



Introduction to SystemC AMS - TLM interaction for SoC Architectures

Contribution to OSCI Special Session @DATE2008

"Where Two Worlds Meet: SystemC AMS and TLM Interaction"

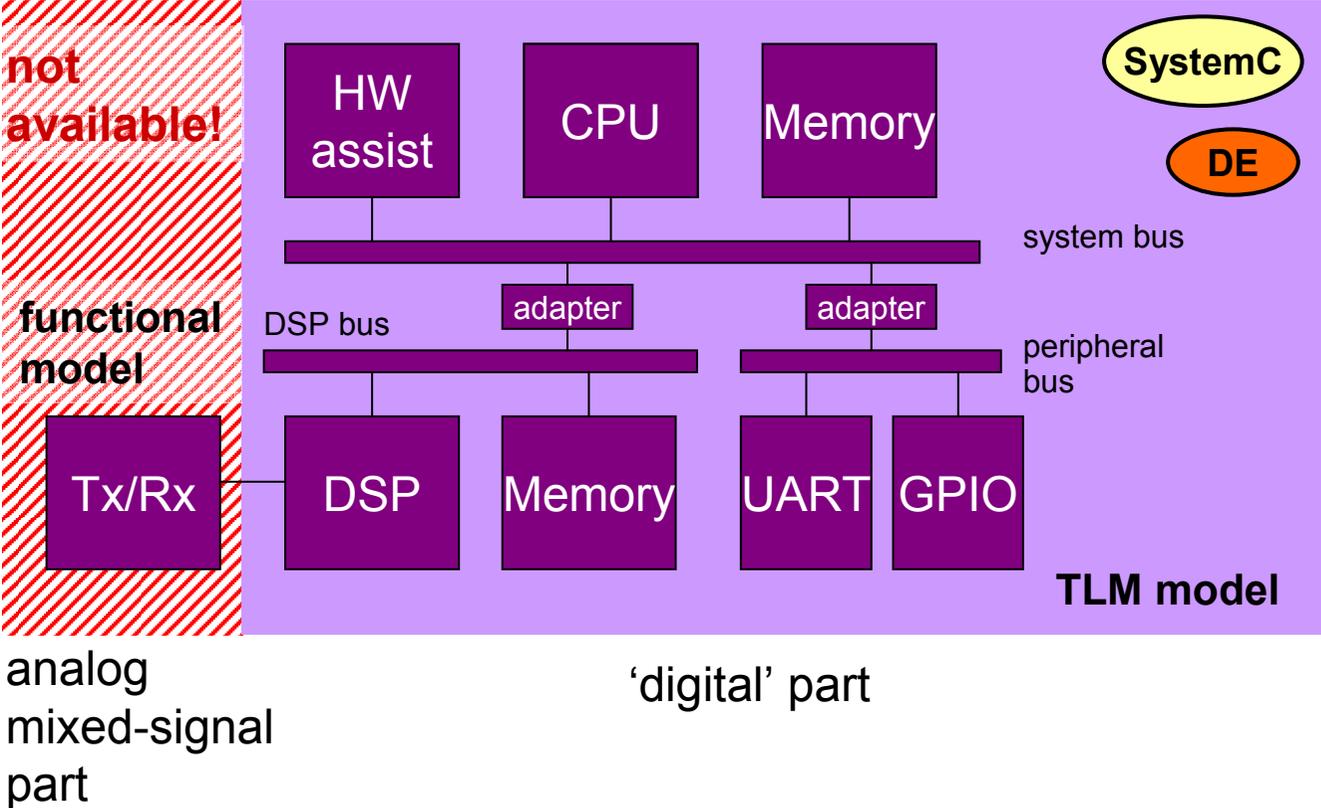
March 2008



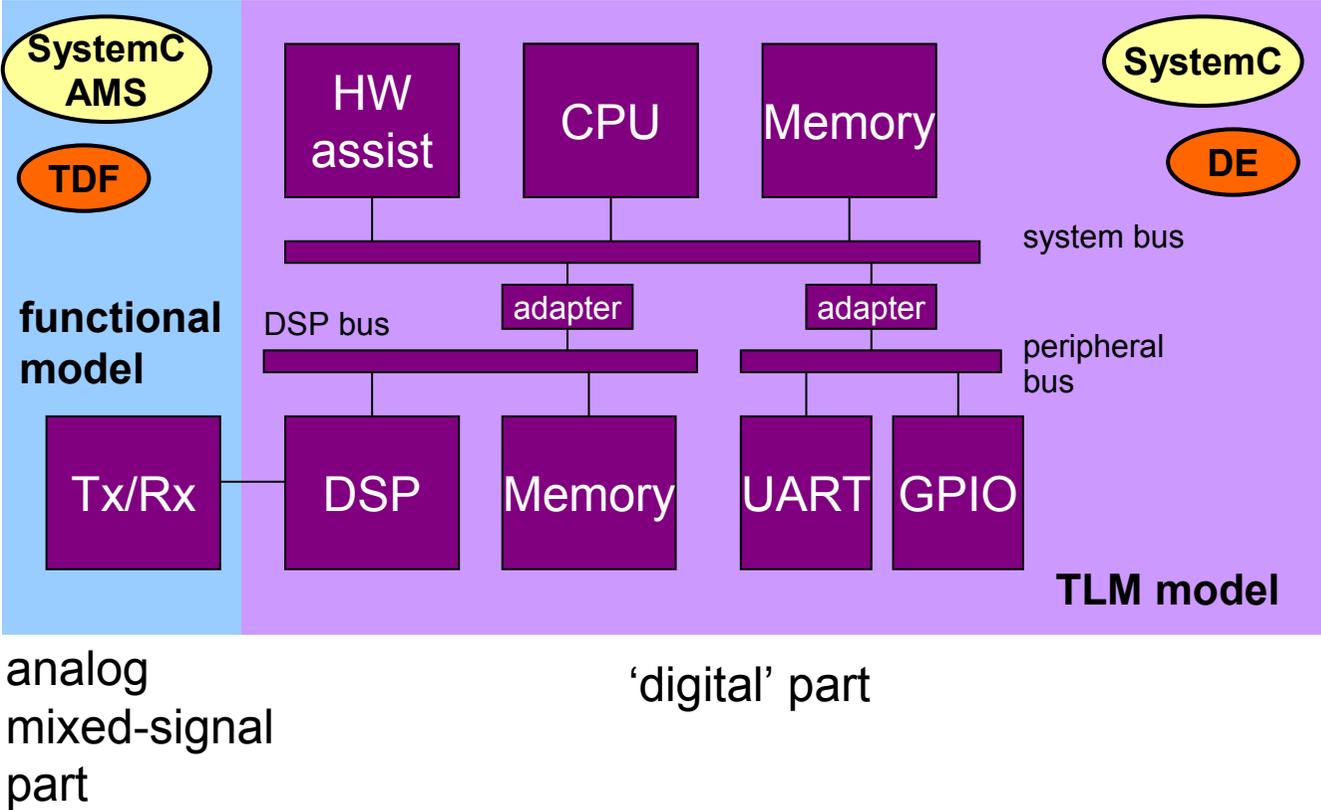
Requirements

- ▶ Compatibility and Interoperability with digital functional models and subsystems
- ▶ Interface and communication: SystemC TLM (2.0) compliant
 - configuration should support initiator and target configuration
 - support of polling or interrupt based synchronization
- ▶ Register interface based on SCML memory model
 - register interface splits communication in architectural context from functional modeling
 - note: register communication itself not part of TLM standard
- ▶ Wrap SystemC AMS subsystem into ‘TLM shell’
 - AMS subsystem represented as functional model using Timed Data Flow (TDF) semantic
 - embed functional model in architecture block
 - two options
 - option 1: SystemC FIFO interface
 - option 2: communicate directly to SystemC AMS extension using converter channels

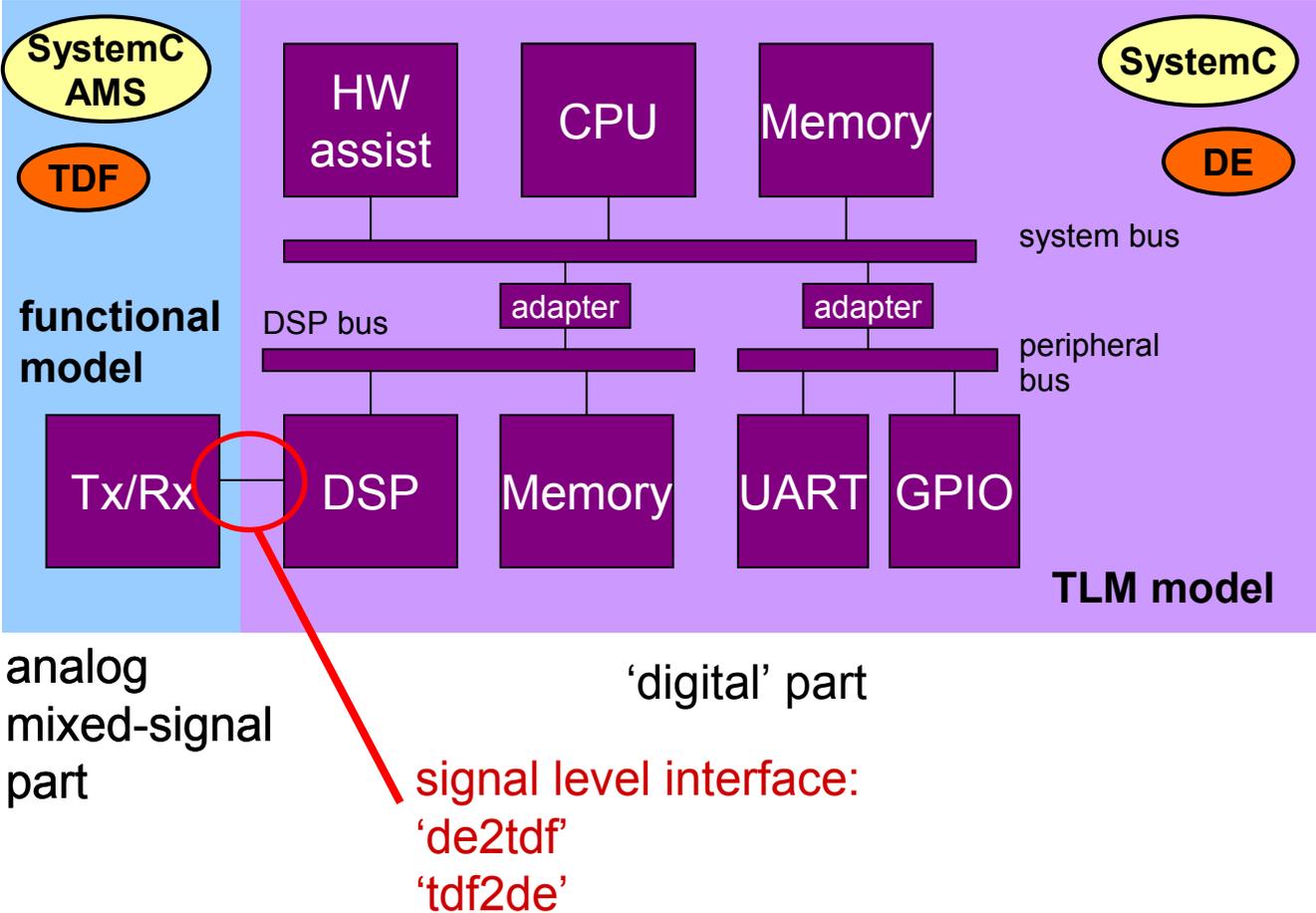
SoC Architecture high-level view



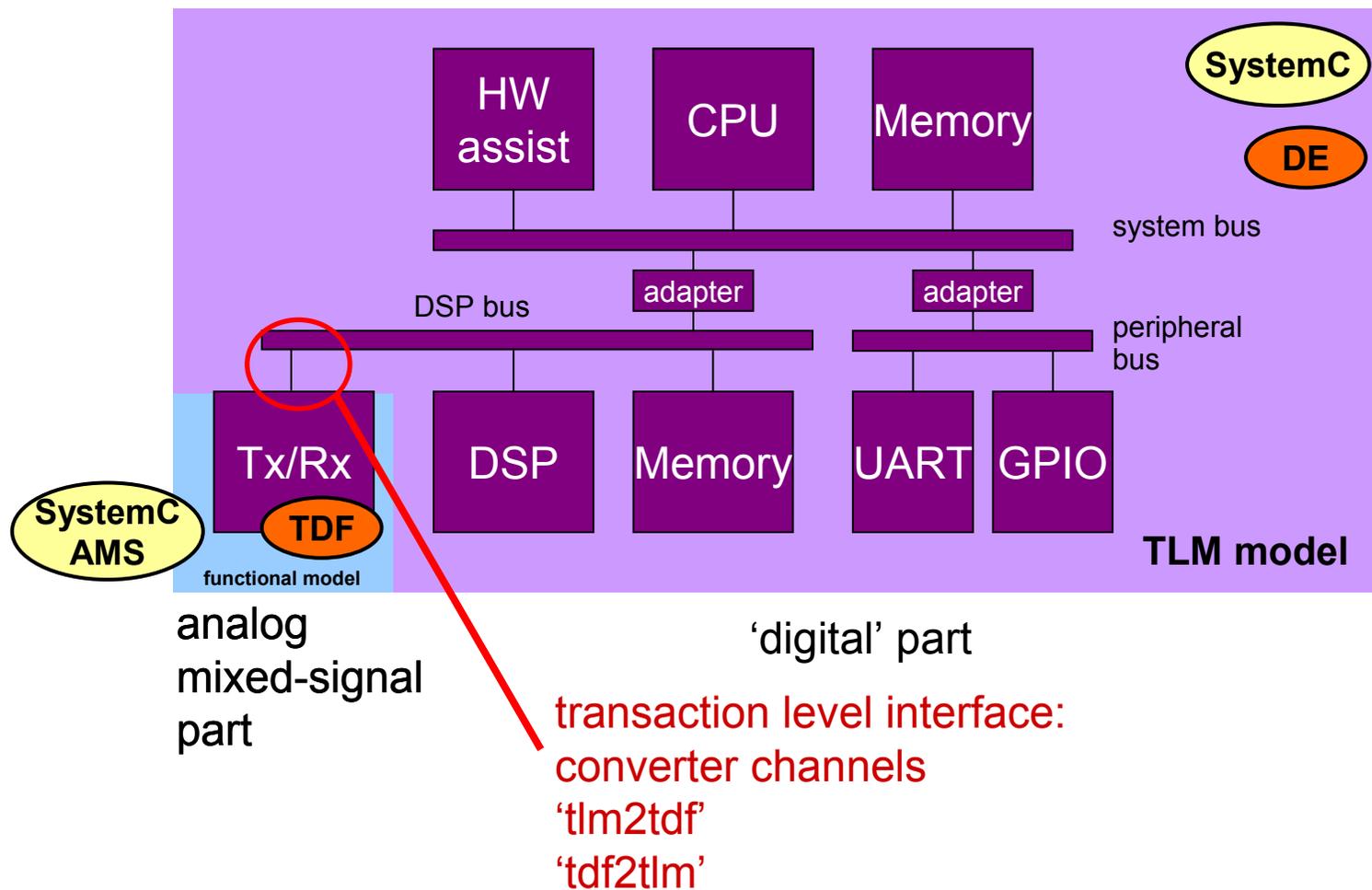
SoC Architecture high-level view



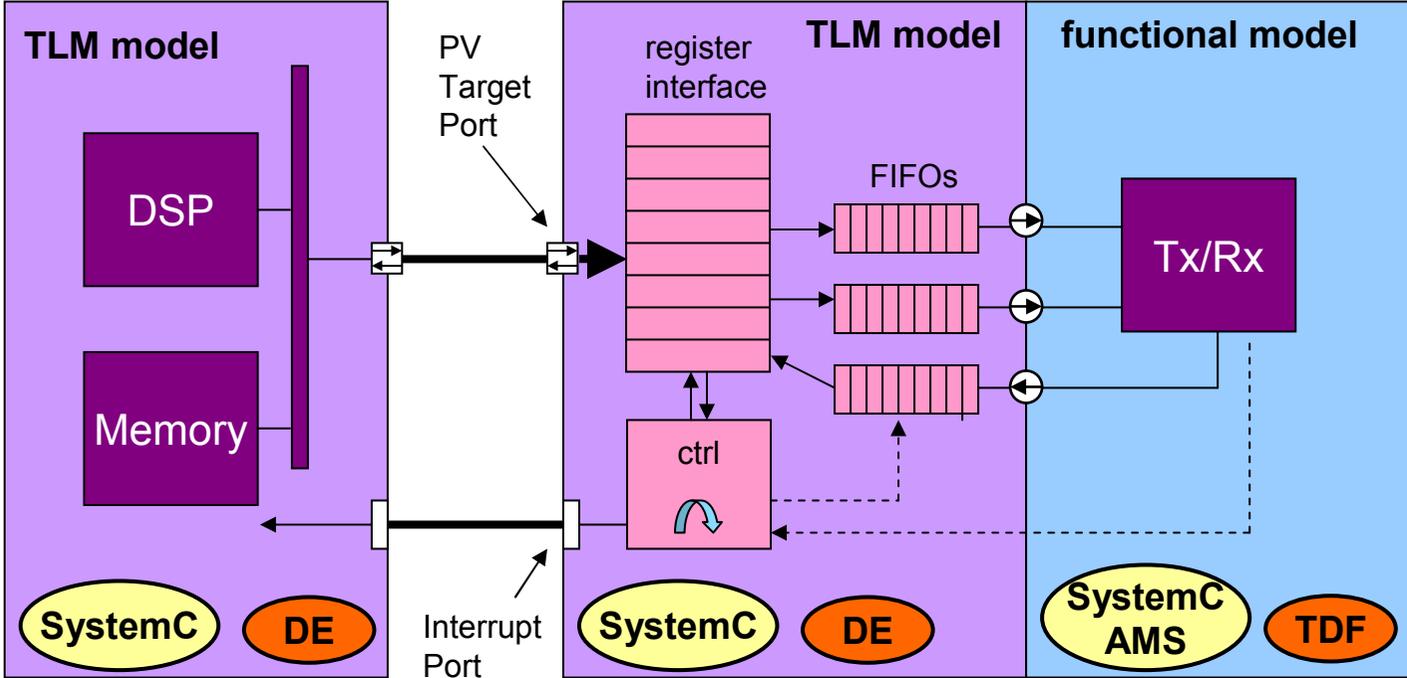
SoC Architecture high-level view



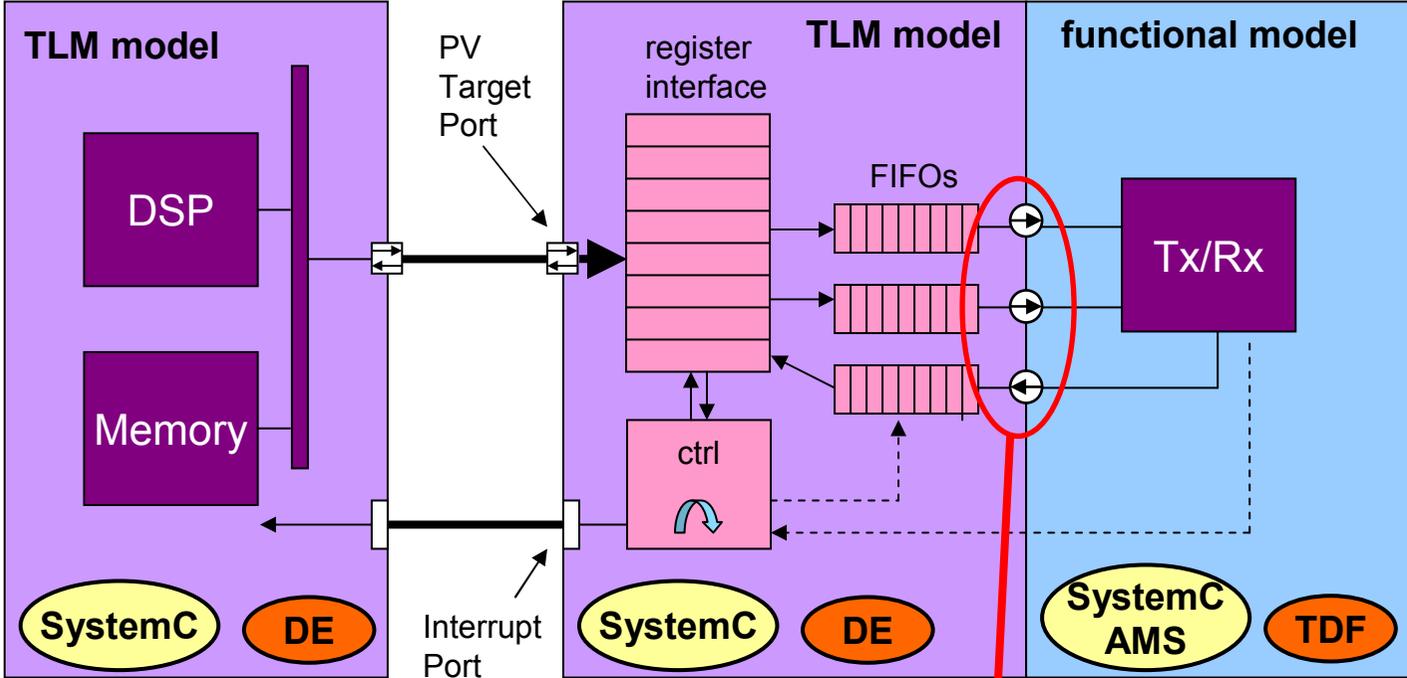
SoC Architecture high-level view



SoC Architecture interface - option 1

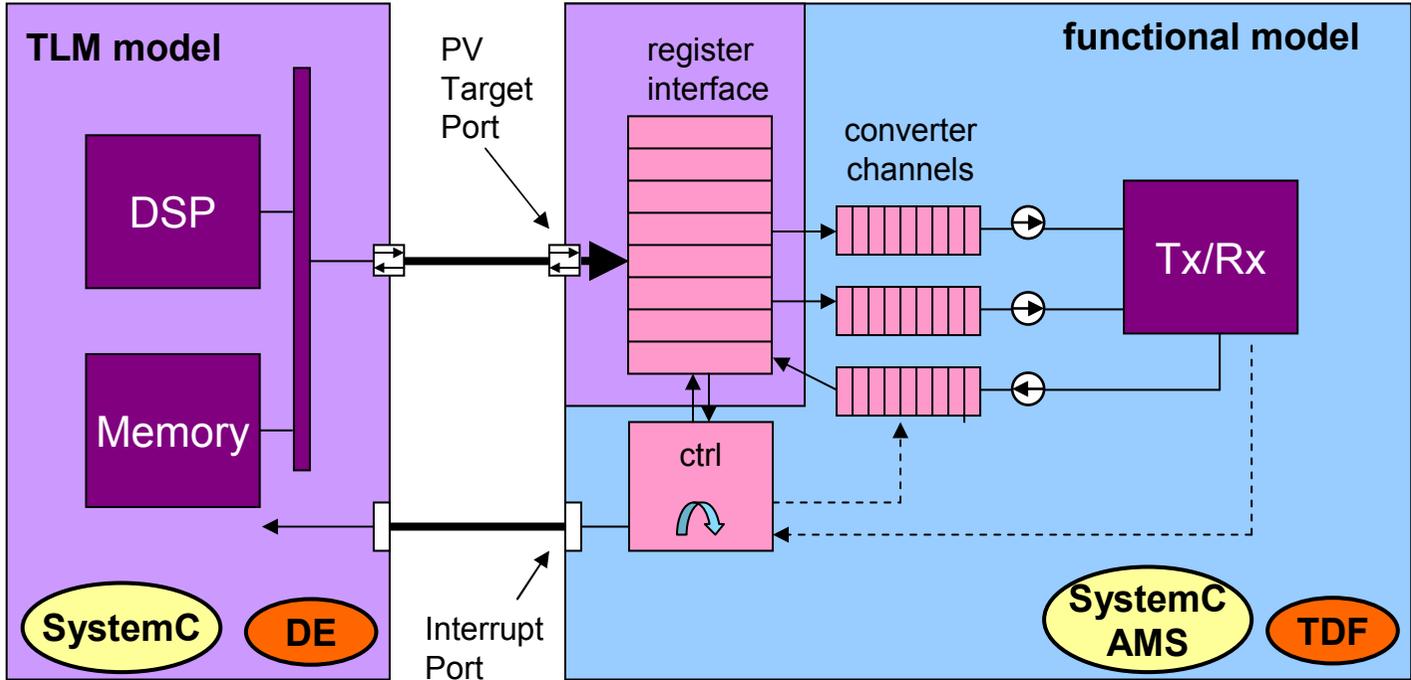


SoC Architecture interface - option 1

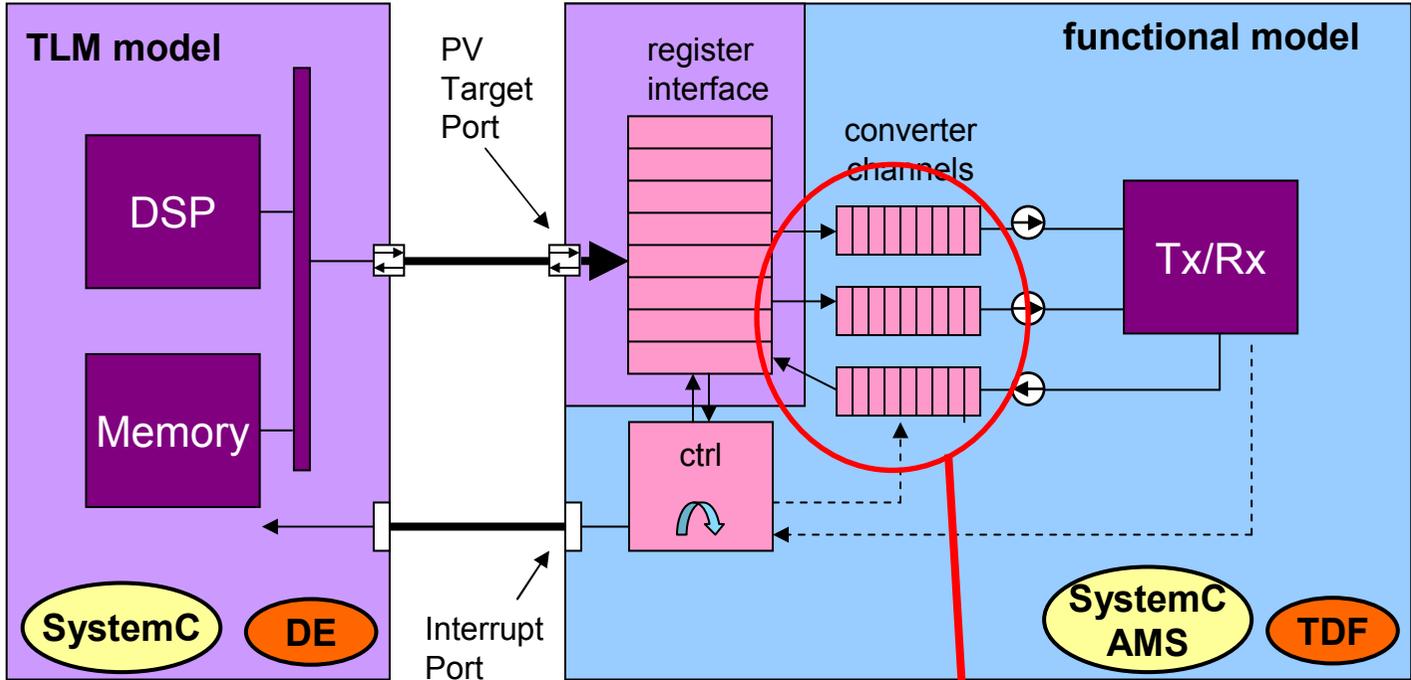


signal level interface
'de2tdf'
'tdf2de'

SoC Architecture interface - option 2



SoC Architecture interface - option 2



transaction level interface:
converter channels
'tlm2tdf'
'tdf2tlm'

Conclusions

- ▶ Compatibility and interoperability
 - using communication, protocol and payload as proposed by TLM WG
 - using SCML register interface similar to other sub-systems
- ▶ Embedding/wrapping of functional models
 - valid for AMS and digital subsystems
 - two approaches possible
 1. Based on standard SystemC FIFO and communication
 2. New register communication in SystemC AMS extensions using converter channels
- ▶ Evaluation of pros and cons for both options
 - Impact on standardization for both AMS and TLM
 - future AMS-TLM (in)dependency
 - Impact on simulation performance, flexibility, compatibility and interoperability

