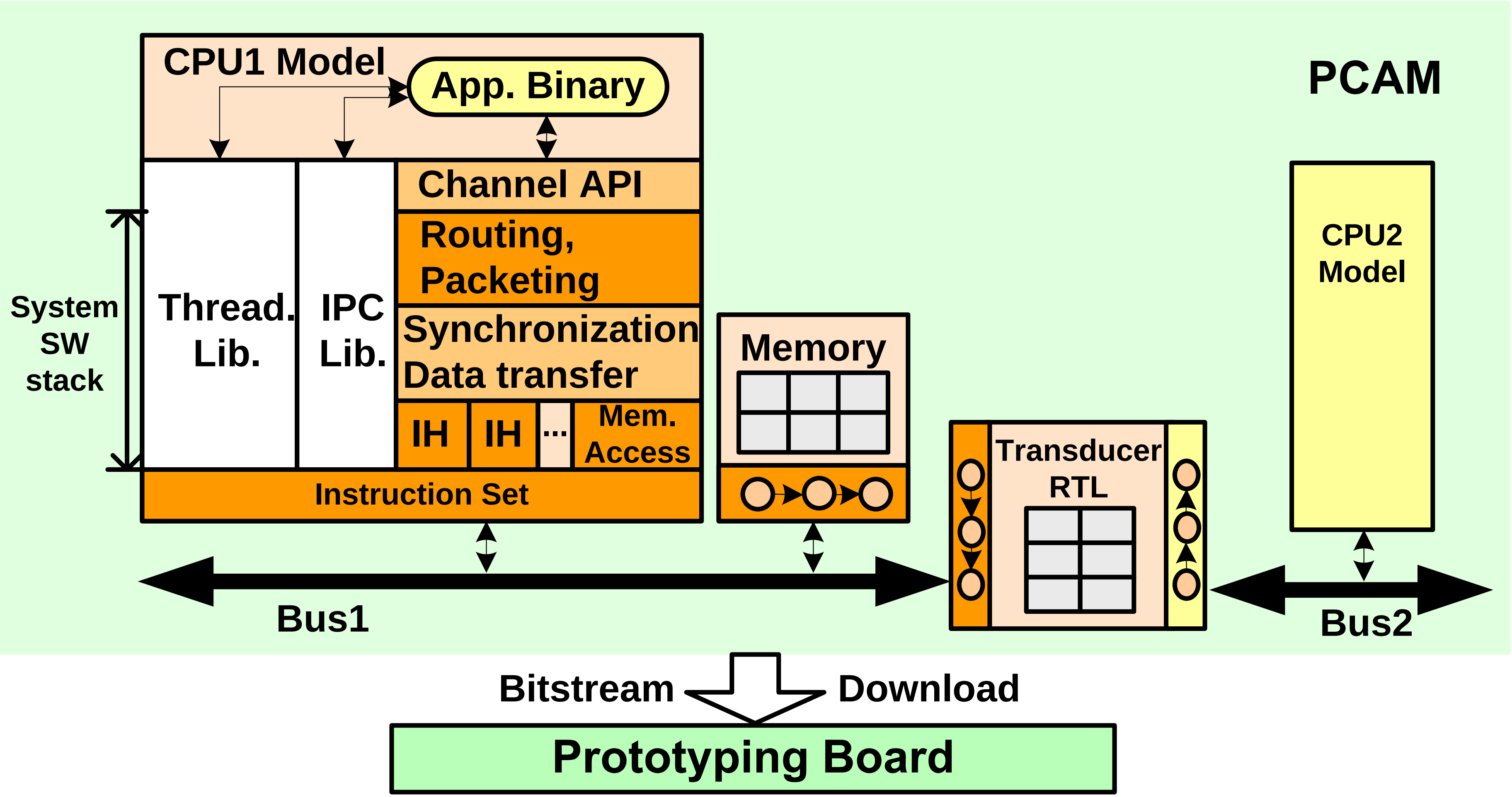


Pin/Cycle-Accurate Model (PCAM)



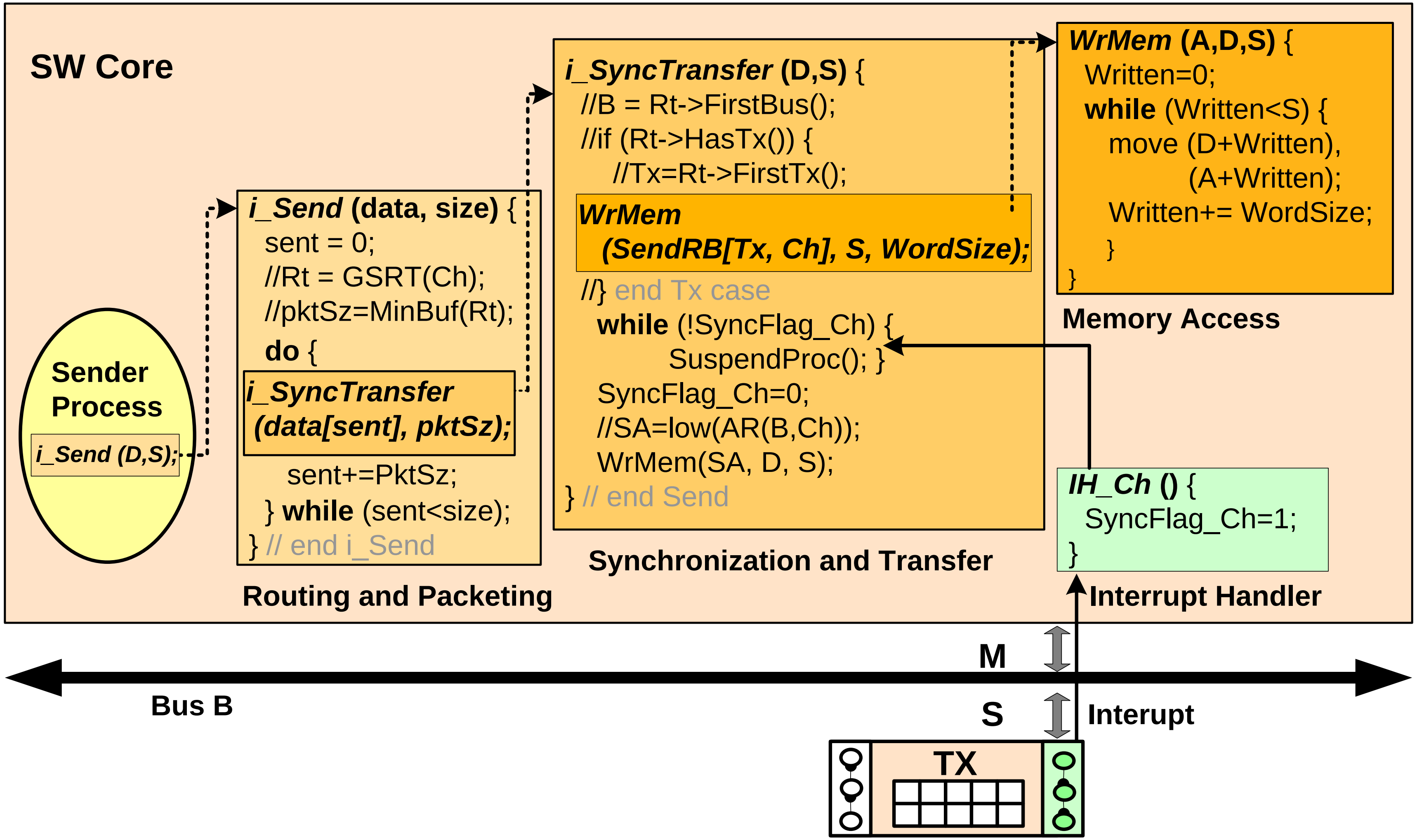
PCAM is input for fast prototyping with FPGAs or implementation with ASIC design tools

Pin and Cycle Accurate Model (PCAM)



- Sync. and transfer layers of HdS generated in PCAM
- Application, System libs and HdS is cross-compiled

HdS Layers in PCAM



SW Core

Sender Process

```
i_Send (D,S);
```

Routing and Packeting

```
i_Send (data, size) {
  sent = 0;
  //Rt = GSRT(Ch);
  //pktSz=MinBuf(Rt);
  do {
i_SyncTransfer
 (data[sent], pktSz);
     sent+=PktSz;
  } while (sent<size);
} // end i_Send
```

Synchronization and Transfer

```
i_SyncTransfer (D,S) {
  //B = Rt->FirstBus();
  //if (Rt->HasTx()) {
     //Tx=Rt->FirstTx();
WrMem
  (SendRB[Tx, Ch], S, WordSize);
  //} end Tx case
   while (!SyncFlag_Ch) {
          SuspendProc(); }
   SyncFlag_Ch=0;
   //SA=low(AR(B,Ch));
   WrMem(SA, D, S);
} // end Send
```

Memory Access

```
WrMem (A,D,S) {
  Written=0;
  while (Written<S) {
    move (D+Written),
            (A+Written);
    Written+= WordSize;
    }
}
```

Interrupt Handler

```
IH_Ch () {
  SyncFlag_Ch=1;
}
```

- **Synchronization and transfer models are replaced with core, platform and application-specific HdS**

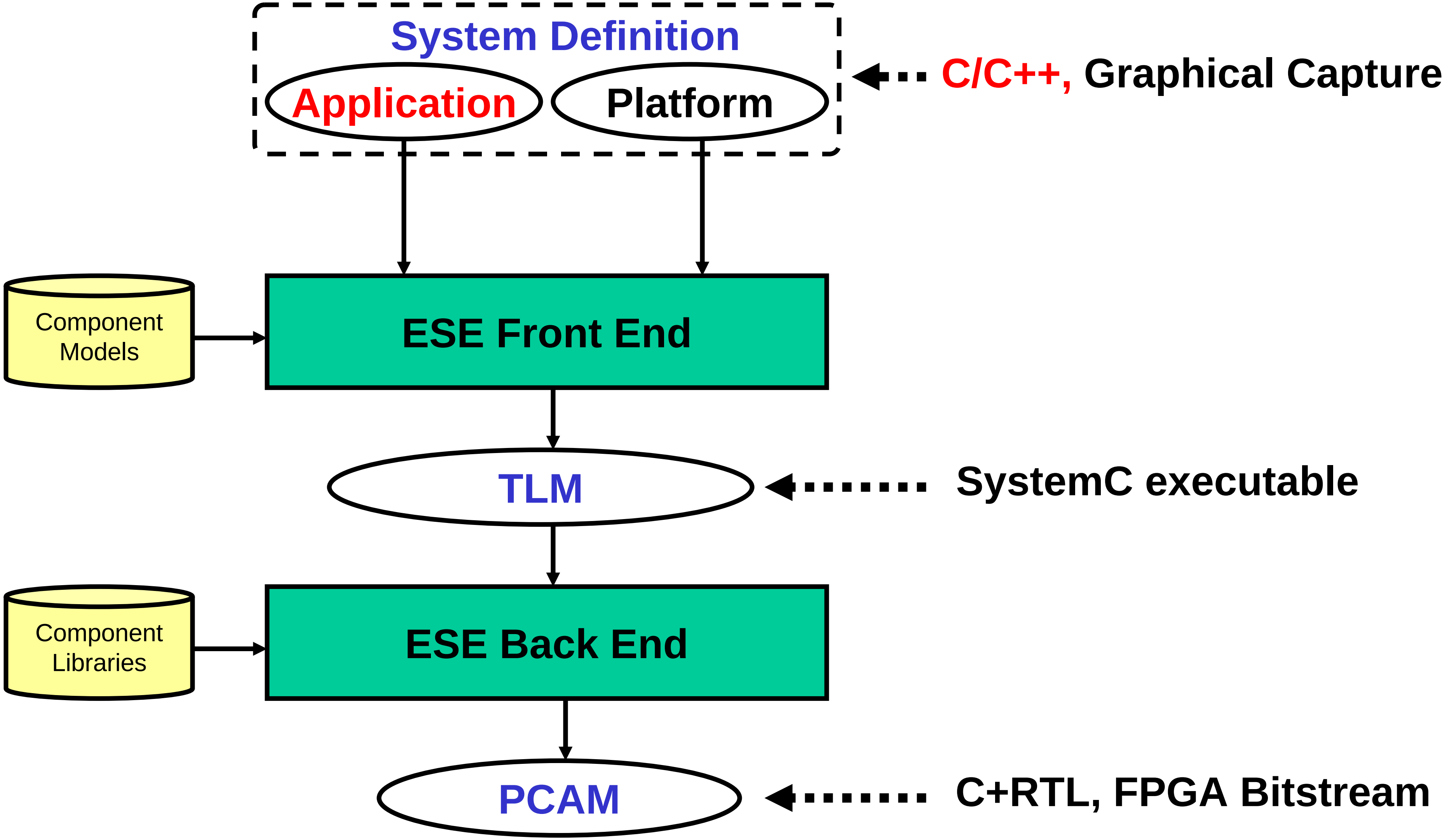
Summary of HdS Layers

- **Routing**
  - ➢ Statically decided for each channel
  - ➢ Transducer request code is generated if necessary
- **Packeting**
  - ➢ packet size is minimum buffer size in route
  - ➢ Message is divided into synchronized packet transactions
- **Synchronization**
  - ➢ Interrupt handlers and flags are generated per channel
  - ➢ Polling addresses and frequencies are selected per channel
- **Data transfer**
  - ➢ Addressing is defined per channel-per bus-segment
  - ➢ Core-specific read-write methods are created

Outline

- Motivation for many-core HdS synthesis
- Model based approach to system synthesis
- HdS synthesis for TLM (Front-End)
- HdS synthesis for PCAM (Back-End)
- Embedded System Environment (ESE)
- Experimental results for MP3 decoder example
- Conclusion

Embedded System Environment



C/C++, Graphical Capture

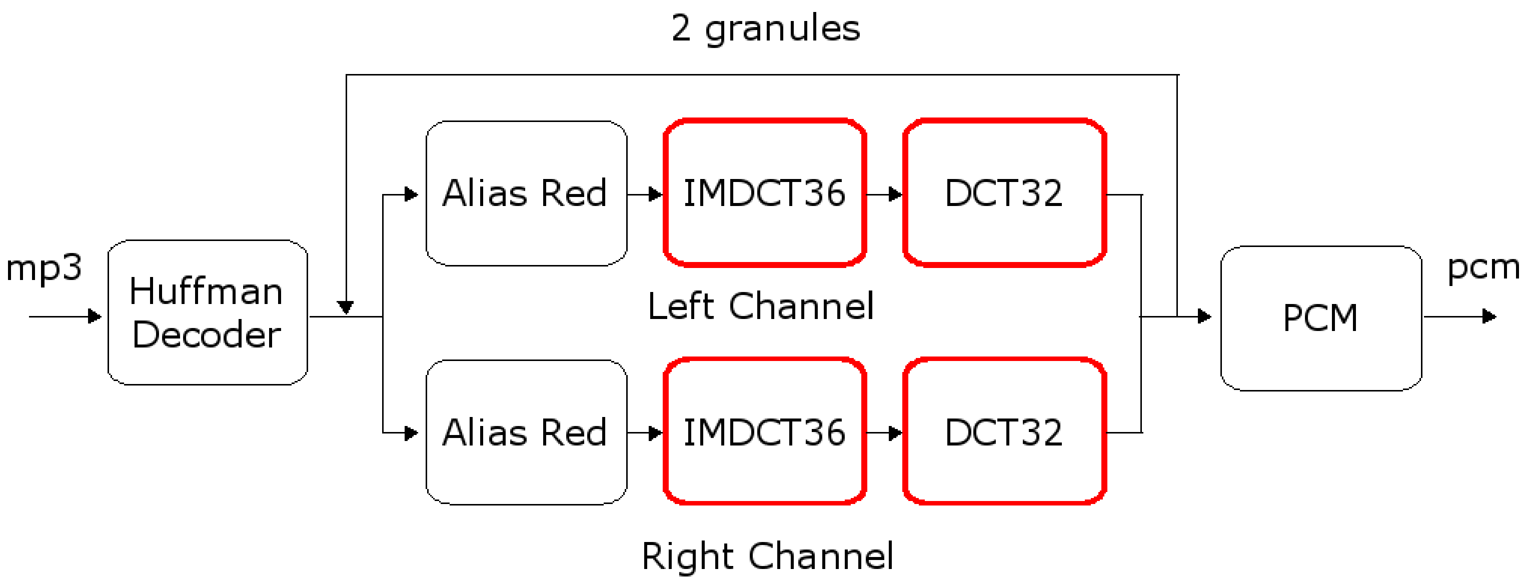
SystemC executable

C+RTL, FPGA Bitstream

Available for download at http://www.cecs.uci.edu/~ese

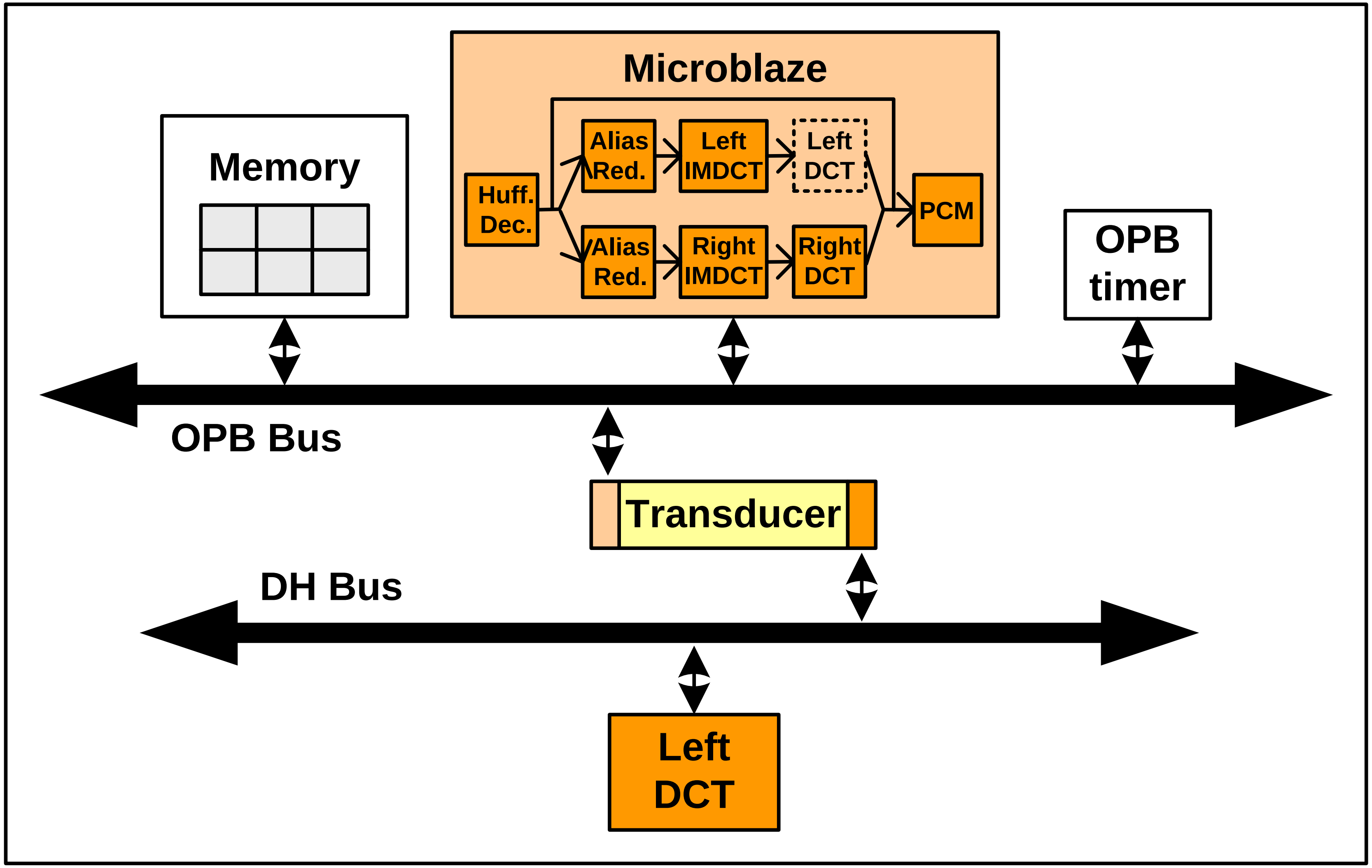
MP3 Decoder Application

- **Functional block diagram (major blocks only)**

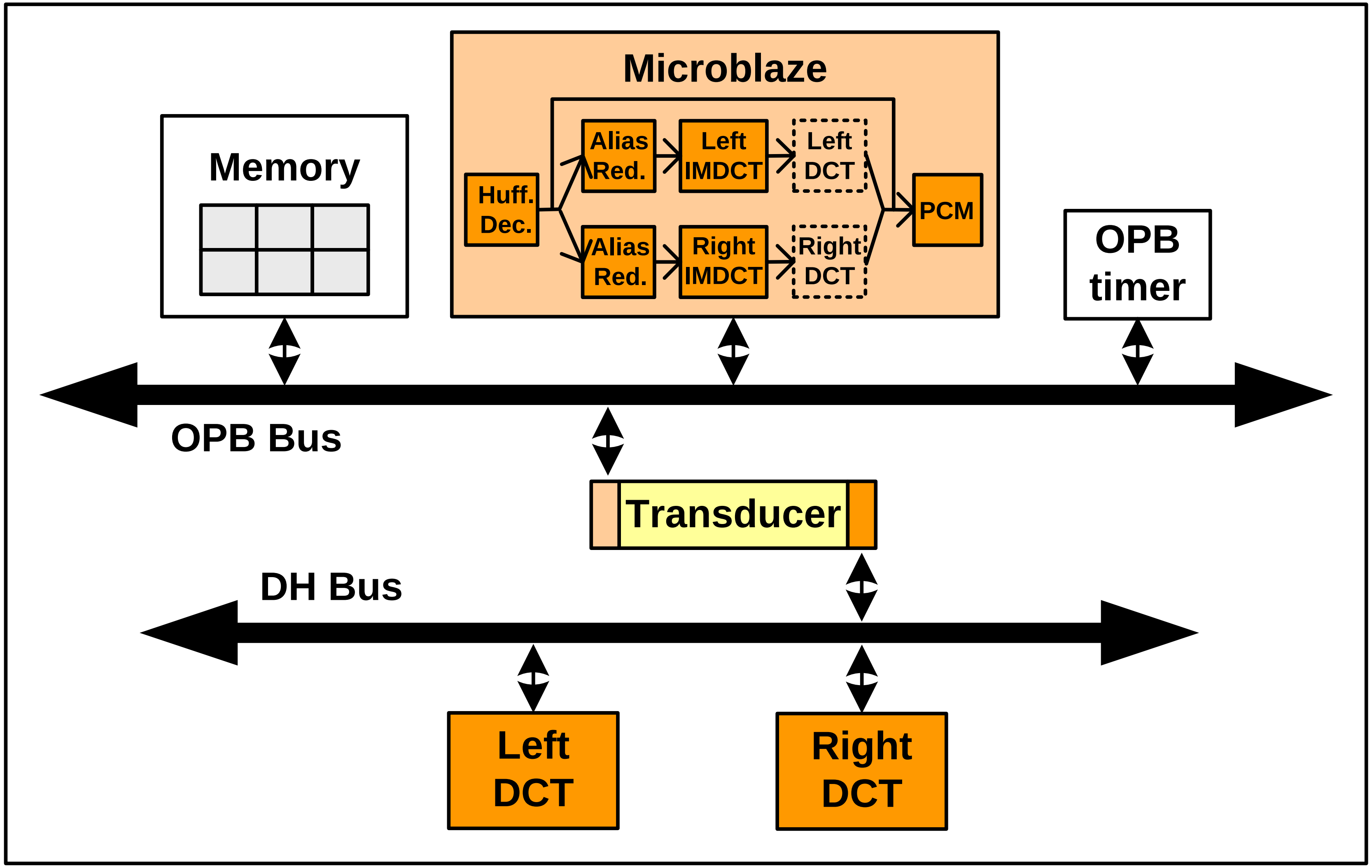


- **Application features**
  - **12K lines of C code**
  - **IMDCT and DCT are compute intensive**
    - **Candidates for HW implementation**
  - **Left channel and right channel are data independent**
    - **Concurrent execution possible**

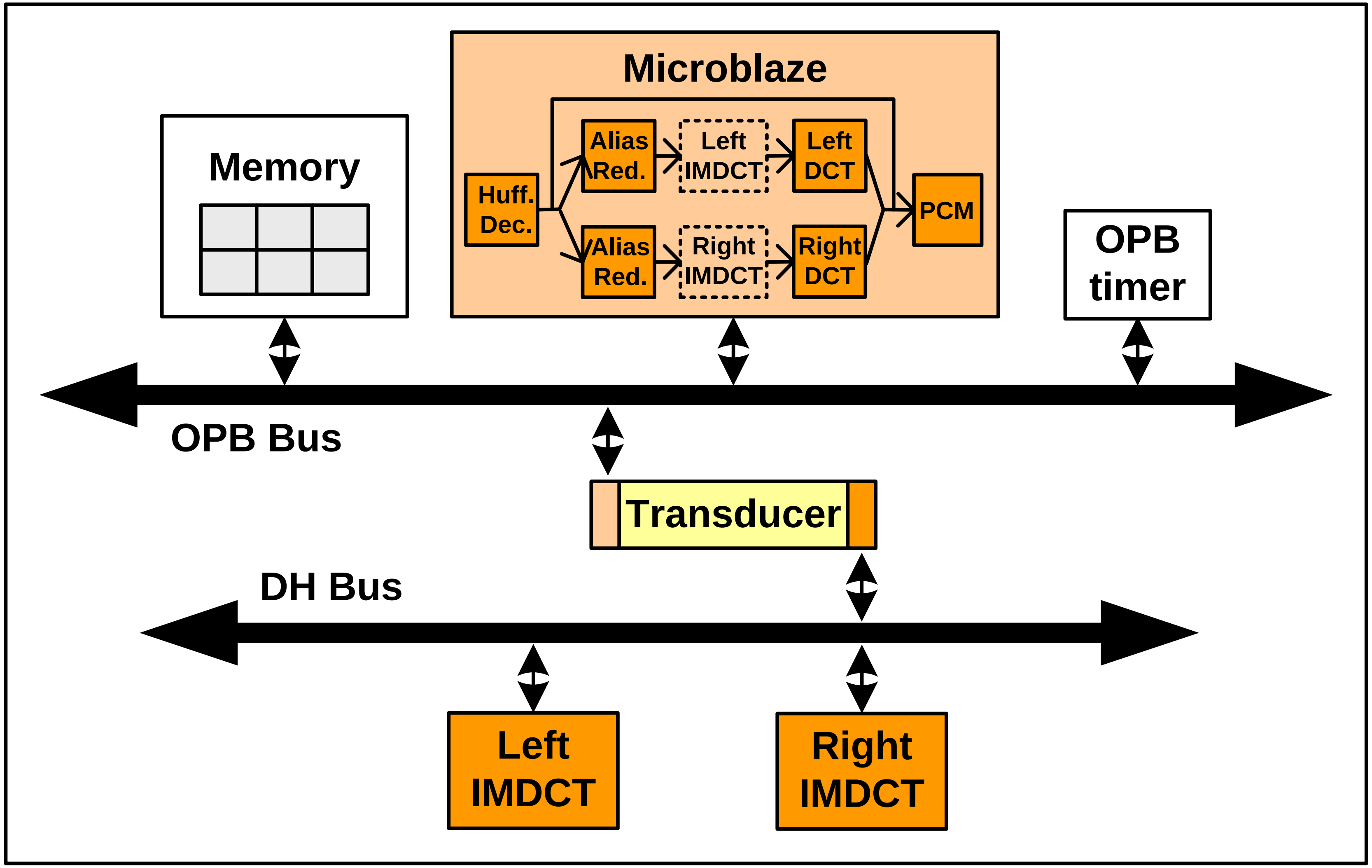
Platform 1: SW + 1DCT



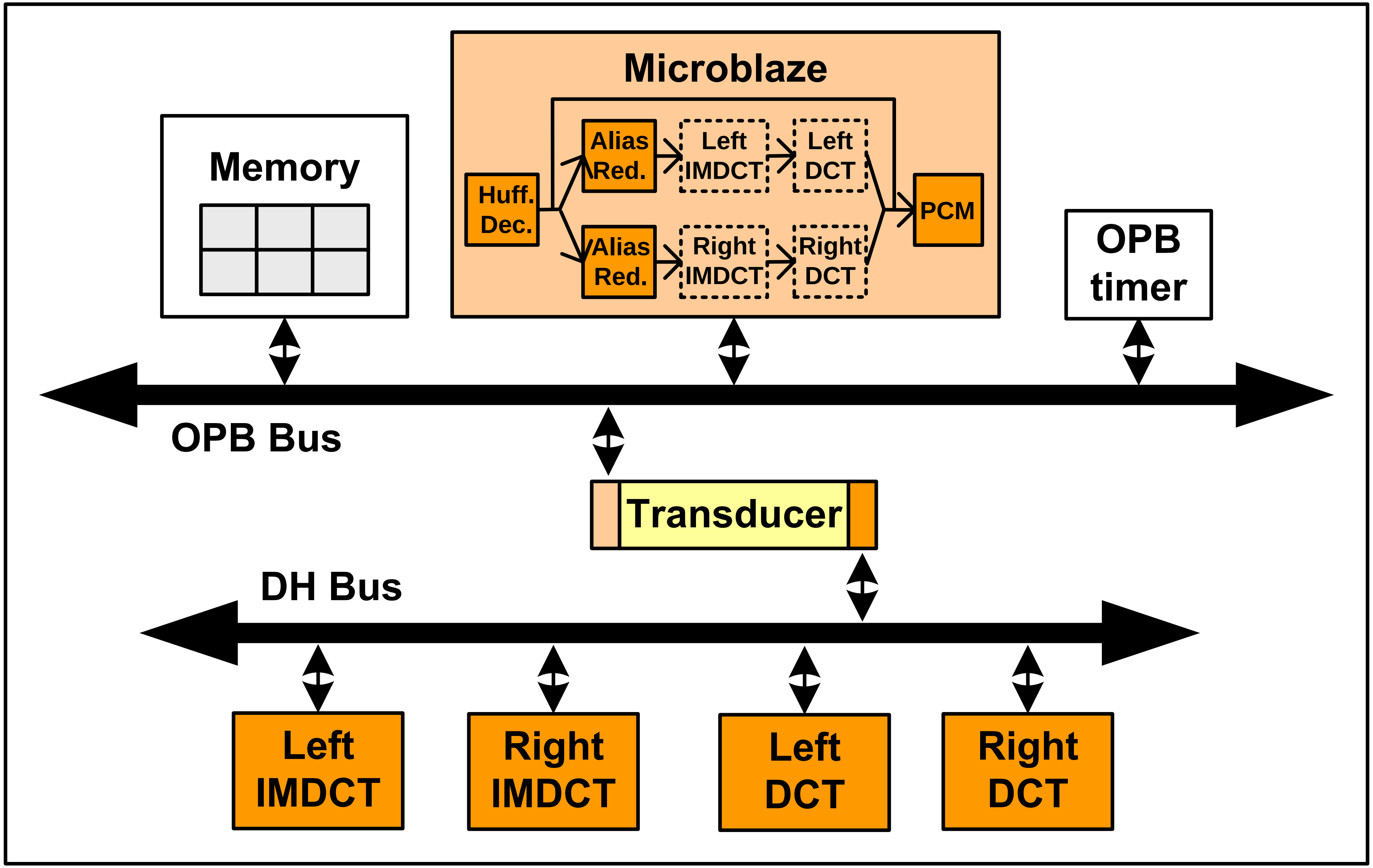
Platform 2: SW + 2DCT



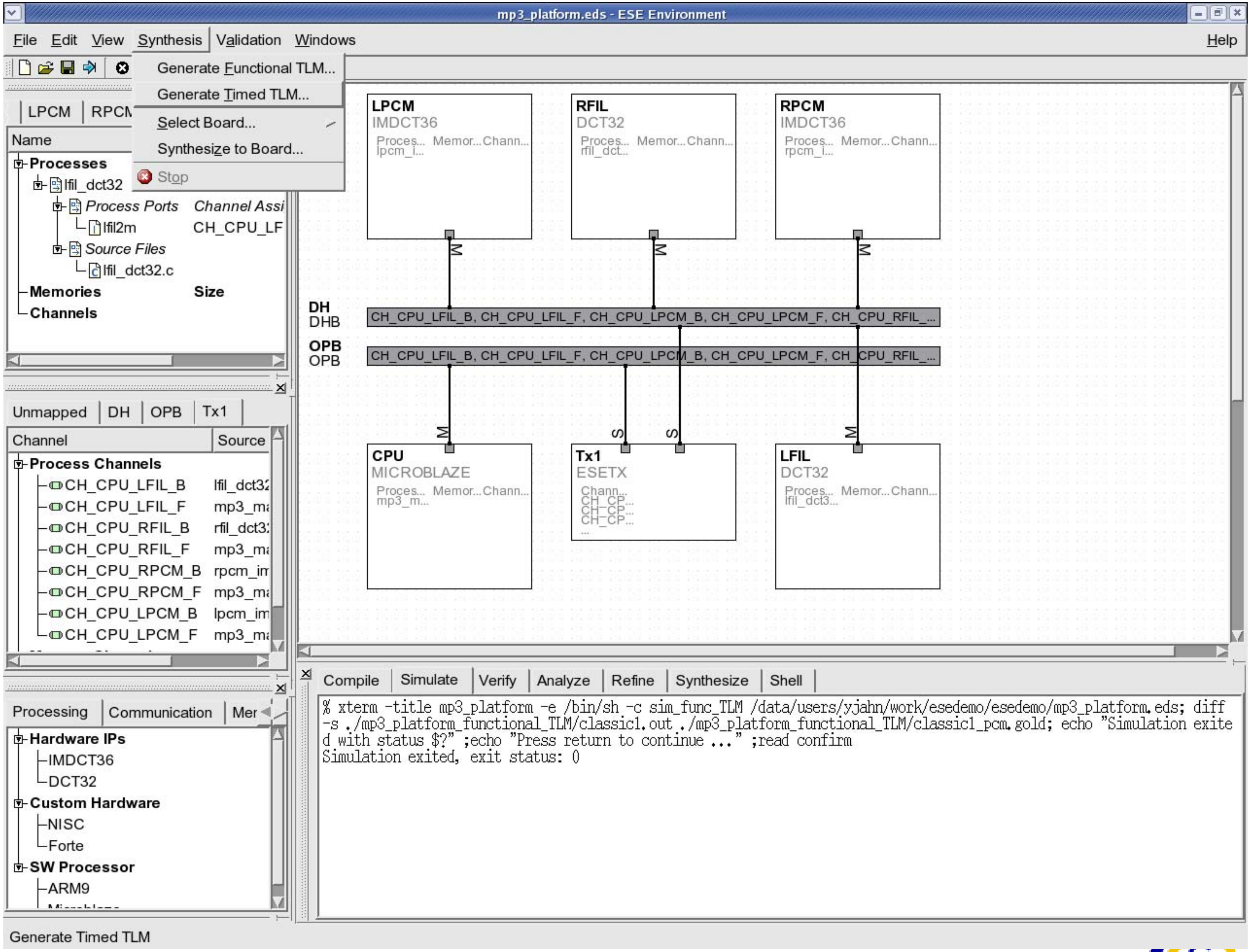
Platform 3: SW + 2IMDCT



Platform 4: SW + 2DCT + 2IMDCT

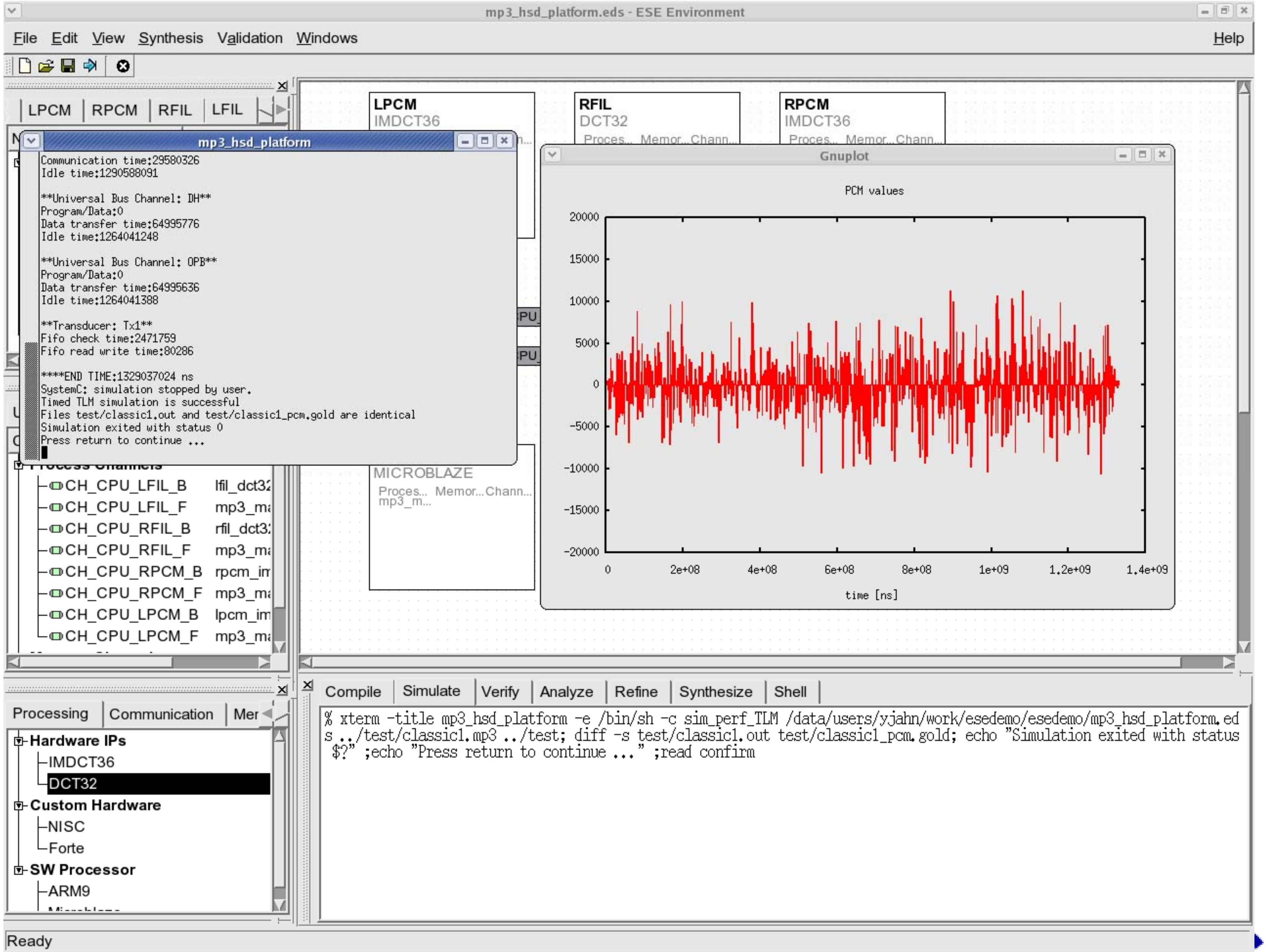


ESE System Specification


mp3_platform.eds - ESE Environment
File Edit View Synthesis Validation Windows Help
Generate Functional TLM...
Generate Timed TLM...
Select Board...
Synthesize to Board...
Stop
LPCM RPCM
Name
Processes
lfil_dct32
Process Ports Channel Assi
lfil2m CH_CPU_LF
Source Files
lfil_dct32.c
Memories Size
Channels
Unmapped DH OPB Tx1
Channel Source
Process Channels
CH_CPU_LFIL_B lfil_dct32
CH_CPU_LFIL_F mp3_ma
CH_CPU_RFIL_B rfil_dct3
CH_CPU_RFIL_F mp3_ma
CH_CPU_RPCM_B rpcm_im
CH_CPU_RPCM_F mp3_ma
CH_CPU_LPCM_B lpcm_im
CH_CPU_LPCM_F mp3_ma
Processing Communication Mem
Hardware IPs
IMDCT36
DCT32
Custom Hardware
NISC
Forte
SW Processor
ARM9
Microblaze
LPCM IMDCT36 Proces... Memor... Chann... lpcm_i...
RFIL DCT32 Proces... Memor... Chann... rfil_dct...
RPCM IMDCT36 Proces... Memor... Chann... rpcm_i...
M
DH DHB
CH_CPU_LFIL_B, CH_CPU_LFIL_F, CH_CPU_LPCM_B, CH_CPU_LPCM_F, CH_CPU_RFIL_...
OPB OPB
CH_CPU_LFIL_B, CH_CPU_LFIL_F, CH_CPU_LPCM_B, CH_CPU_LPCM_F, CH_CPU_RFIL_...
S
CPU MICROBLAZE Proces... Memor... Chann... mp3_m...
Tx1 ESETX Chann... CH_CP... CH_CP... CH_CP...
LFIL DCT32 Proces... Memor... Chann... lfil_dct3...
Compile Simulate Verify Analyze Refine Synthesize Shell
Generate Timed TLM

```
% xterm -title mp3_platform -e /bin/sh -c sim_func_TLM /data/users/yjahn/work/esedemo/esedemo/mp3_platform.eds; diff
-s ./mp3_platform_functional_TLM/classic1.out ./mp3_platform_functional_TLM/classic1_pcm.gold; echo "Simulation exite
d with status $?" ;echo "Press return to continue ..." ;read confirm
Simulation exited, exit status: 0
```

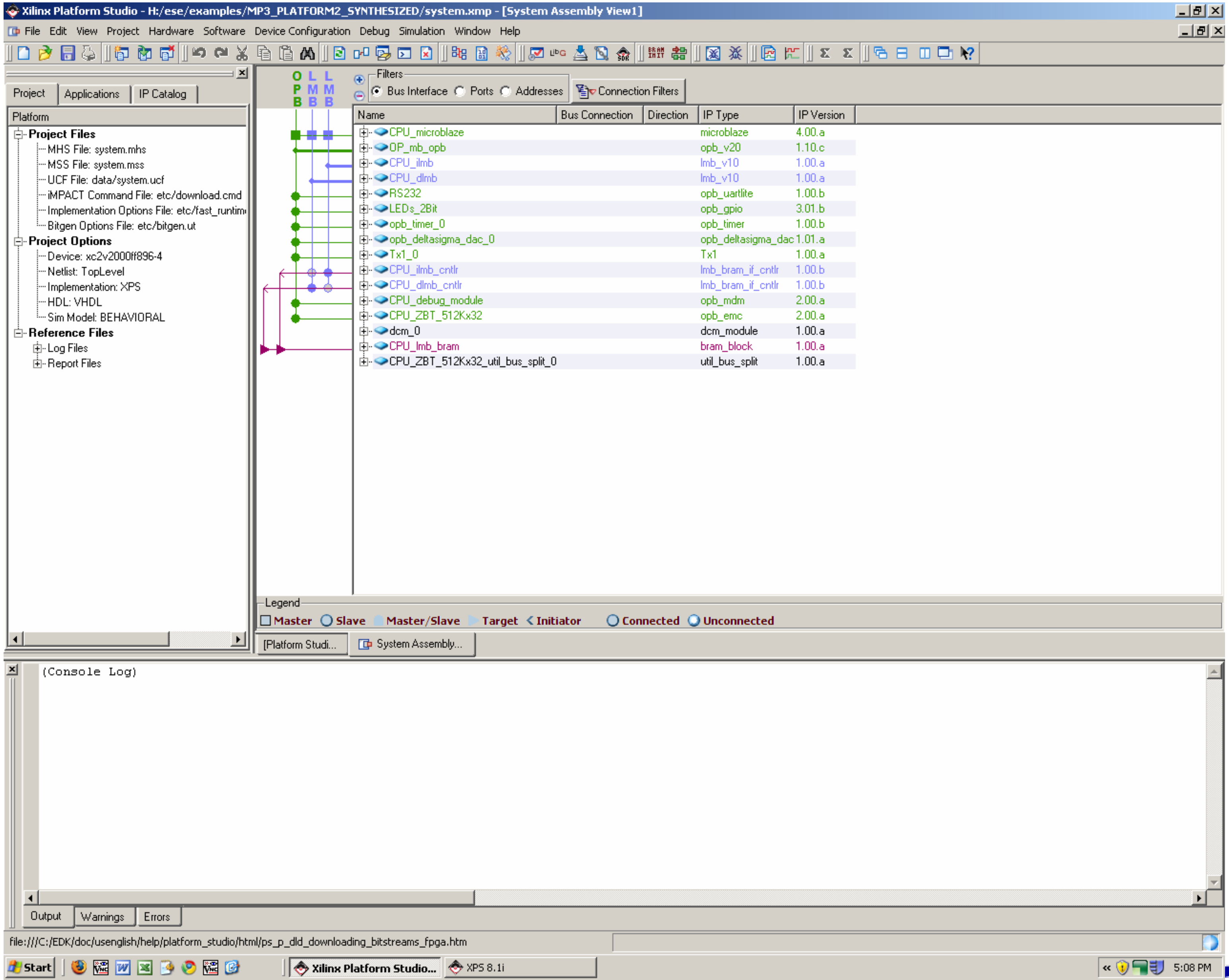
ESE TLM Simulation



```
Communication time:29580326
Idle time:1290588091

**Universal Bus Channel: DH**
Program/Data:0
Data transfer time:64995776
Idle time:1264041248

**Universal Bus Channel: OPB**
Program/Data:0
Data transfer time:64995636
Idle time:1264041388

**Transducer: Tx1**
Fifo check time:2471759
Fifo read write time:80286

****END TIME:1329037024 ns
SystemC: simulation stopped by user.
Timed TLM simulation is successful
Files test/classic1.out and test/classic1_pcm.gold are identical
Simulation exited with status 0
Press return to continue ...
```

```
% xterm -title mp3_hsd_platform -e /bin/sh -c sim_perf_TLM /data/users/yjahn/work/esedemo/esedemo/mp3_hsd_platform.ed
s ../test/classic1.mp3 ../test; diff -s test/classic1.out test/classic1_pcm.gold; echo "Simulation exited with status
 $?" ;echo "Press return to continue ..." ;read confirm
```

ESE Generated PCAM



Xilinx Platform Studio - H:/ese/examples/MP3_PLATFORM2_SYNTHESIZED/system.xmp - [System Assembly View1]

File Edit View Project Hardware Software Device Configuration Debug Simulation Window Help
Project
Applications
IP Catalog
O P B
L M B
L M B

Platform

- Project Files
  - MHS File: system.mhs
  - MSS File: system.mss
  - UCF File: data/system.ucf
  - iMPACT Command File: etc/download.cmd
  - Implementation Options File: etc/fast_runtim
  - Bitgen Options File: etc/bitgen.ut
- Project Options
  - Device: xc2v2000ff896-4
  - Netlist: TopLevel
  - Implementation: XPS
  - HDL: VHDL
  - Sim Model: BEHAVIORAL
- Reference Files
  - Log Files
  - Report Files

Filters: ● Bus Interface ○ Ports ○ Addresses

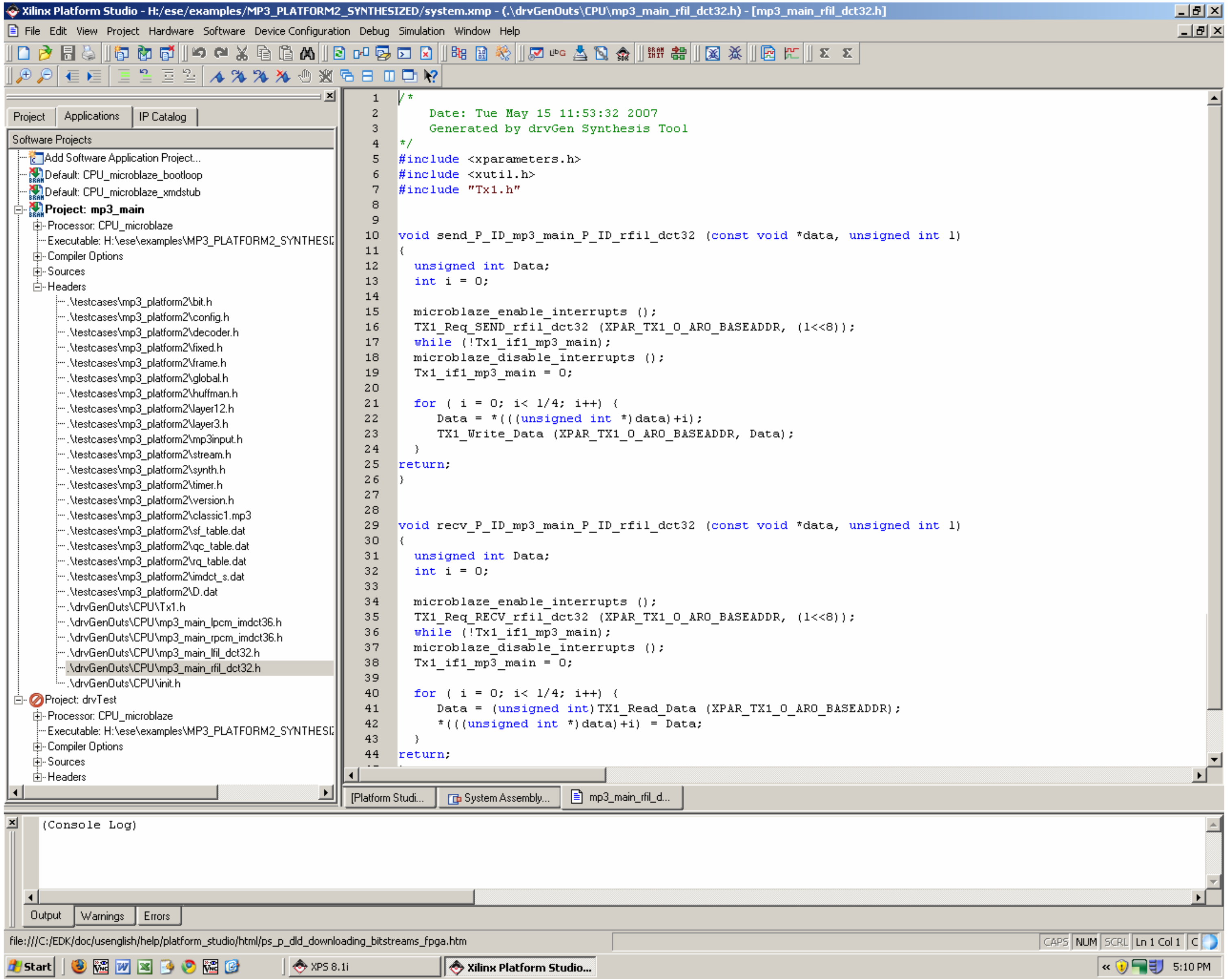
Name	Bus Connection	Direction	IP Type	IP Version
CPU_microblaze			microblaze	4.00.a
OP_mb_opb			opb_v20	1.10.c
CPU_ilmb			lmb_v10	1.00.a
CPU_dlmb			lmb_v10	1.00.a
RS232			opb_uartlite	1.00.b
LEDs_2Bit			opb_gpio	3.01.b
opb_timer_0			opb_timer	1.00.b
opb_deltasigma_dac_0			opb_deltasigma_dac	1.01.a
Tx1_0			Tx1	1.00.a
CPU_ilmb_cntlr			lmb_bram_if_cntlr	1.00.b
CPU_dlmb_cntlr			lmb_bram_if_cntlr	1.00.b
CPU_debug_module			opb_mdm	2.00.a
CPU_ZBT_512Kx32			opb_emc	2.00.a
dcm_0			dcm_module	1.00.a
CPU_lmb_bram			bram_block	1.00.a
CPU_ZBT_512Kx32_util_bus_split_0			util_bus_split	1.00.a

Legend: □ Master ○ Slave Master/Slave Target < Initiator ○ Connected ○ Unconnected

```
(Console Log)
```

file:///C:/EDK/doc/usenglish/help/platform_studio/html/ps_p_dld_downloading_bitstreams_fpga.htm

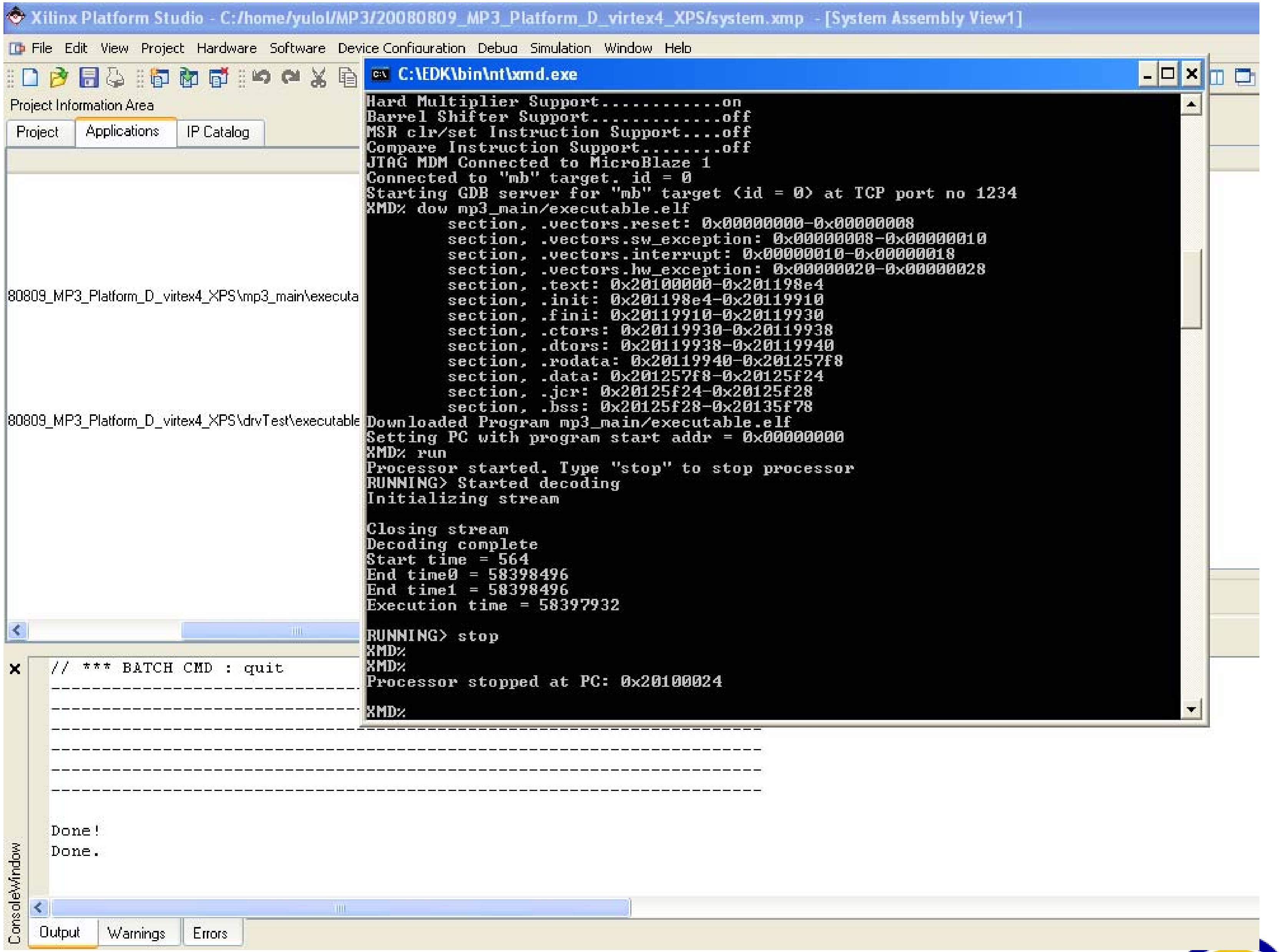
ESE Generated HdS Code


Xilinx Platform Studio - H:/ese/examples/MP3_PLATFORM2_SYNTHESIZED/system.xmp - (.\drvGenOuts\CPU\mp3_main_rfil_dct32.h) - [mp3_main_rfil_dct32.h]
File Edit View Project Hardware Software Device Configuration Debug Simulation Window Help
Project
Applications
IP Catalog
Software Projects
Add Software Application Project...
Default: CPU_microblaze_bootloop
Default: CPU_microblaze_xmdstub
Project: mp3_main
Processor: CPU_microblaze
Executable: H:\ese\examples\MP3_PLATFORM2_SYNTHESI
Compiler Options
Sources
Headers
.\testcases\mp3_platform2\bit.h
.\testcases\mp3_platform2\config.h
.\testcases\mp3_platform2\decoder.h
.\testcases\mp3_platform2\fixed.h
.\testcases\mp3_platform2\frame.h
.\testcases\mp3_platform2\global.h
.\testcases\mp3_platform2\huffman.h
.\testcases\mp3_platform2\layer12.h
.\testcases\mp3_platform2\layer3.h
.\testcases\mp3_platform2\mp3input.h
.\testcases\mp3_platform2\stream.h
.\testcases\mp3_platform2\synth.h
.\testcases\mp3_platform2\timer.h
.\testcases\mp3_platform2\version.h
.\testcases\mp3_platform2\classic1.mp3
.\testcases\mp3_platform2\sf_table.dat
.\testcases\mp3_platform2\qc_table.dat
.\testcases\mp3_platform2\rq_table.dat
.\testcases\mp3_platform2\imdct_s.dat
.\testcases\mp3_platform2\D.dat
.\drvGenOuts\CPU\Tx1.h
.\drvGenOuts\CPU\mp3_main_lpcm_imdct36.h
.\drvGenOuts\CPU\mp3_main_rpcm_imdct36.h
.\drvGenOuts\CPU\mp3_main_lfil_dct32.h
.\drvGenOuts\CPU\mp3_main_rfil_dct32.h
.\drvGenOuts\CPU\init.h
Project: drvTest
Processor: CPU_microblaze
Executable: H:\ese\examples\MP3_PLATFORM2_SYNTHESI
Compiler Options
Sources
Headers
[Platform Studi...
System Assembly...
mp3_main_rfil_d...
Output
Warnings
Errors
file:///C:/EDK/doc/usenglish/help/platform_studio/html/ps_p_dld_downloading_bitstreams_fpga.htm
CAPS NUM SCRL Ln 1 Col 1
Start
XPS 8.1i
Xilinx Platform Studio...
5:10 PM

```
/*
    Date: Tue May 15 11:53:32 2007
    Generated by drvGen Synthesis Tool
*/
#include <xparameters.h>
#include <xutil.h>
#include "Tx1.h"


void send_P_ID_mp3_main_P_ID_rfil_dct32 (const void *data, unsigned int l)
{
  unsigned int Data;
  int i = 0;

  microblaze_enable_interrupts ();
  TX1_Req_SEND_rfil_dct32 (XPAR_TX1_0_ARO_BASEADDR, (1<<8));
  while (!Tx1_if1_mp3_main);
  microblaze_disable_interrupts ();
  Tx1_if1_mp3_main = 0;

  for ( i = 0; i< l/4; i++) {
     Data = *(((unsigned int *)data)+i);
     TX1_Write_Data (XPAR_TX1_0_ARO_BASEADDR, Data);
  }
return;
}


void recv_P_ID_mp3_main_P_ID_rfil_dct32 (const void *data, unsigned int l)
{
  unsigned int Data;
  int i = 0;

  microblaze_enable_interrupts ();
  TX1_Req_RECV_rfil_dct32 (XPAR_TX1_0_ARO_BASEADDR, (1<<8));
  while (!Tx1_if1_mp3_main);
  microblaze_disable_interrupts ();
  Tx1_if1_mp3_main = 0;

  for ( i = 0; i< l/4; i++) {
     Data = (unsigned int)TX1_Read_Data (XPAR_TX1_0_ARO_BASEADDR);
     *(((unsigned int *)data)+i) = Data;
  }
return;
```

```
(Console Log)
```

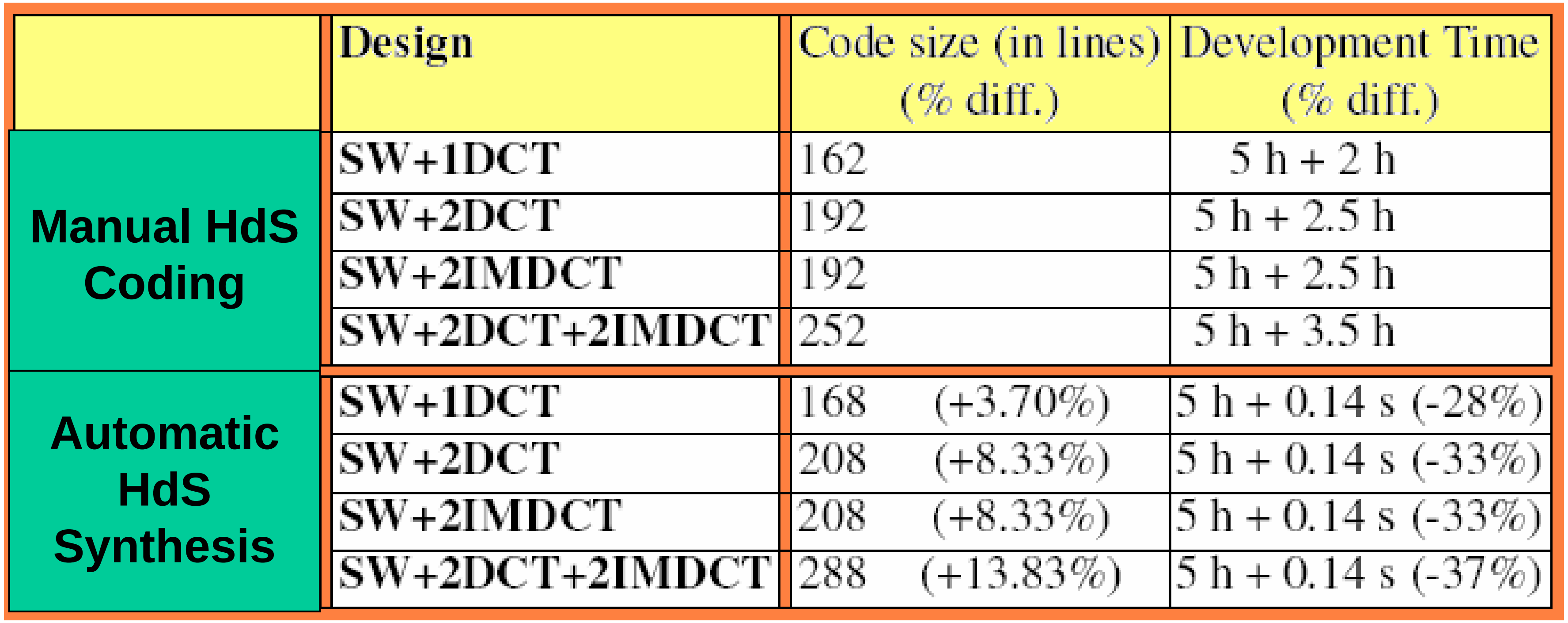
PCAM Prototyping and Execution


Xilinx Platform Studio - C:/home/yulol/MP3/20080809_MP3_Platform_D_virtex4_XPS/system.xmp - [System Assembly View1]
File Edit View Project Hardware Software Device Configuration Debug Simulation Window Help
Project Information Area
Project Applications IP Catalog
80809_MP3_Platform_D_virtex4_XPS\mp3_main\executa
80809_MP3_Platform_D_virtex4_XPS\drvTest\executable
C:\EDK\bin\nt\xmd.exe
ConsoleWindow
Output Warnings Errors

```
Hard Multiplier Support.............on
Barrel Shifter Support.............off
MSR clr/set Instruction Support....off
Compare Instruction Support........off
JTAG MDM Connected to MicroBlaze 1
Connected to "mb" target. id = 0
Starting GDB server for "mb" target (id = 0) at TCP port no 1234
XMD% dow mp3_main/executable.elf
        section, .vectors.reset: 0x00000000-0x00000008
        section, .vectors.sw_exception: 0x00000008-0x00000010
        section, .vectors.interrupt: 0x00000010-0x00000018
        section, .vectors.hw_exception: 0x00000020-0x00000028
        section, .text: 0x20100000-0x201198e4
        section, .init: 0x201198e4-0x20119910
        section, .fini: 0x20119910-0x20119930
        section, .ctors: 0x20119930-0x20119938
        section, .dtors: 0x20119938-0x20119940
        section, .rodata: 0x20119940-0x201257f8
        section, .data: 0x201257f8-0x20125f24
        section, .jcr: 0x20125f24-0x20125f28
        section, .bss: 0x20125f28-0x20135f78
Downloaded Program mp3_main/executable.elf
Setting PC with program start addr = 0x00000000
XMD% run
Processor started. Type "stop" to stop processor
RUNNING> Started decoding
Initializing stream

Closing stream
Decoding complete
Start time = 564
End time0 = 58398496
End time1 = 58398496
Execution time = 58397932

RUNNING> stop
XMD%
XMD%
Processor stopped at PC: 0x20100024

XMD%
```

```
// *** BATCH CMD : quit
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------


Done!
Done.
```

Manual vs synthesized HdS



	Design	Code size (in lines) (% diff.)	Development Time (% diff.)
Manual HdS Coding	SW+1DCT	162	5 h + 2 h
	SW+2DCT	192	5 h + 2.5 h
	SW+2IMDCT	192	5 h + 2.5 h
	SW+2DCT+2IMDCT	252	5 h + 3.5 h
Automatic HdS Synthesis	SW+1DCT	168 (+3.70%)	5 h + 0.14 s (-28%)
	SW+2DCT	208 (+8.33%)	5 h + 0.14 s (-33%)
	SW+2IMDCT	208 (+8.33%)	5 h + 0.14 s (-33%)
	SW+2DCT+2IMDCT	288 (+13.83%)	5 h + 0.14 s (-37%)

- Synthesized HdS is marginally larger than manual code, with identical performance
- Manual development time
  - Platform design (5 hours), HdS implementation (2-4 hours vs. <1 sec)
  - Overall development time savings of 33% on average
- Higher productivity gain for more complex examples

Conclusion

We presented a model based design methodology which:

- supports HdS synthesis for heterogeneous many-core platforms
- contains well defined system models at 3 abstraction levels
- Supported by ESE toolset , available for free download

Results of HdS synthesis for MP3 decoder example:

- Over 30% reduction in overall design time
- Code quality comparable to manual implementation

Future work:

- Extend design framework with to security oriented application models
- Provide support for platform templates for real-time architectures

Thank You

Visit http://www.cecs.uci.edu/~ese