# Notes Embedded System Design

**RTL** Register-Transfer Level. This is mainly used about RTL library, which is a collection of low-level components and their mapping to FPGA..RTL is described in e.g. HDL. A combination of RTLs may be translated into a CAM.

**MoC** Model of Computation. Highest level model. Often written in SystemC or equivalent.

**SM** Specification Model. Top level model. May be written as a MoC.

**TLM** Transaction Level Model. Used by System designers.

**FSM** Finite State Machine.

**FSMD** Finite State Machine with Data. Same as FSM, except it is always possible to exchange data between states.

**BB** Basic Block. Collection of expressions with no if-tests. A basic block can effectively be cached as it is always run from start to finish and all code in the BB is always run. See CDFG.

**CDFG** Control-Data Flow Graph. Collection of BB and if-diamonds – can represent any programming language code.

**PE** Processing Element. A processing element is a component the computes some specific functions. System modelling is collecting PEs and CEs in a system. A PE is built up from putting RTLs together and can then be realized in SW/HW.

**CE** Communication Element.

**SLDL** System-Level Design Language. E.g. SystemC

**IP** Intellectual Property. E.g. an RTL description of functionality, i.e. an implementation of functionality in HW.

**ISS** Instruction Set Simulator. Used to simulate a processor at the instruction level.

**CAM** Cycle-Accurate Model. Used by HW designers. Lowest level model. May be simulated or realised directly with full accuracy.

**BCAM** Bus Cycle-Accurate Model. Half of CAM where focus is only on communication (CE).

**ASIC** Application-specific Integrated Circuit. Micro-controller designed for something specific, e.g. low power consumption, cellular phones, etc. – as oppose to general purpose CPU’s or FPGA’s

**LUT** Look-up Table

**FPGA** Field-programmable Gate Array

**MPSoC** Many-Processor System on a Single Chip.

**SoPC** System-on-a-Programmable-Chip.

**NOC** Network-on-a-Chip.

**SER** Specify-Explore-Refine. Principle of present system design technique. You specify a model from the information at hand, you explore the model to determine its accuracy and alternatives and then you refine it by building new model with more detail where needed. These new models also use the SER approach.

**PCAM** Pin/Cycle Accurate Model.

**CCAM** Computation Cycle-Accurate Model. Half of CAM where focus is only on computation (PE).

**HDL** Hardware Description Language. Used to express components at the RTL.

**VHDL** Very high speed integrated circuit Hardware Description Language

**HLS** High-level Synthesis.

**IS** Instruction Set. The assembler instruction understood by a given processor.

**IF** Interface components. Bus controller (Arbiter)

**PSM** Process State Machine. Collection of states or FSMs running in parallel.

**CAD** Computer-aided Design

**Timed TLM** Same as TLM, but with timing information on the communication. Is there not always that, in Chapter 3 they say there is.

* System
  + Behaviour (Function) -> Structure (Netlist)
    - rr
  + Structure (Netlist) -> Physical (Layout)
    - rr
* Processor
  + Behaviour (Function) -> Structure (Netlist)
    - R
  + Structure (Netlist) -> Physical (Layout)
* Logic
  + Behaviour (Function) -> Structure (Netlist)
    - R
  + Structure (Netlist) -> Physical (Layout)
* Circuit
  + Behaviour (Function) -> Structure (Netlist)
    - R
  + Structure (Netlist) -> Physical (Layout)
* System Level Modelling
  + System Model (SM)
  + Model of Computation (MoC). Highest level mode expressed in e.g. SystemC.
  + Process State Machine (PSM)
  + Written by application designers
  + Transaction Level Model (TLM)
    - Written by system designers.
* Processor Modelling
  + FSMD,
* Processor Level Synthesis = High Level Synthesis.
* Register-Transfer Level (RTL)
  + Written by HW designers.
* Cycle-Accurate Model (CAM)

Flow:

MoC -> TLM - > TTLM-> BCAM -> CAM

Top Down:

1. We create a purely functional model (behaviour only) – System Model. This may be done as MoC written in SystemC. Possibly combined with a PSM.
2. Then we refine this model as a TLM using