

PHILIPS

SystemC based Platform Modeling: Issues related with Hardware Modeling

Sandeep Dube: sandeep.dube@philips.com

Karthick Gururaj: karthick.gururaj@philips.com

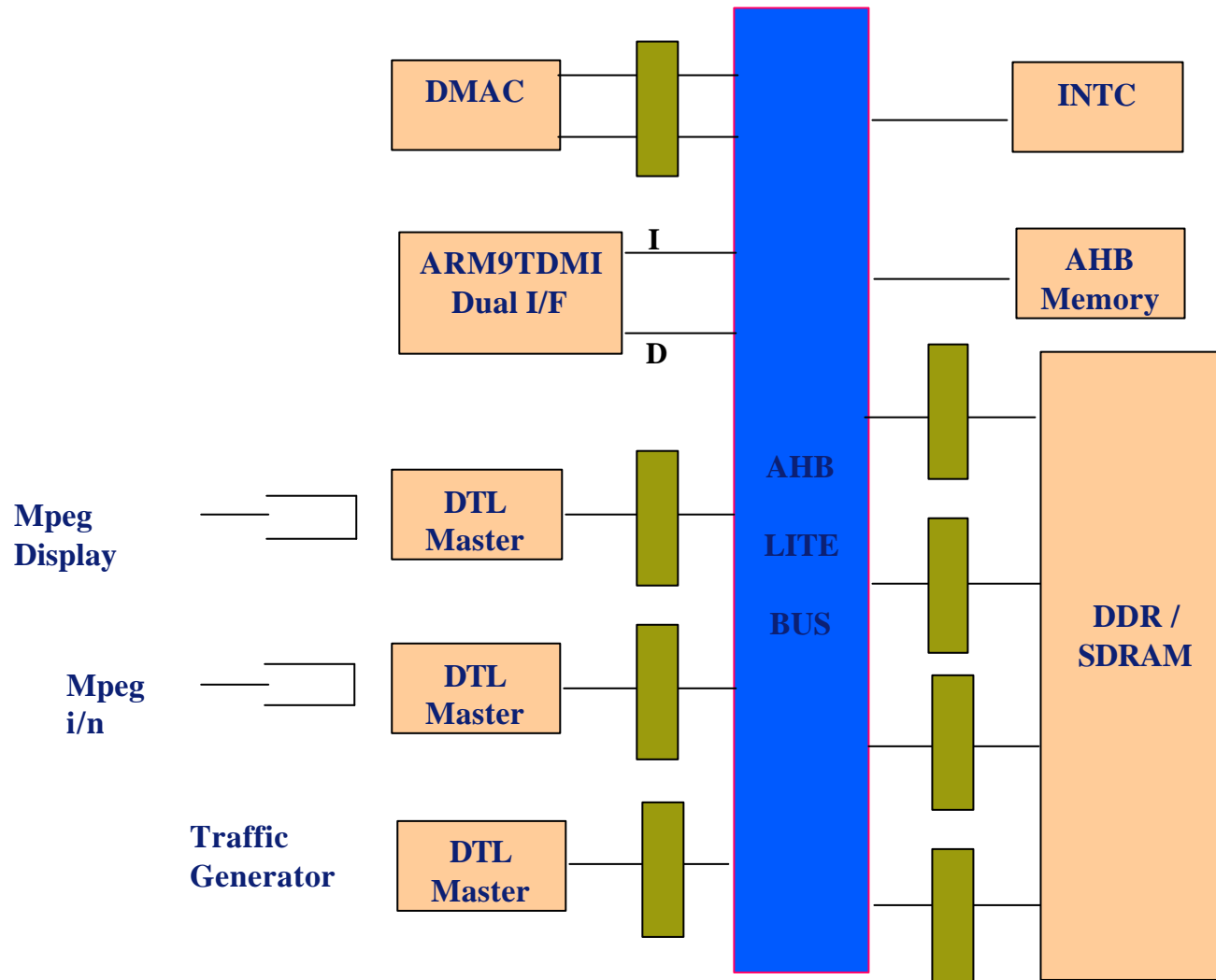
Pallavi Desai: pallavi.desai@philips.com

ESCUG, November 2003

Agenda

- Introduction
- Hardware modeling in SystemC
- Integration issues
- Adapting non-SystemC models
- Conclusion
- Q & A

Introduction

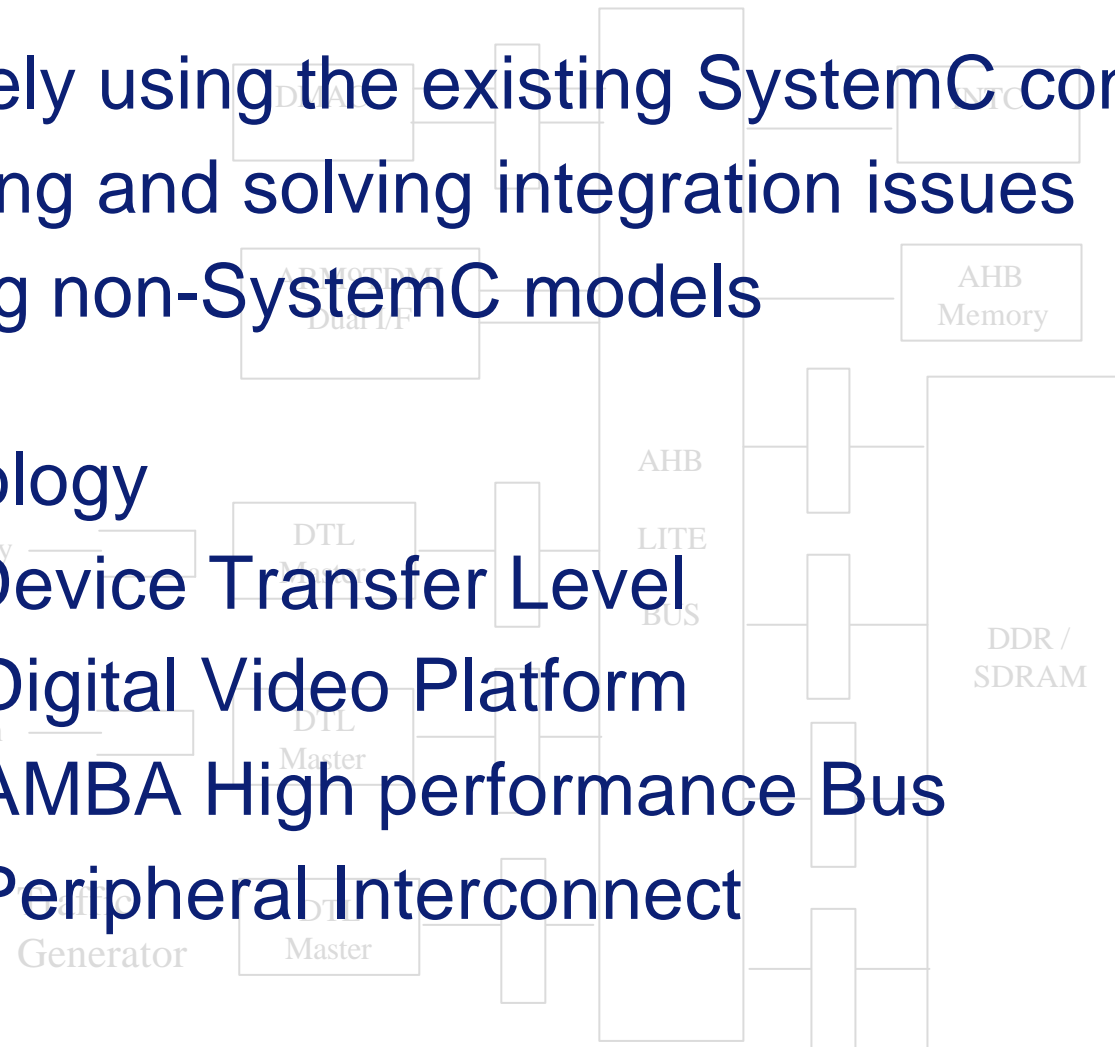


Introduction

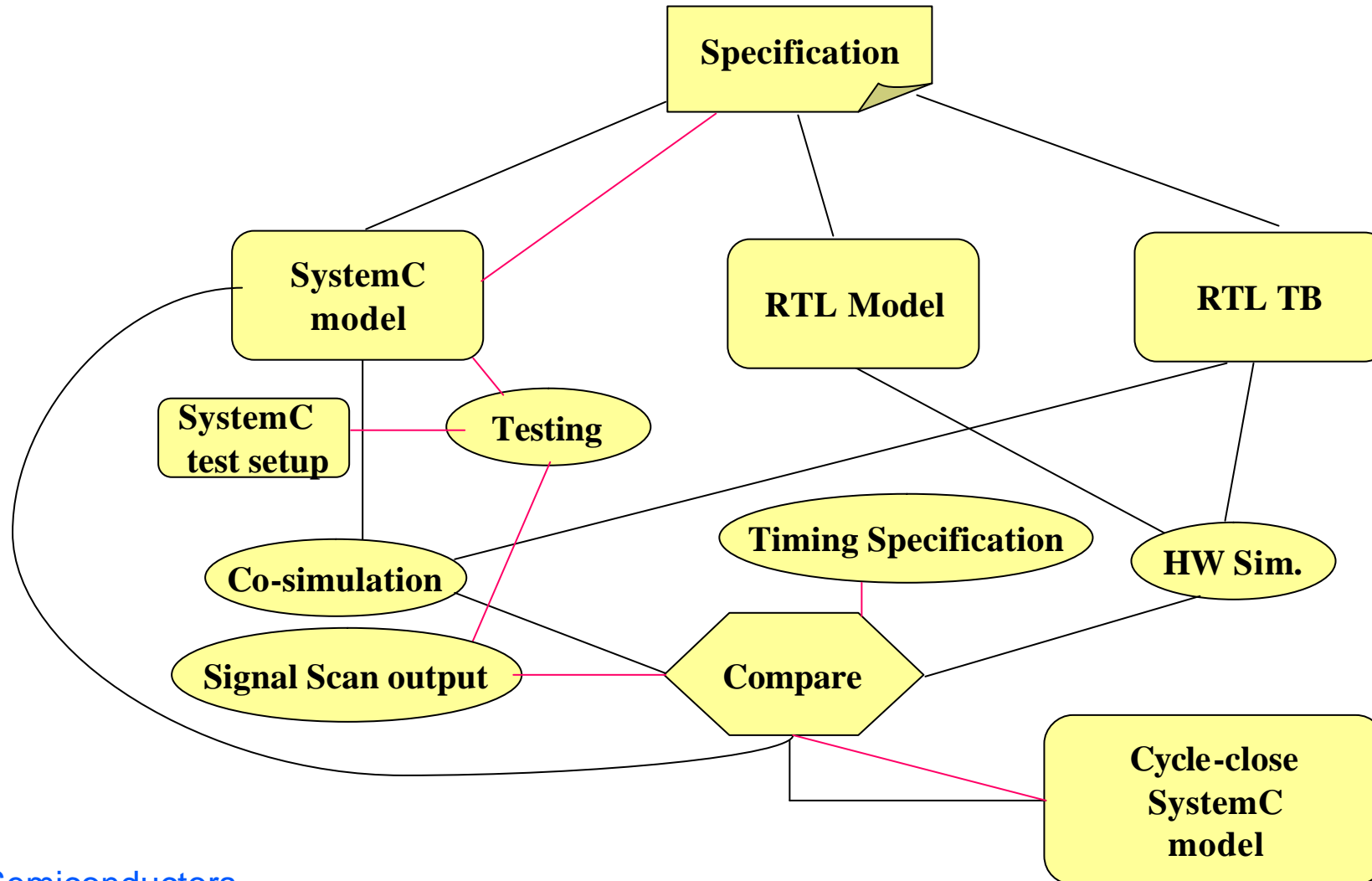
- Effectively using the existing SystemC constructs
- Identifying and solving integration issues
- Adapting non-SystemC models

Terminology

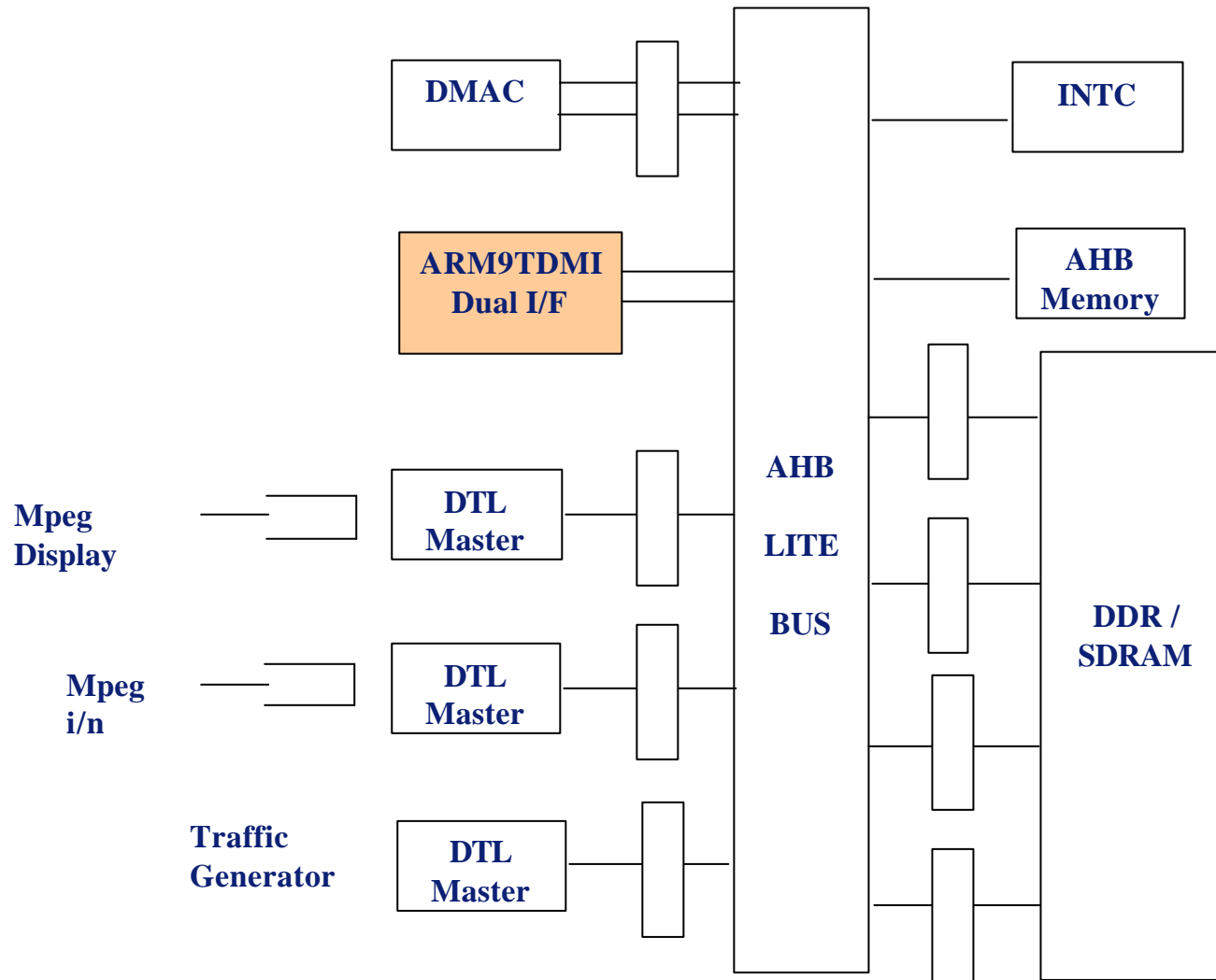
- DTL – Device Transfer Level
- DVP – Digital Video Platform
- AHB – AMBA High performance Bus
- PI – Peripheral Interconnect



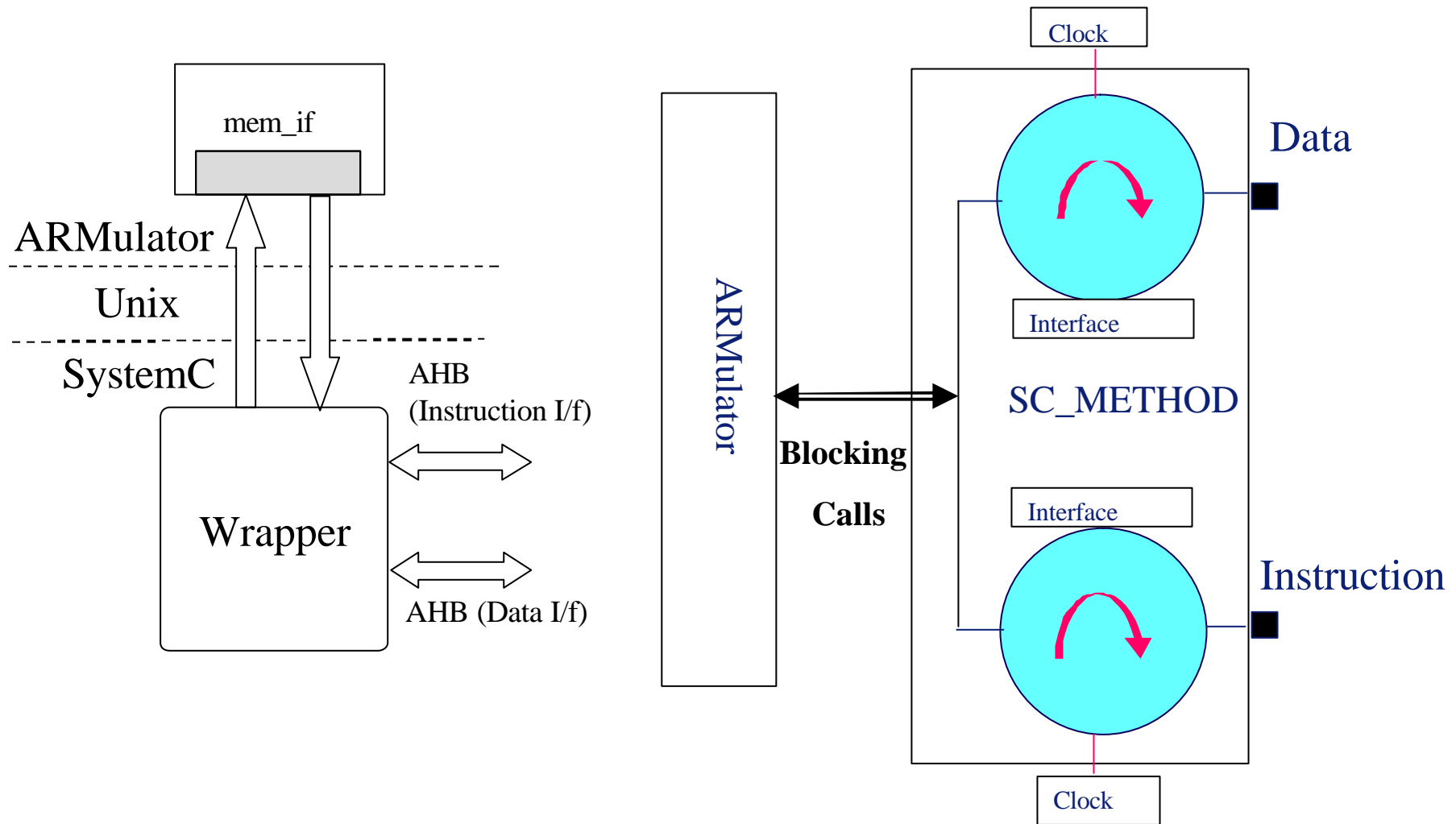
Hardware modeling in SystemC



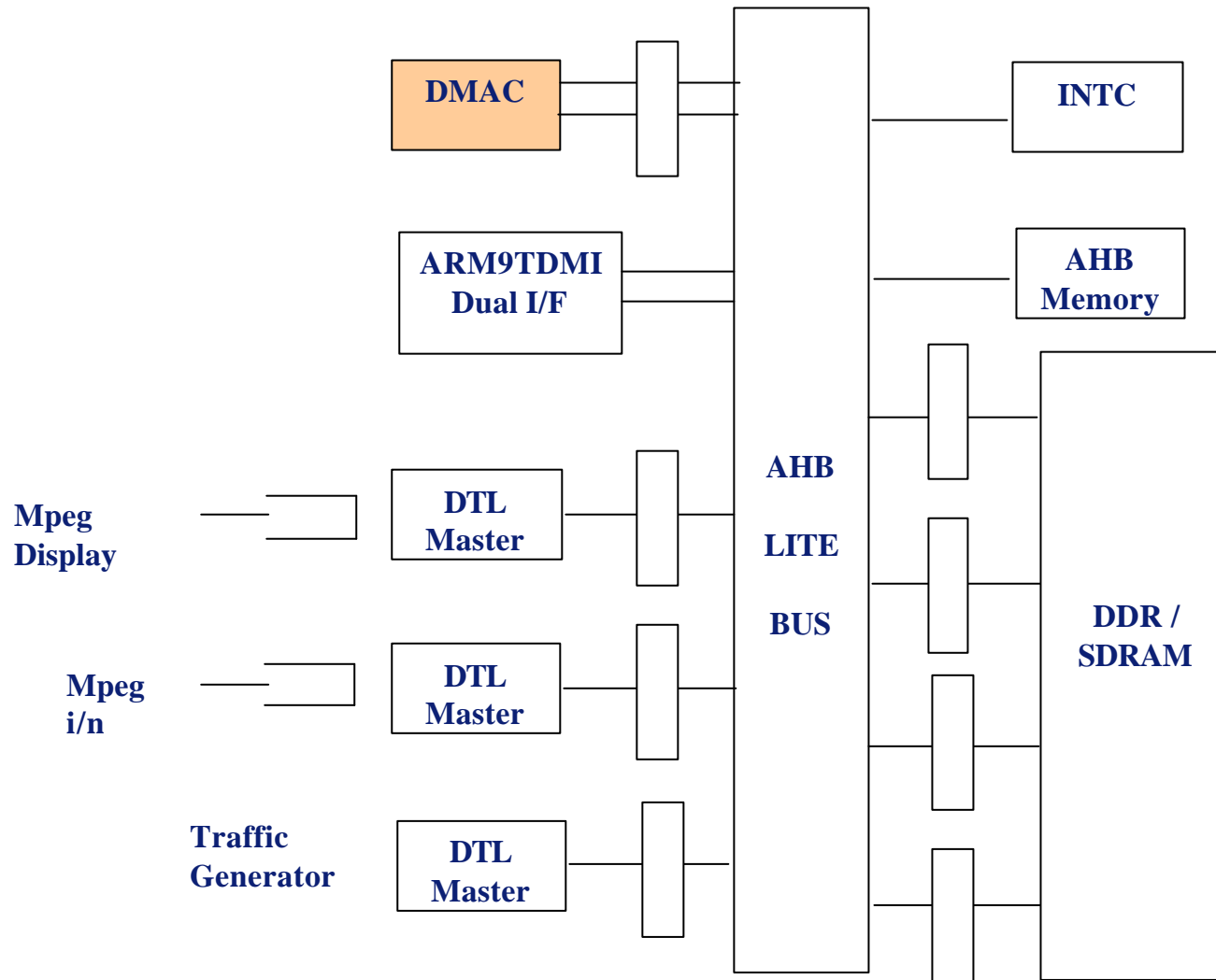
ARM9TDMI - Context



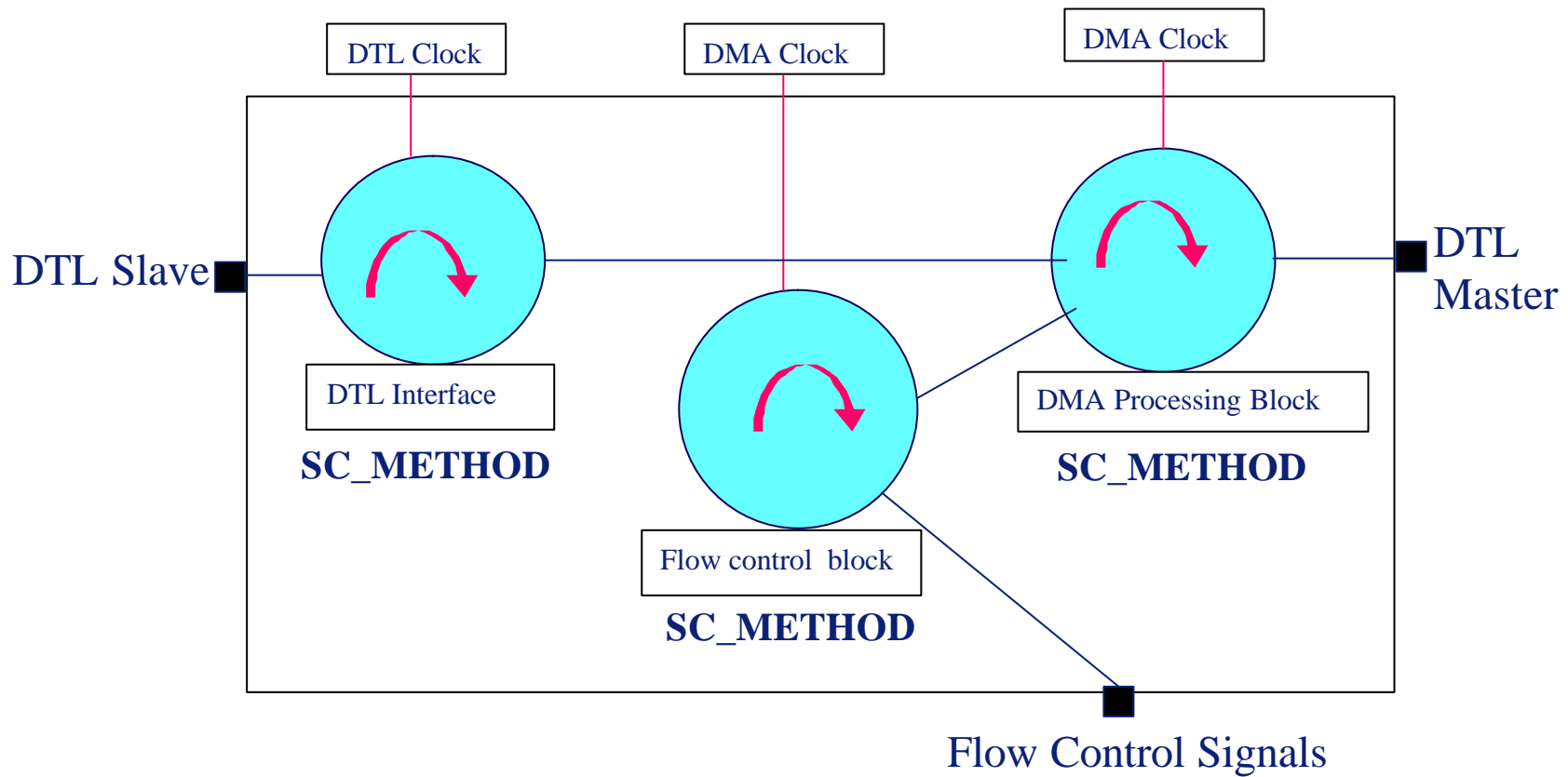
ARM9TDMI



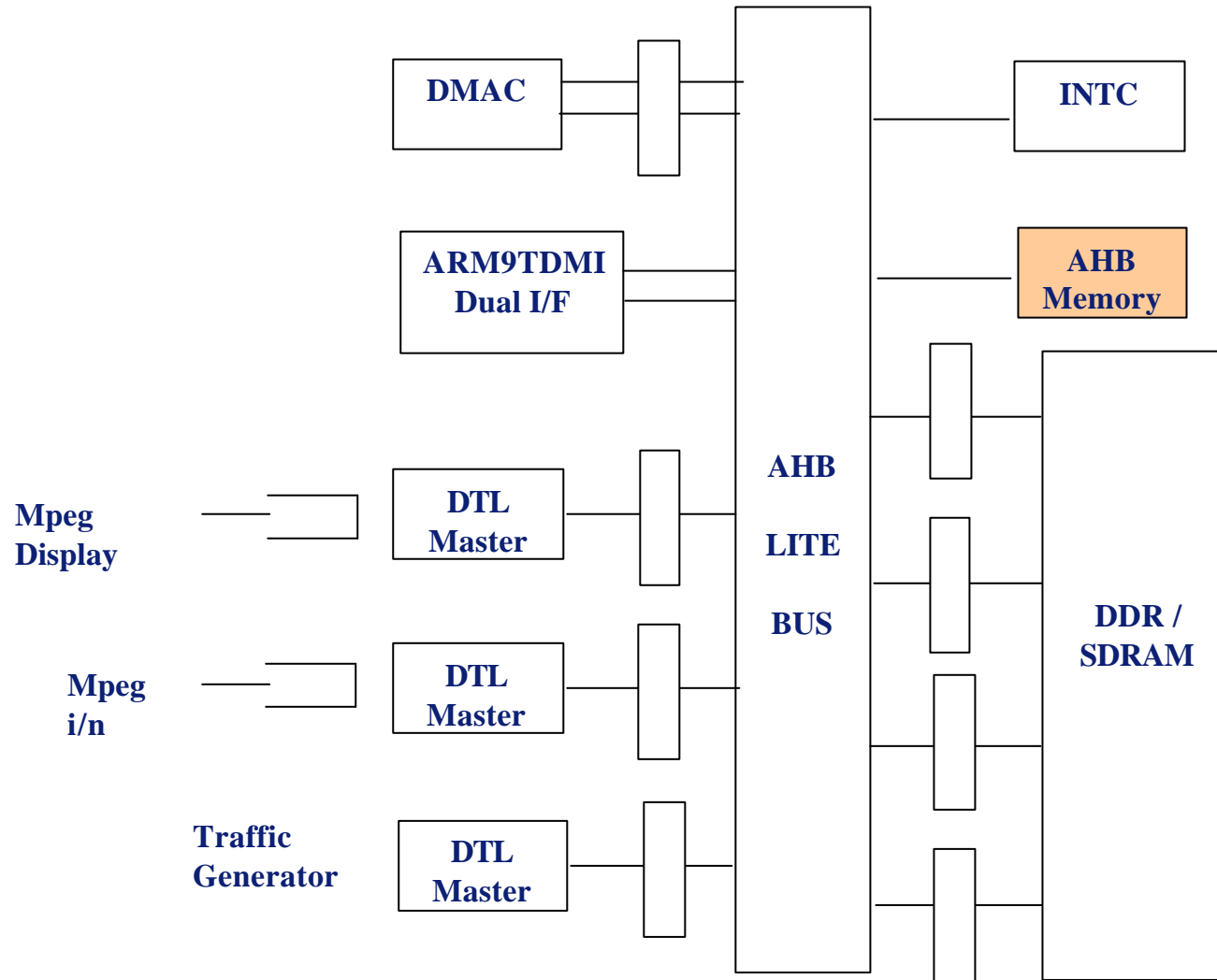
DMA Controller - Context



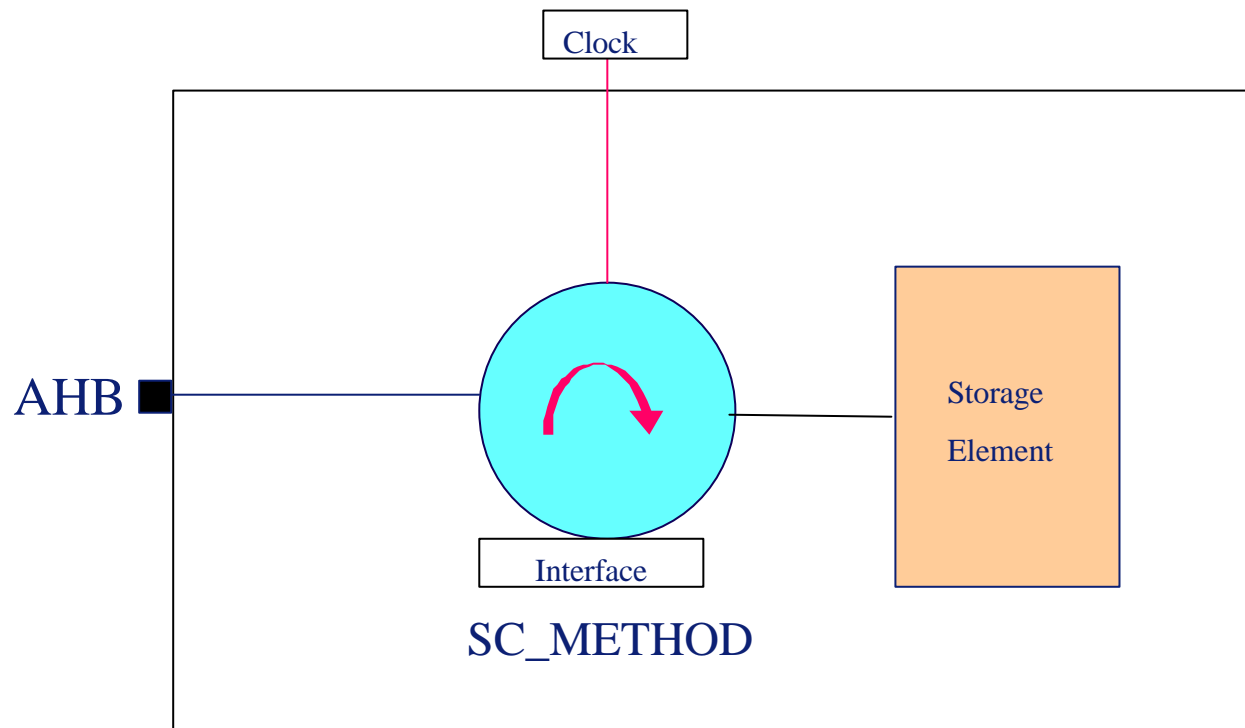
DMA Controller



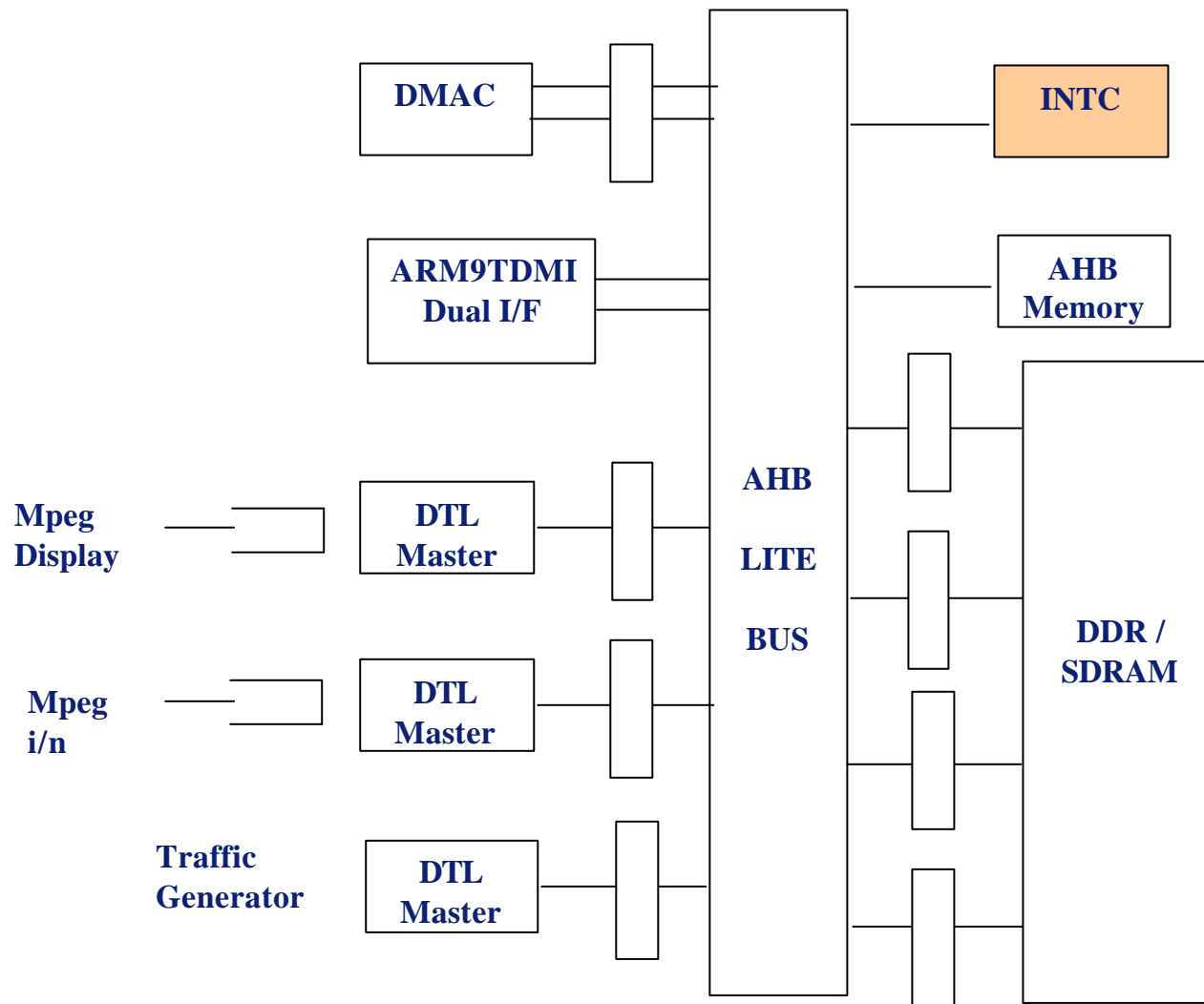
AHB Generic Memory - Context



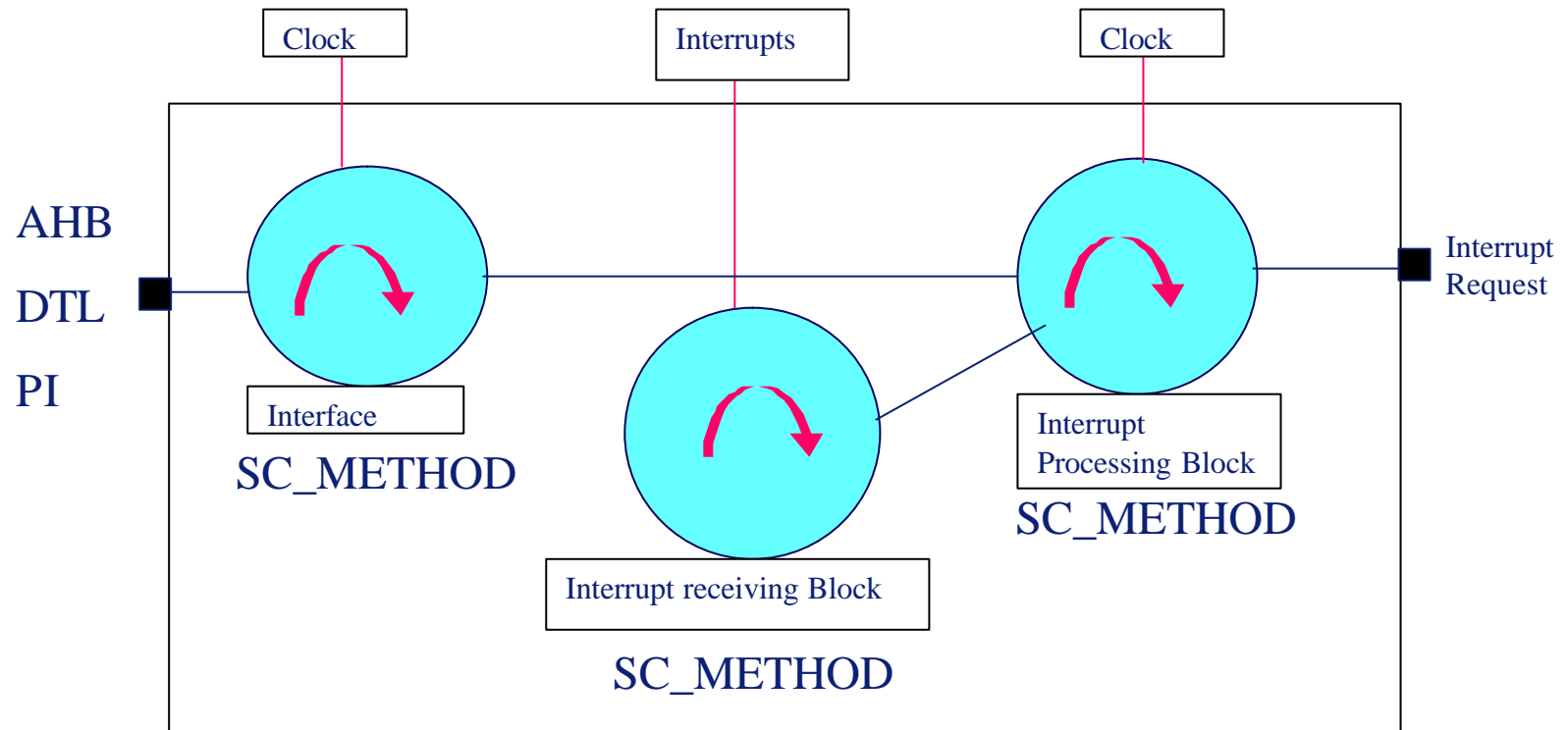
AHB Generic Memory



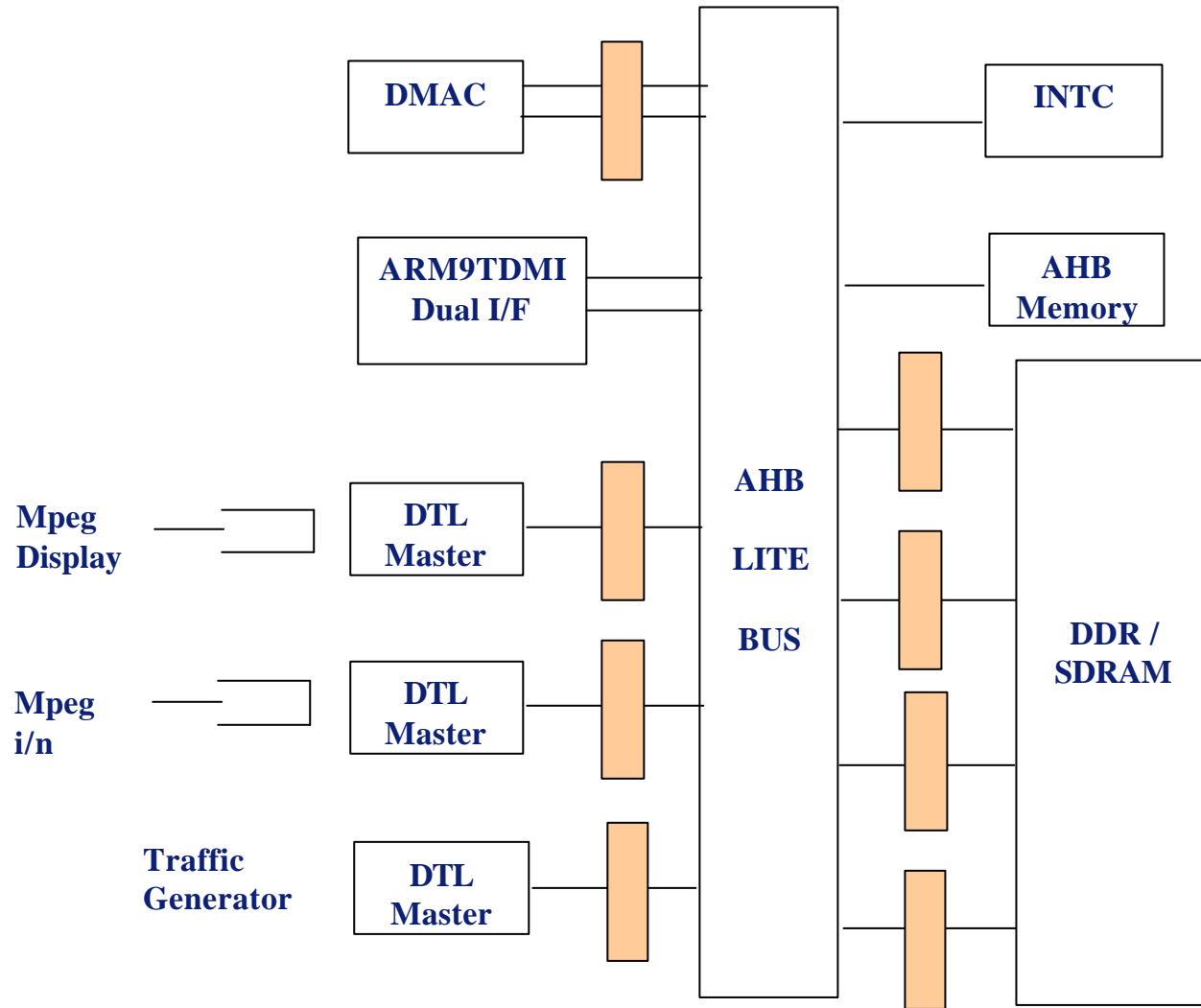
Interrupt Controller - Context



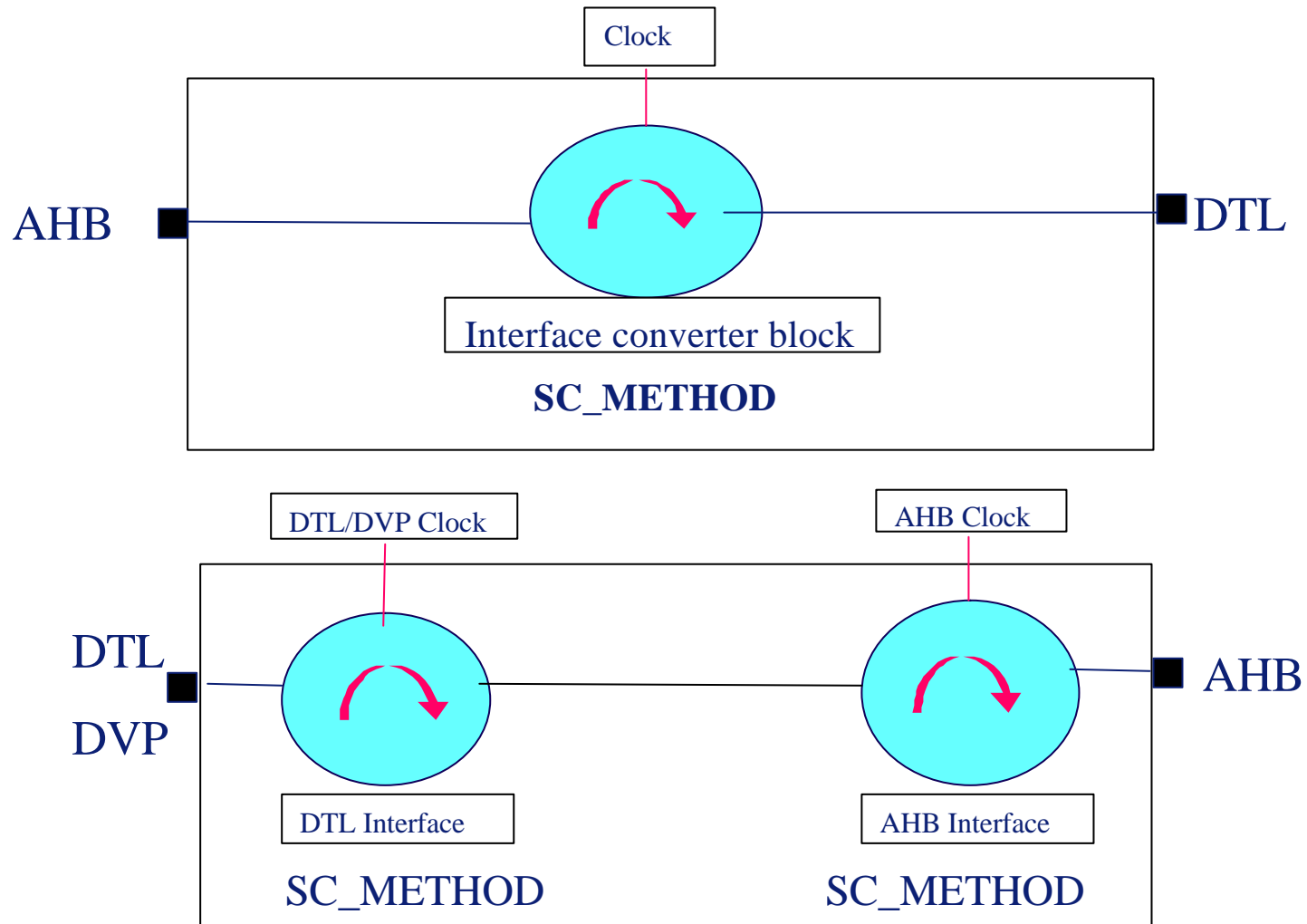
Interrupt controller



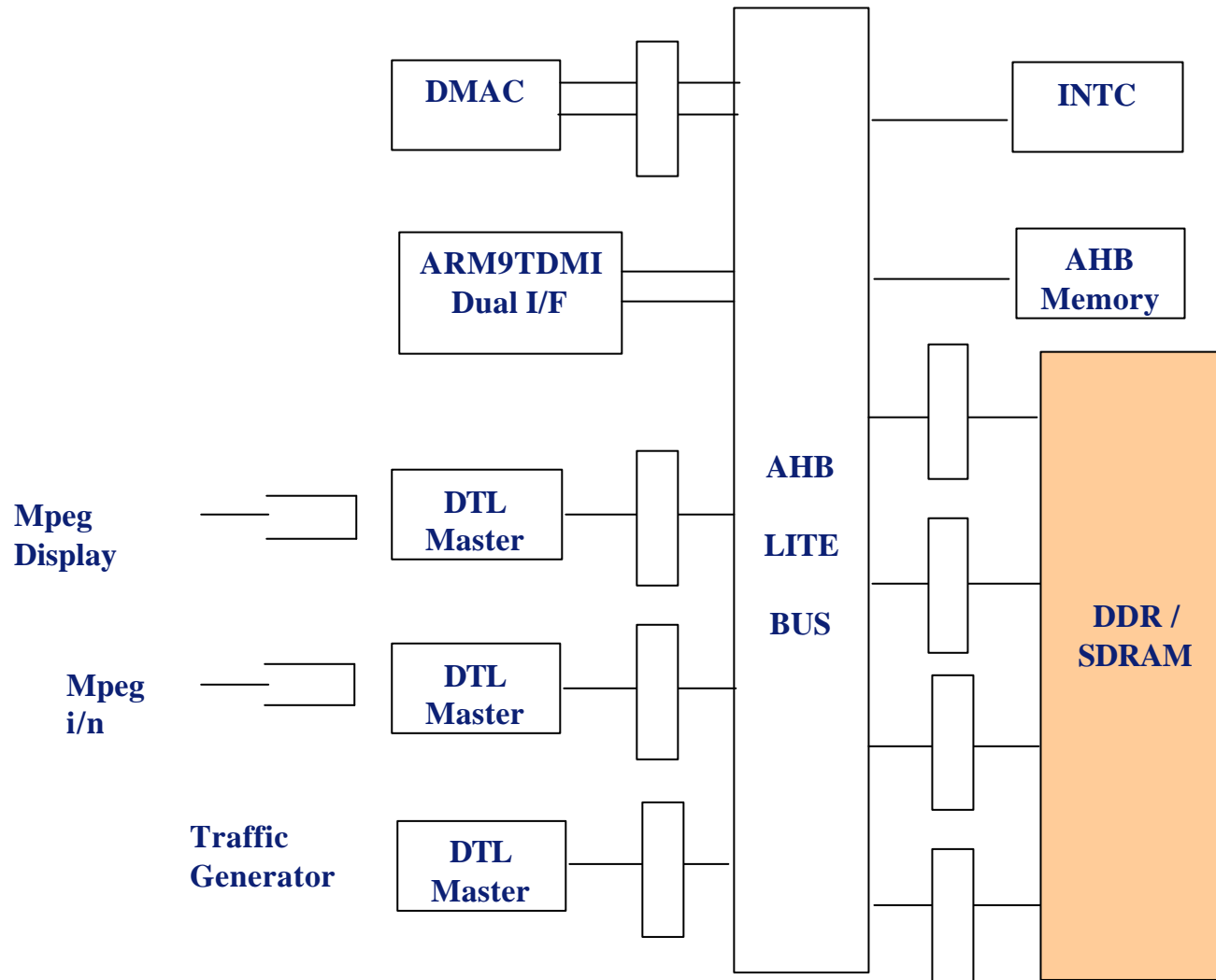
Adapters - Context



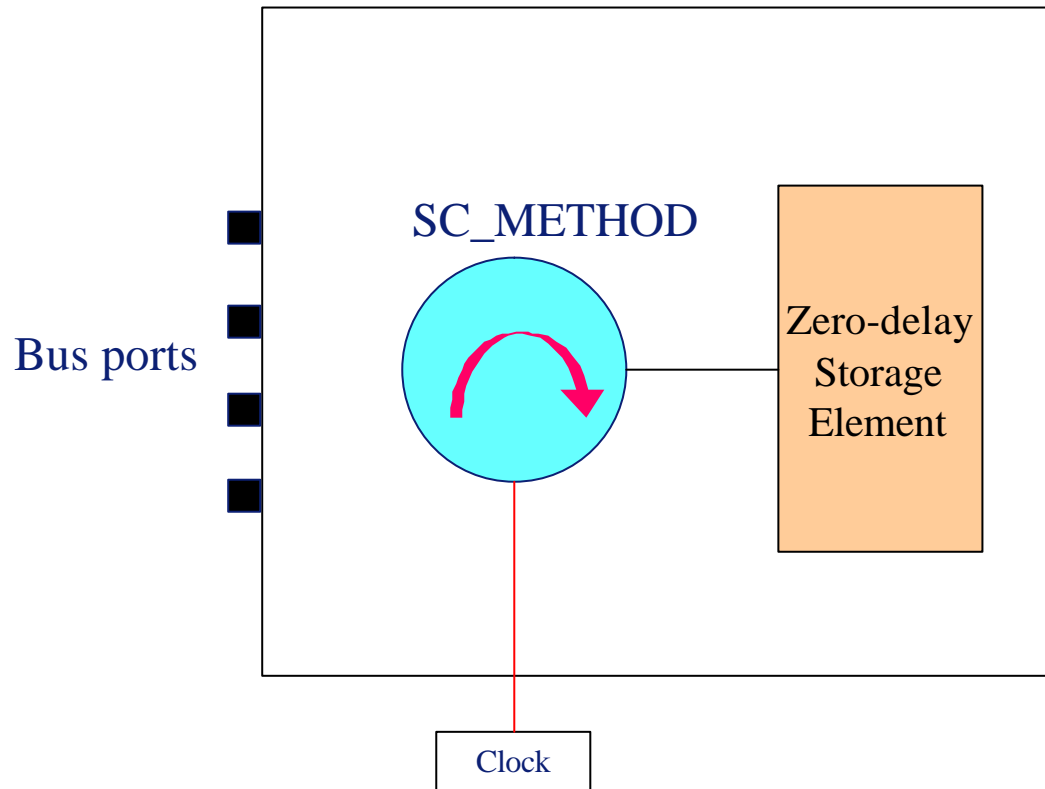
Adapters



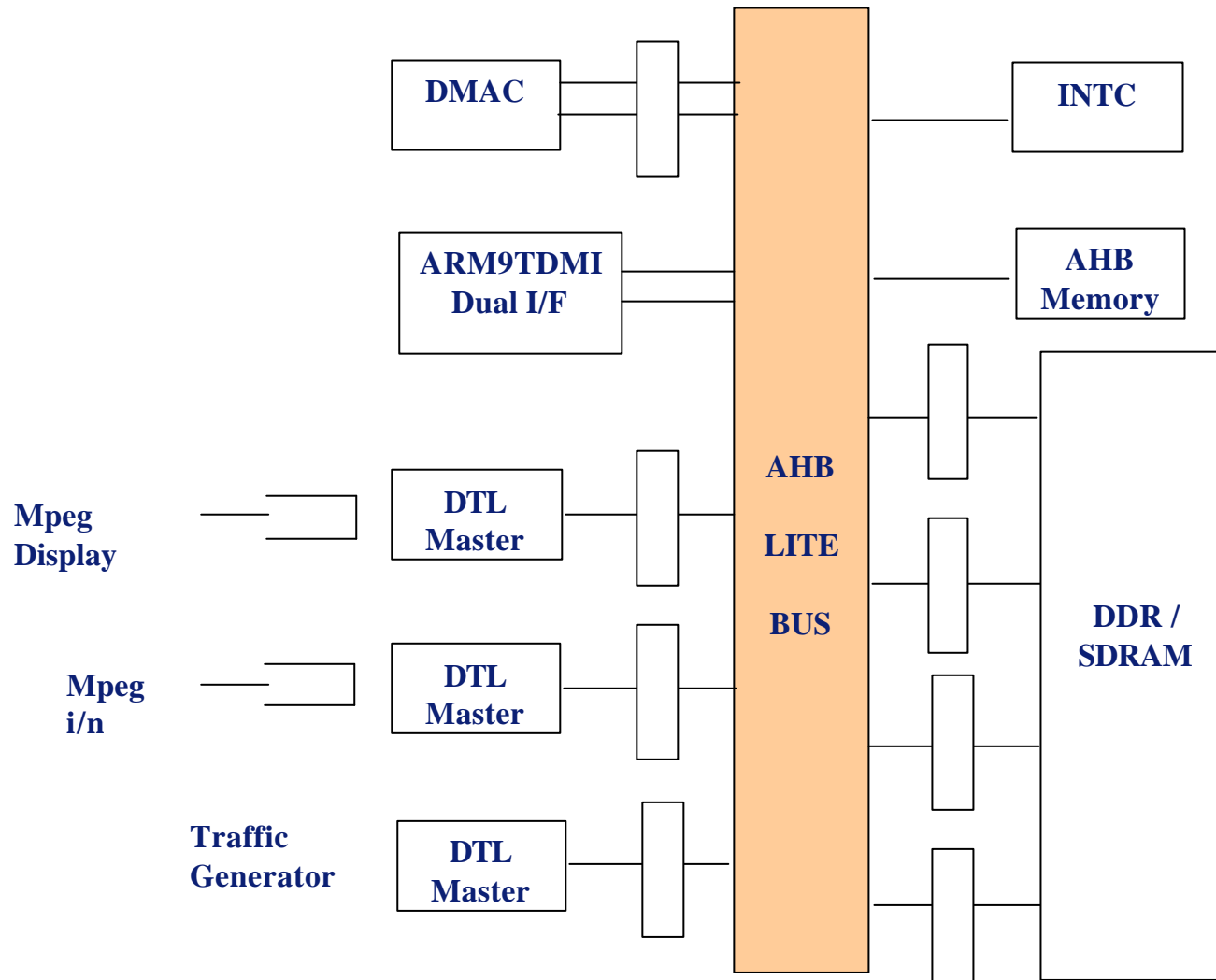
DDR/SDRAM Controller - Context



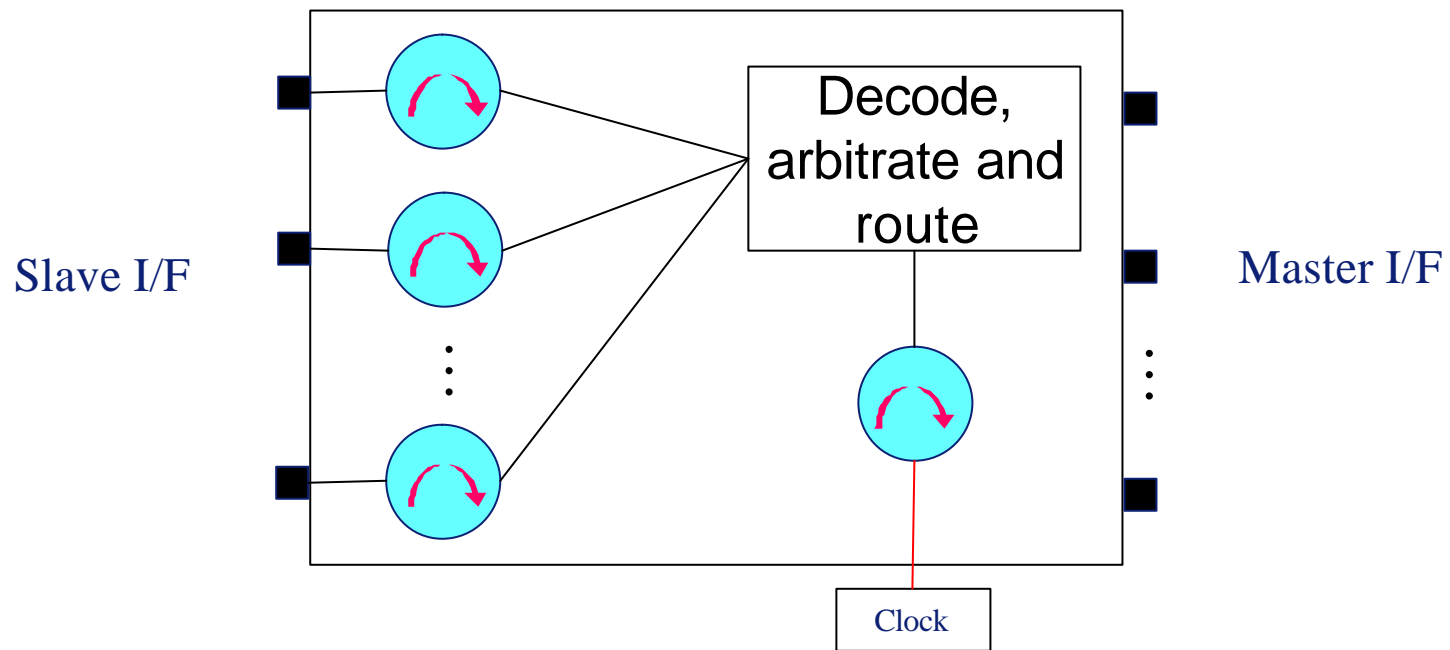
DDR/SDRAM Controller



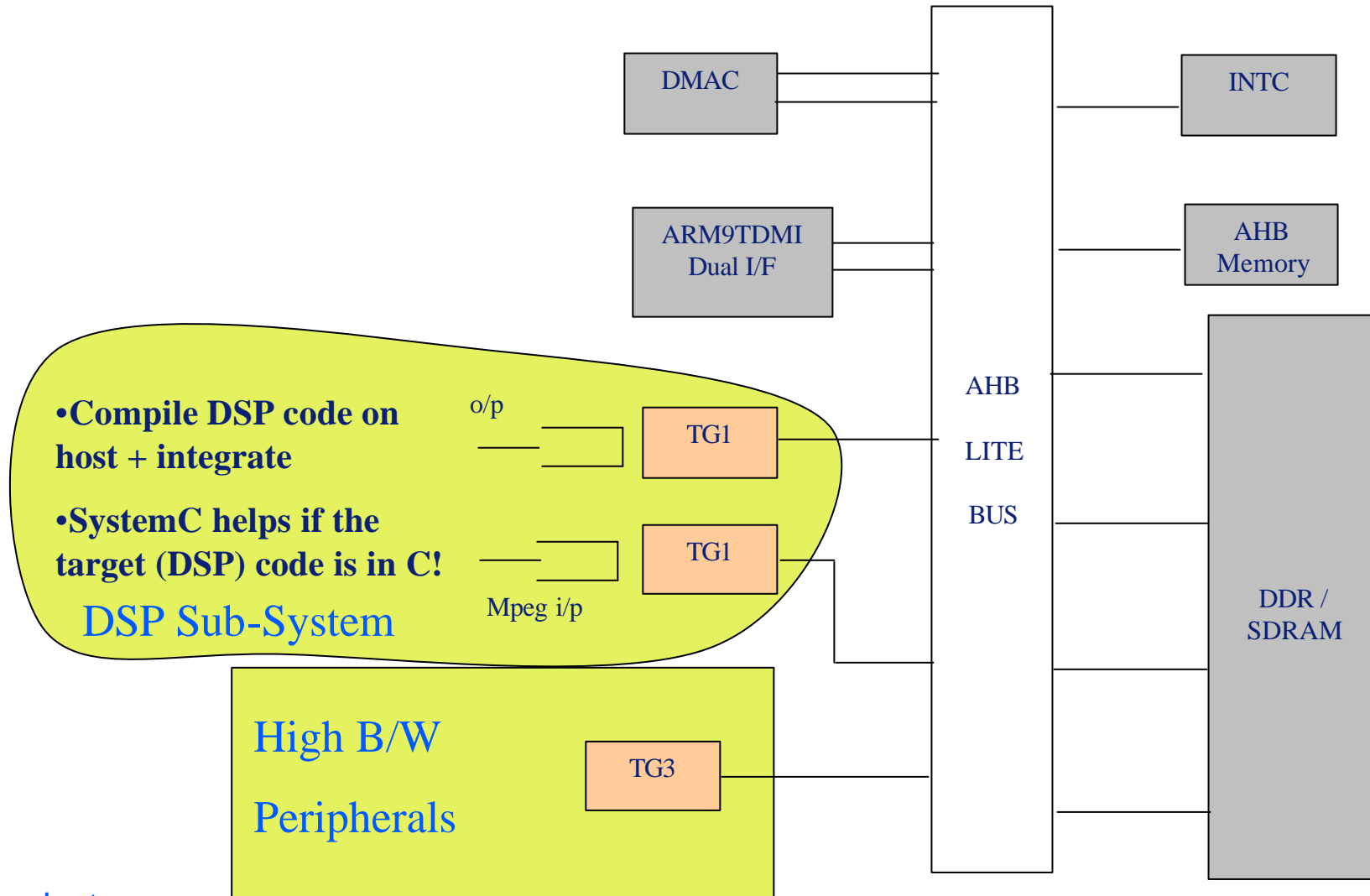
AHB Lite Bus - Context



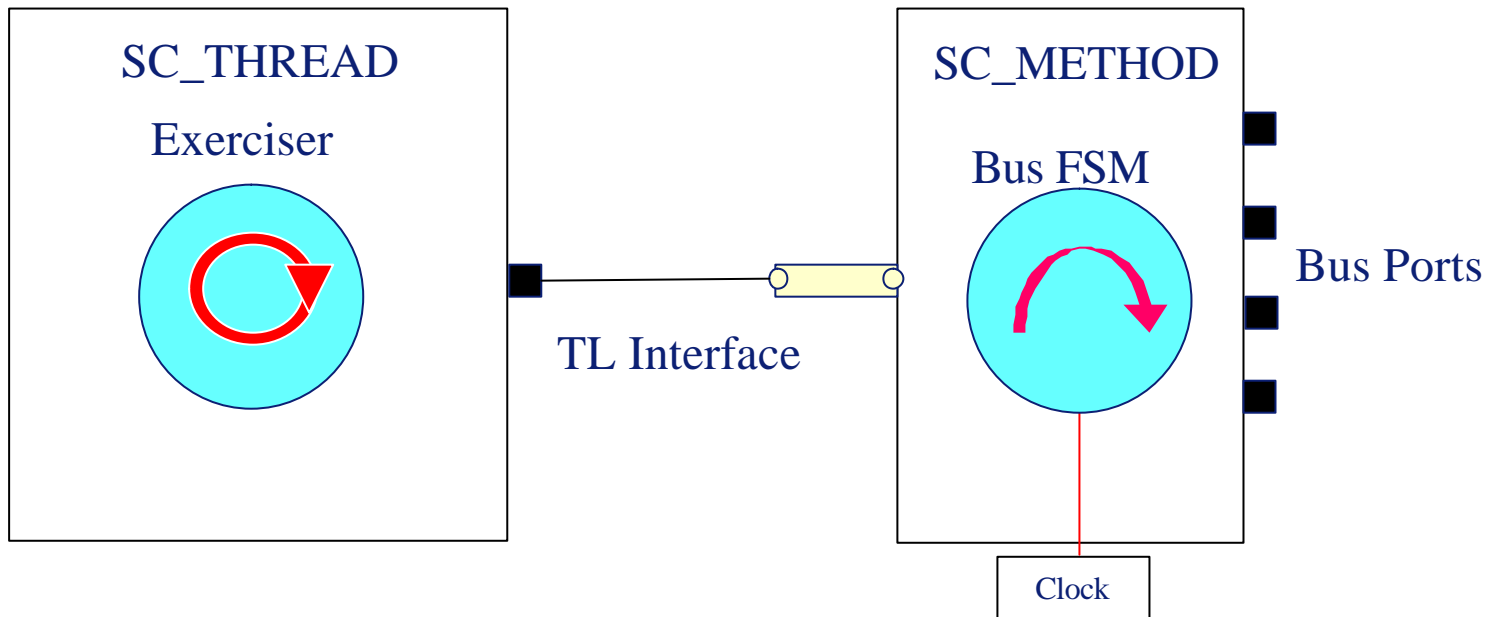
AHB Lite



Traffic Generators - Context



Traffic Generators



Integration Issues

- Assumptions lead to expensive reworks
 - Too many ways to define ports
 - Connecting address lines
- Offset addresses
- Single stepping and looking at internal values
- `sc_buffer` vs. `sc_signal`
- Co-simulation issues
 - Active high and Active Low
 - Naming of ports

Issues in adapting non-SystemC models

- Simulation speed
 - Standalone executable
 - Dynamic linked library
 - Statically Linked Library
- Simulation abstraction
- Debug features



Speed

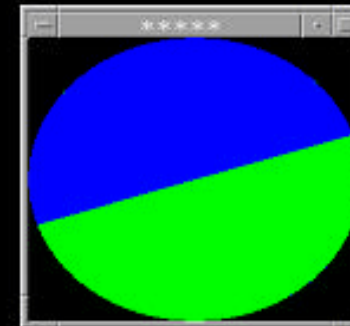
Applications Running on the Architecture and Performance Measurements

- Applications
 - IS-95 Viterbi Decoder
 - Simple Sorting
 - MPEG2 streaming through IPs
- Performance Measurement
 - DDRC memory performance
 - Bus load
 - Transaction latencies
 - Effect of parameters on the performance

System Running With Displays

```

armsd: let 0x80000008 = 0xE0008000
armsd: let 0x80000008 = 0xE0008000
armsd:
armsd: | destination address 0xE100d1d0
armsd: let 0x8000000c = 0x0000d1d0
armsd: let 0x8000000c = 0x0000d1d0
armsd:
armsd: let 0x80000010 = 0x00200020
armsd: let 0x80000010 = 0x00200020
armsd:
armsd: || block size 688 = 0x2b0
armsd: let 0x80000014 = 0x00000018
armsd: let 0x80000014 = 0x00000018
armsd:
armsd:
armsd: let 0x80000400 = 0x00000001
armsd: let 0x80000400 = 0x00000001
armsd:
armsd:
armsd: let 0x80000418 = 0x00000006
armsd: let 0x80000418 = 0x00000006
armsd:
armsd:
armsd:
armsd: load hello
armsd: go
Hello
Program terminated normally at PC = 0x0000a2d8 (__sys_
exit + 0xc)
+000c 0x0000a2d8: 0xef123456 ..4V : swi 0x123
456
armsd: load sortint
armsd: break @InstallHandler
armsd: go
Hello world!
Breakpoint #1 at PC = 0x0000815c (InstallHandler + 0)
InstallHandler
+0000 0x0000815c: 0xe0400001 .&.. : * sub r0,r0
,r1
armsd: step 50
-
    
```



```

Terminal
Window Edit Options Help
Writing 8 data elements @ address: 0x5200...
-----Reading the same after writing-----
Writing 8 data elements @ address: 0x5800...
-----Reading the same after writing-----
Writing 8 data elements @ address: 0x5e00...
-----Reading the same after writing-----
Writing 8 data elements @ address: 0x6400...
-----Reading the same after writing-----
Writing 8 data elements @ address: 0x6a00...
-----Reading the same after writing-----
Writing 8 data elements @ address: 0x7000...
-----Reading the same after writing-----
Mpeg stream input file end
Mpeg stream input master: raises interrupt
Mpeg output Master: interrupt is recognized
Reading stream from Memory
-
    
```

Conclusions

Highlights

- Understand System Integration issues with respect to SystemC Models
- Issues regarding adapting external models to SystemC
- Gained insight on Performance Analysis using SystemC modeling
- Application could run on such a system

Lowlights

- Yet in proof of concept stage; some issues still open
- Overall simulation speed was 4K cycles/second; total simulation time was 6 to 8 minutes
- Control on simulation available only through the debugger interface attached to the ARMulator

This presentation contains information confidential to Royal Philips Electronics. Contents of the presentation (in full or part), cannot be disclosed, disseminated, distributed or copied without express written permission from Royal Philips Electronics

