HELP: High-Level Models For Low-Power HW/SW Systems

The HELP people, among which, for today's talk:
Tayeb Bouhadiba, Florence Maraninchi
VERIMAG/Synchrone
http://www-verimag.imag.fr/~maraninx/

Verimag

SYNCHRON'11 Le Bois du Lys

1 Context

2 The HELP Generic Modeling Principles

3 A Question Related to the Discrete/Continuous Parts

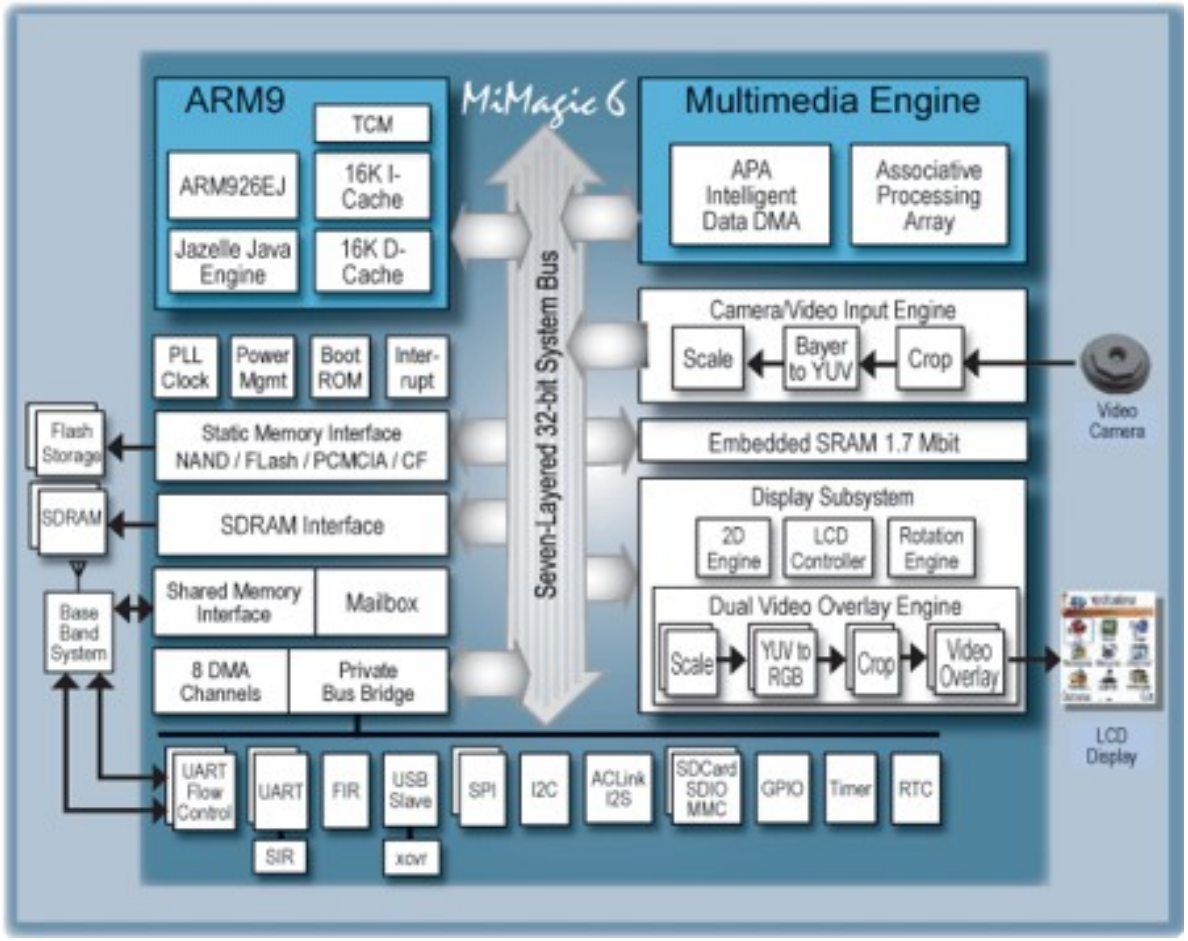
4 Ongoing Work

Context: The HELP Project 2009-2012

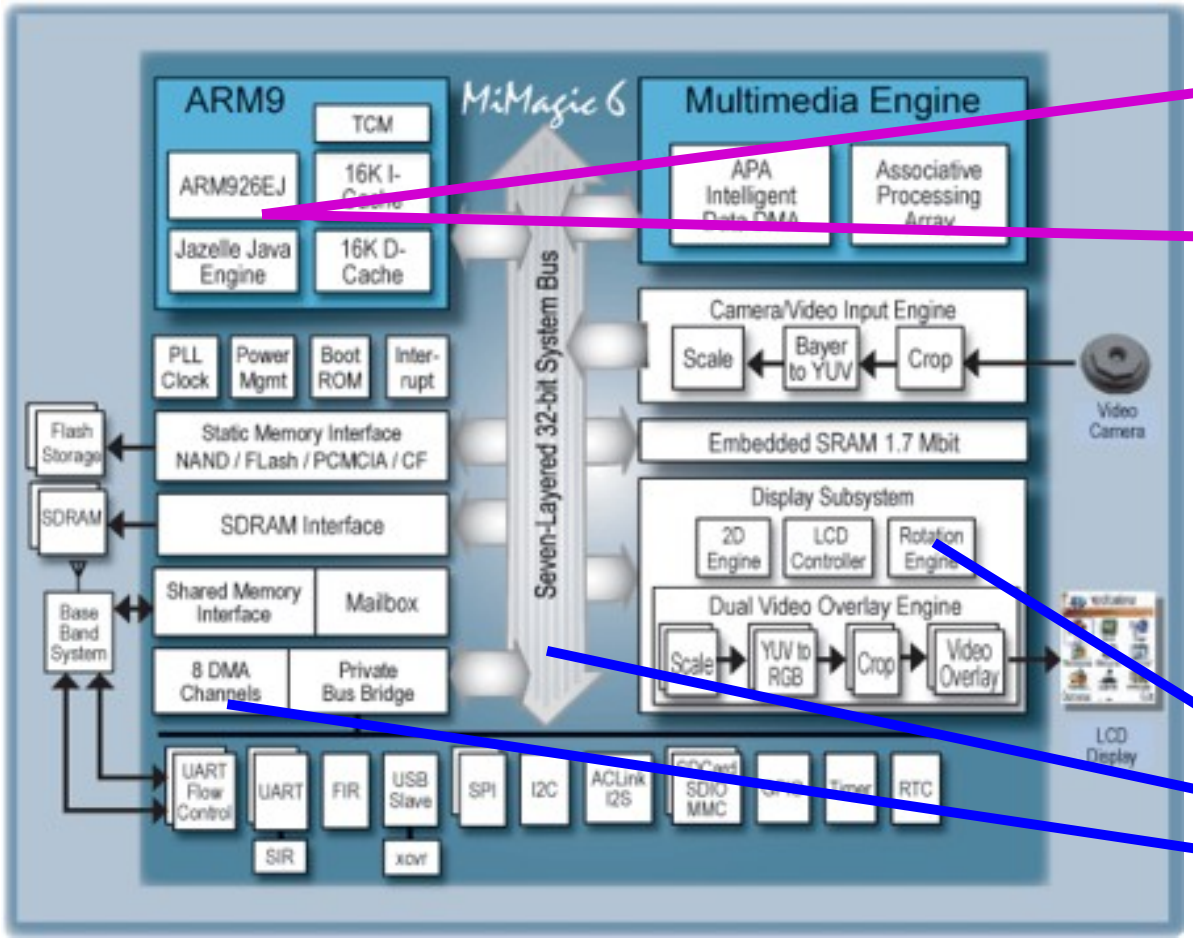
- Verimag/Synchrone (Florence, Matthieu Moy, Tayeb Bouhadiba, Claire Maiza, Catherine Parent)
- INRIA AOSTE (Robert, ...)
- LEAT Nice
- STMicroelectronics Grenoble
- DOCEA Power Grenoble

Provide early evaluations of energy consumption for systems-on-a-chip, at the transactional level (TLM).

Systems on a Chip



Systems on a Chip



On a processor:

A real–time OS + several processes

Hardware IP's (components)

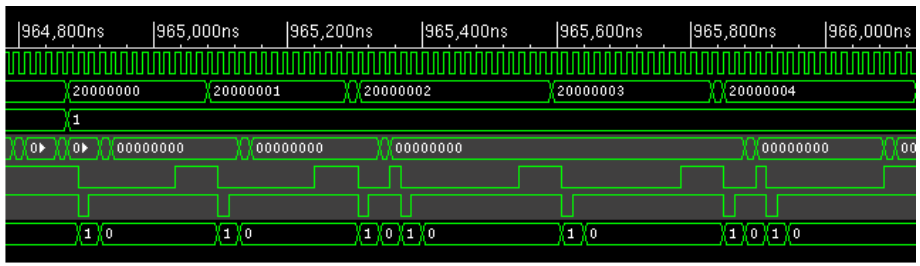
Transaction-Level Modeling

RTL

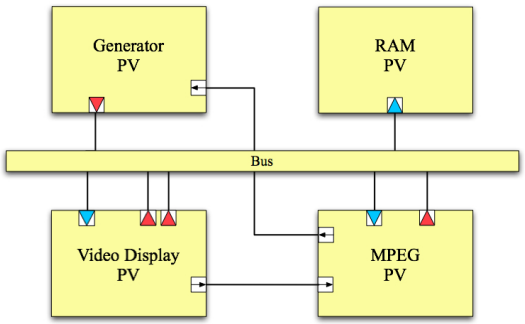
Synthesizable+

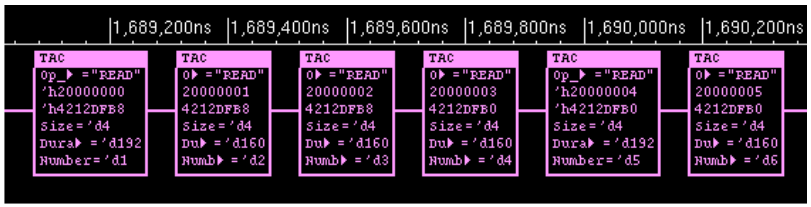
Cycle and Data Accurate+

Slow Simulations–



Transaction-Level Modeling





TLM

\+ Early Available

\+ Fast Simulations

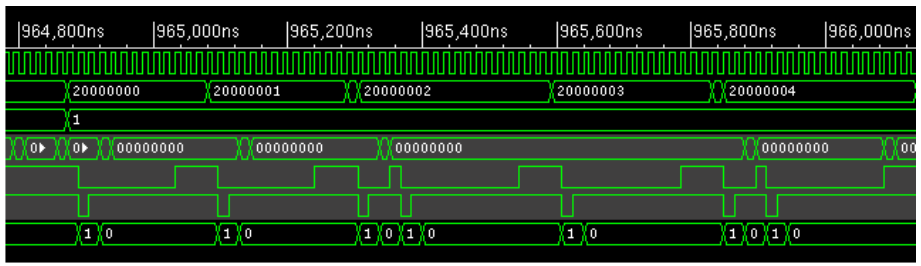
– Not synthesizable

RTL

Synthesizable+

Cycle and Data Accurate+

Slow Simulations–



Sources of Energy Consumption and Available Mechanisms for Reducing It

Sources:

- Static, leakage currents (indep. of the behavior)
- Dynamic, when transistors commute (related to the behavior)

Solutions for reducing energy consumption:

Sources of Energy Consumption and Available Mechanisms for Reducing It

Sources:

- Static, leakage currents (indep. of the behavior)
- Dynamic, when transistors commute (related to the behavior)

Solutions for reducing energy consumption:

Do not build systems-on-a-chip!

Sources of Energy Consumption and Available Mechanisms for Reducing It

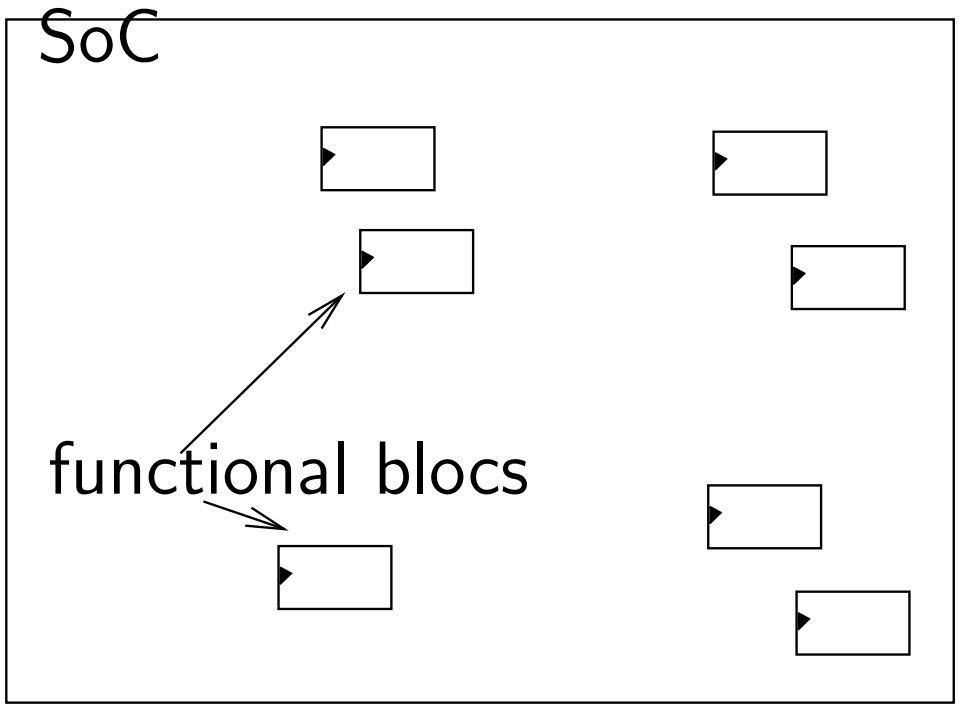
Sources:

- Static, leakage currents (indep. of the behavior)
- Dynamic, when transistors commute (related to the behavior)

Solutions for reducing energy consumption:

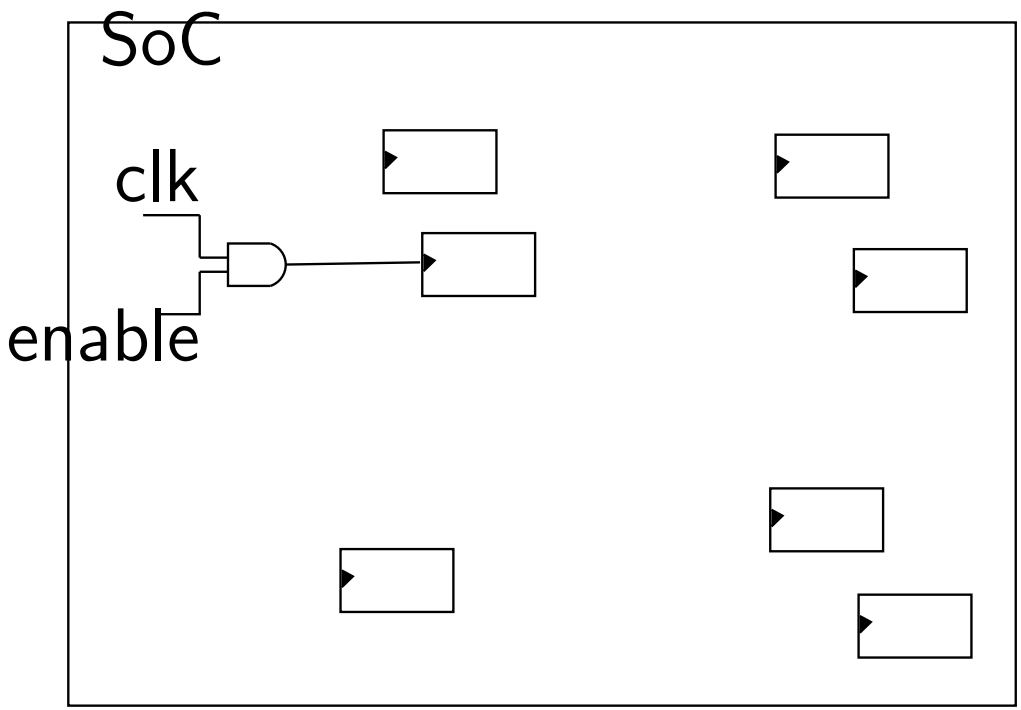
- Clock gating
- Power gating
- Dynamic Voltage and Frequency Scaling (DVFS)

Low Power SoC Design Solutions



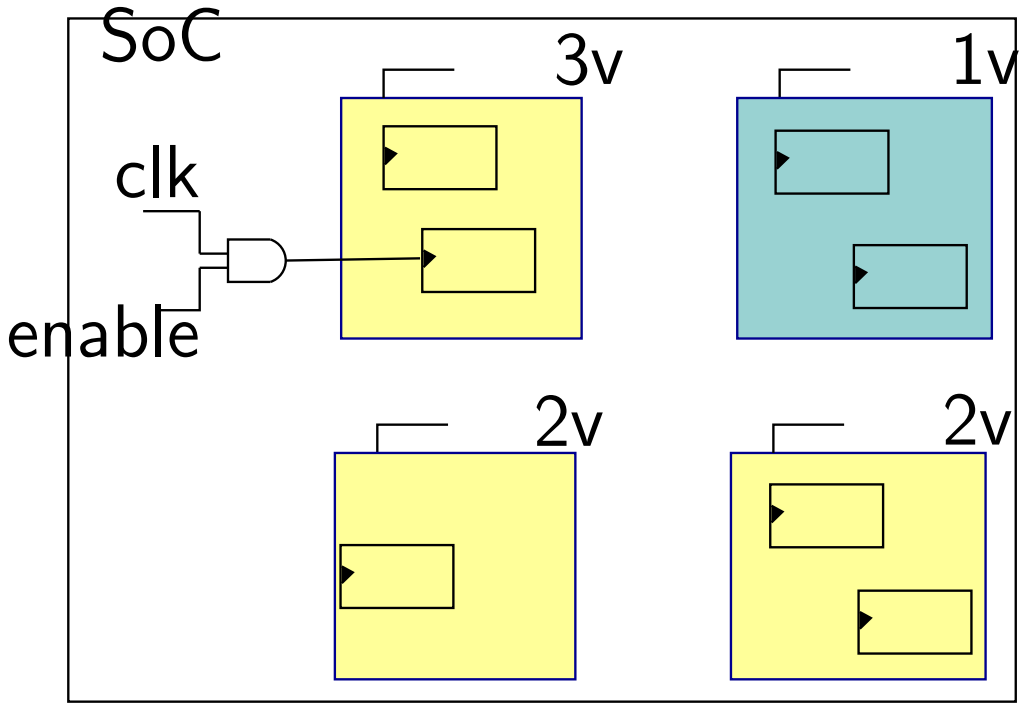
Low Power SoC Design Solutions

- Clock Gating
  - Save Dynamic Power



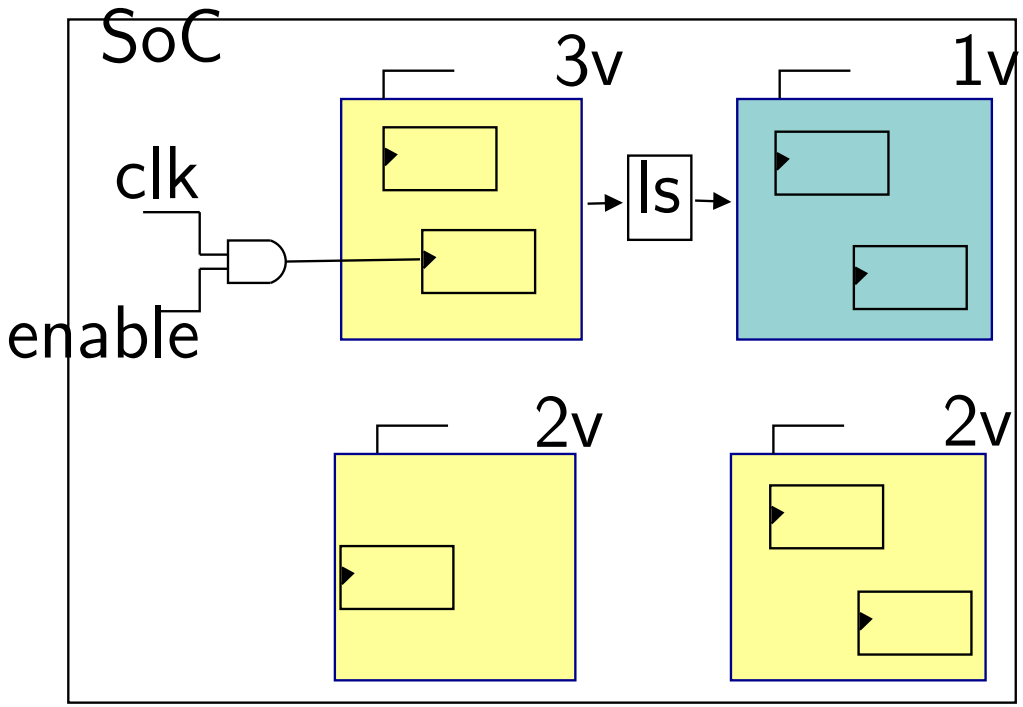
Low Power SoC Design Solutions

- Clock Gating
  - Save Dynamic Power
- Multi-VDD



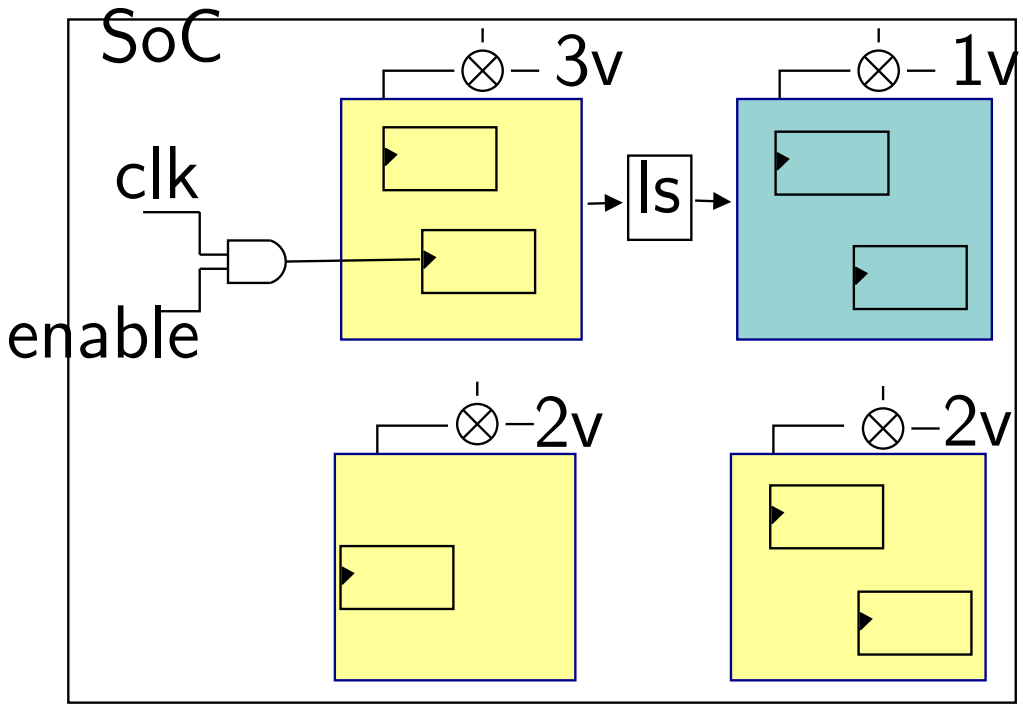
Low Power SoC Design Solutions

- Clock Gating
  - Save Dynamic Power
- Multi-VDD
  - Needs Level Shifters



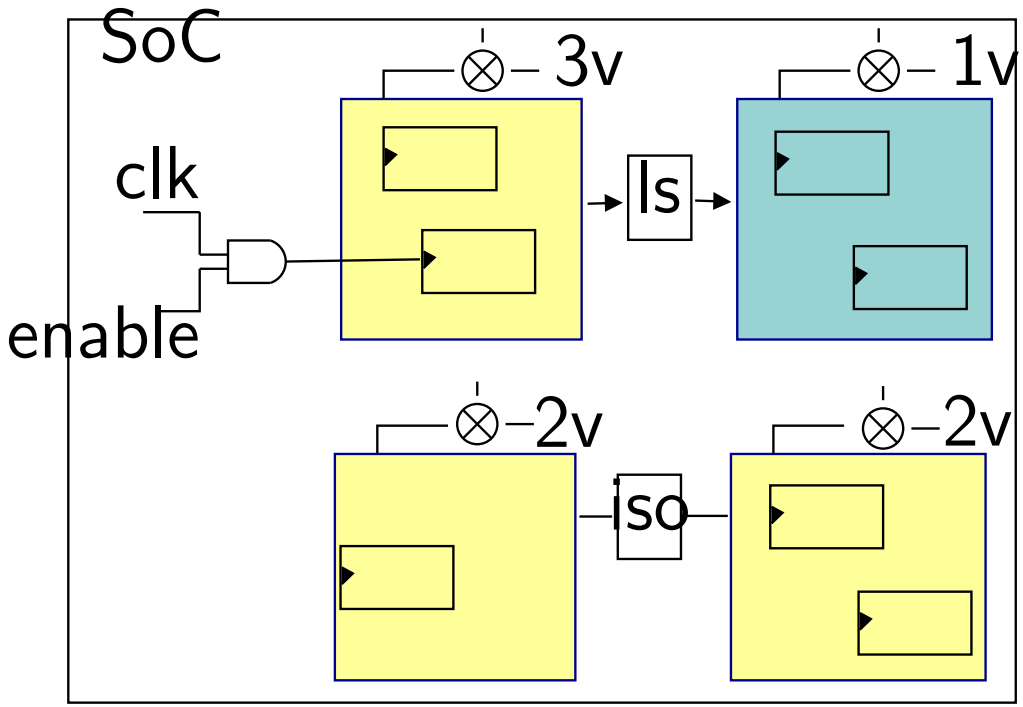
Low Power SoC Design Solutions

- Clock Gating
  - Save Dynamic Power
- Multi-VDD
  - Needs Level Shifters
- Power Gating
  - Save static power



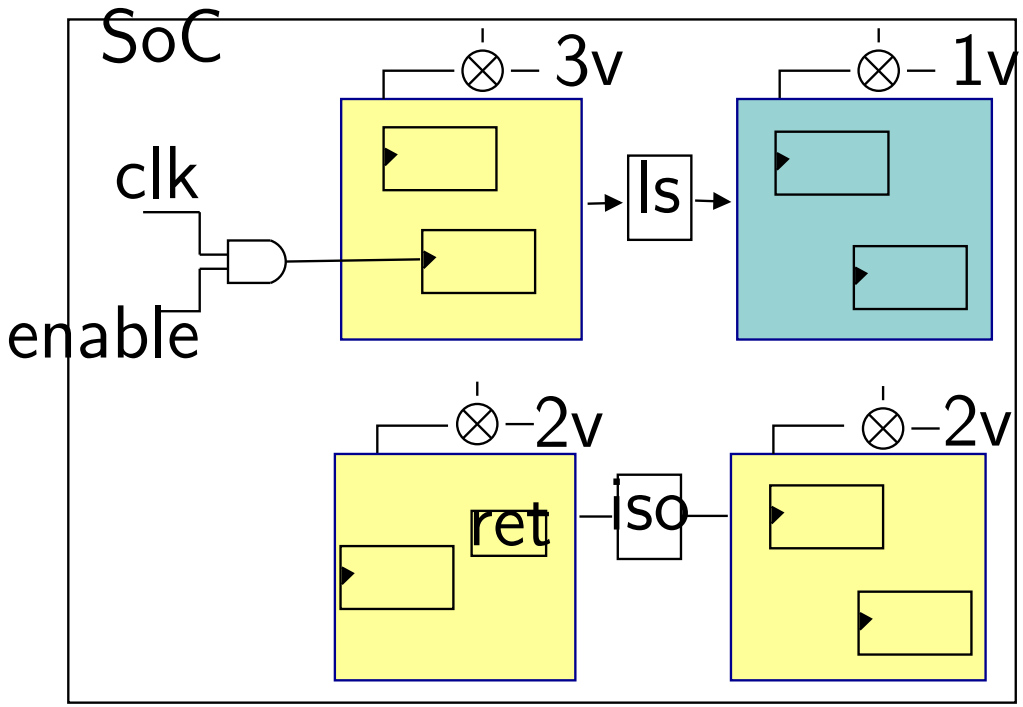
Low Power SoC Design Solutions

- Clock Gating
  - Save Dynamic Power
- Multi-VDD
  - Needs Level Shifters
- Power Gating
  - Save static power
  - Needs isolation



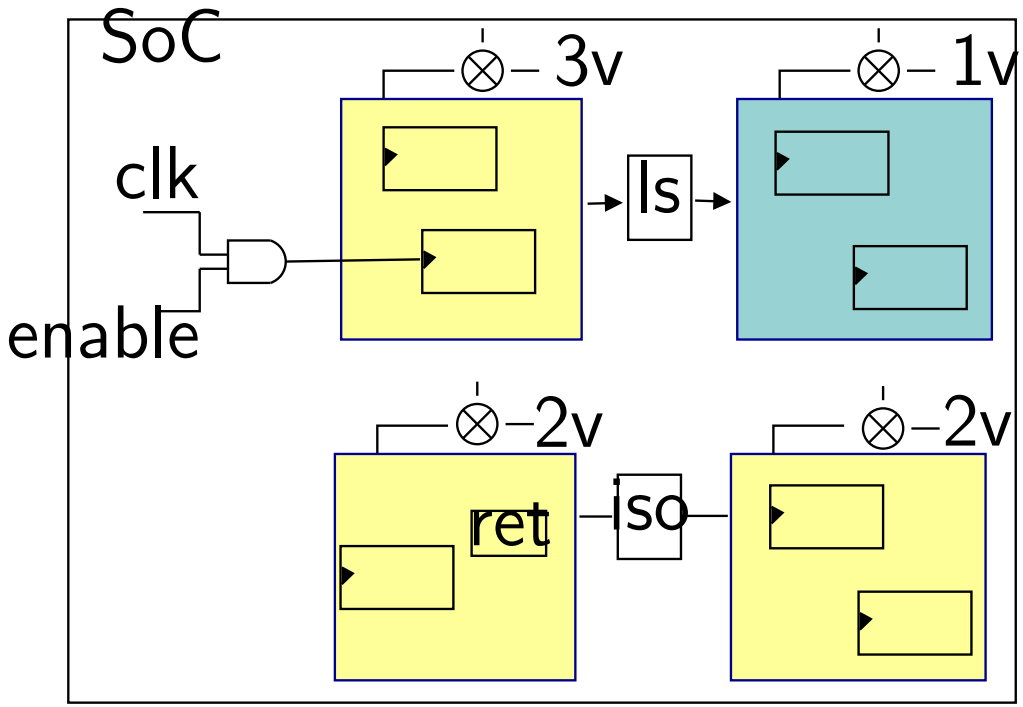
Low Power SoC Design Solutions

- Clock Gating
  - Save Dynamic Power
- Multi-VDD
  - Needs Level Shifters
- Power Gating
  - Save static power
  - Needs isolation
  - (May) needs retention



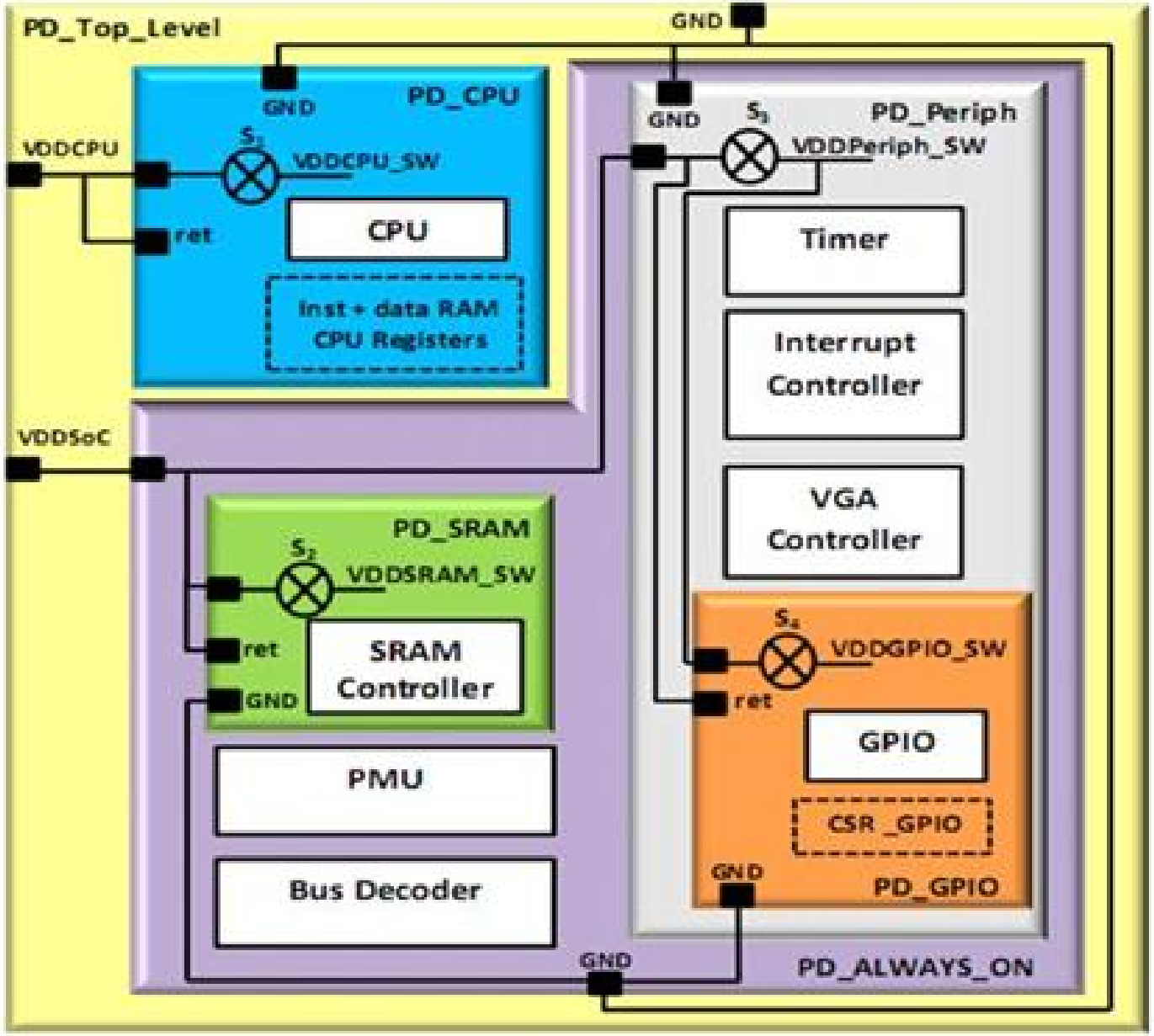
Low Power SoC Design Solutions

- Clock Gating
  - Save Dynamic Power
- Multi-VDD
  - Needs Level Shifters
- Power Gating
  - Save static power
  - Needs isolation
  - (May) needs retention



Most of these are defined in the Unified Power Format (UPF)

Power Domains (picture by Ons Mbarek, LEAT)



Power Management Policies

A power management policy = a strategy for reducing energy, playing with clock/power gating and DVFS, based on observations: temperature sensor, "functional sensors" (contents of a FIFO), ...

Implementations: HW/SW, Centralized/distributed, ...

Complex feedback effect, need for virtual protoyping (= playing the SW on a model of the HW + Φ).

Existing Evaluation Tools

- At the gate or register-transfer (RT) levels: industrial tools, precise but very slow
- At the transaction level: several proposals for specific components (bus, network on chip, processors, ...) but nothing for a system-wide model integrating the SW

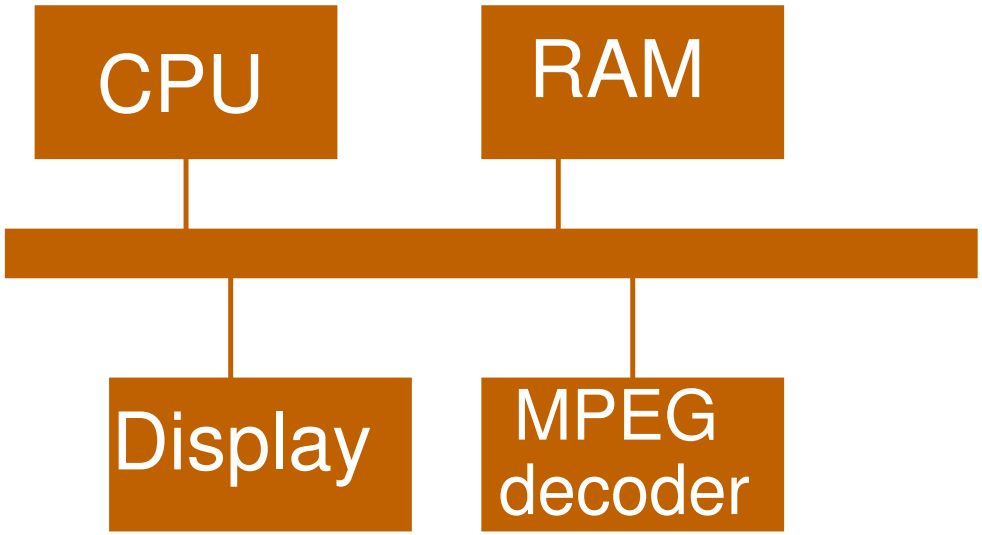
1 Context

2 The HELP Generic Modeling Principles

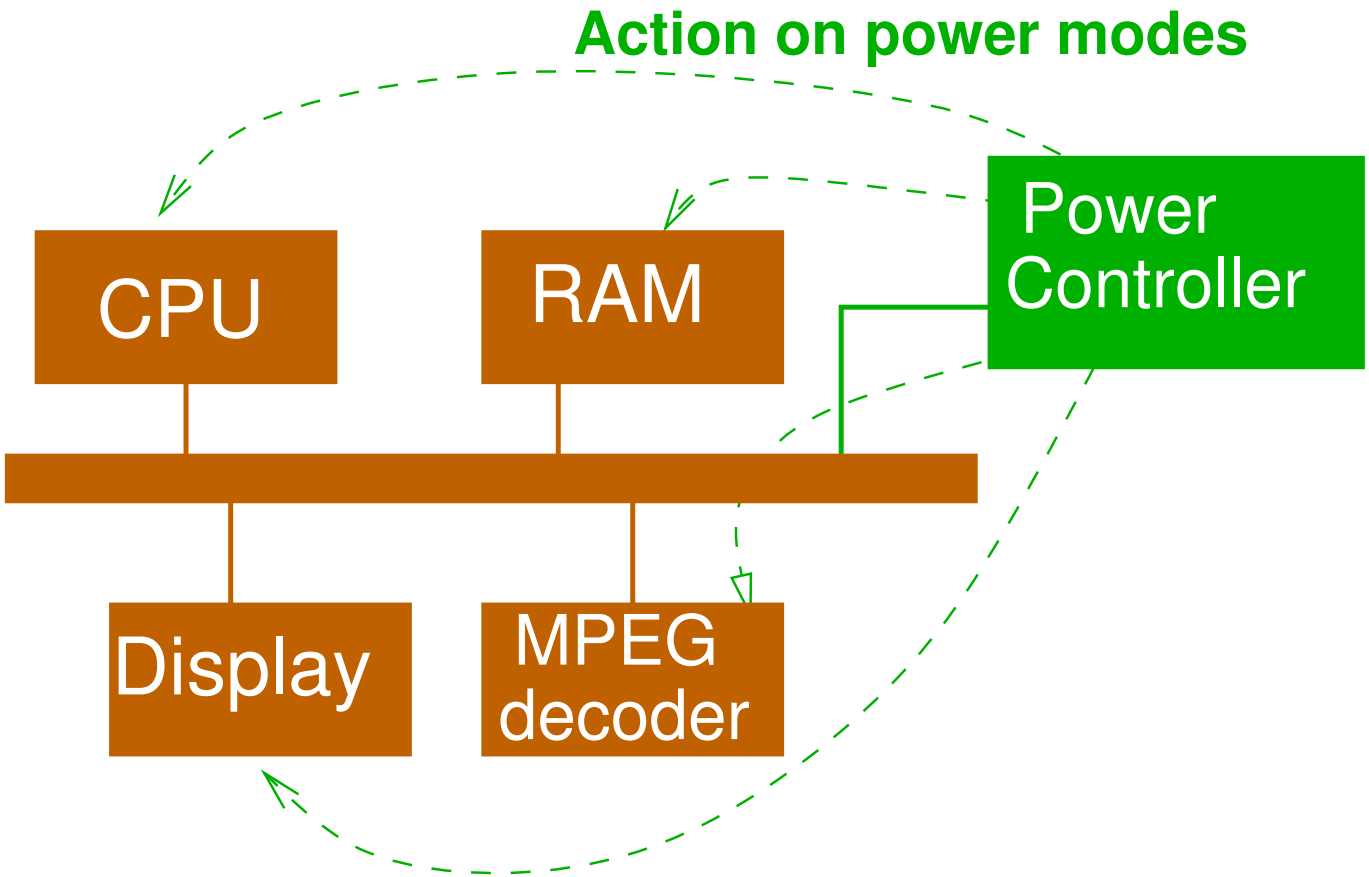
3 A Question Related to the Discrete/Continuous Parts

4 Ongoing Work

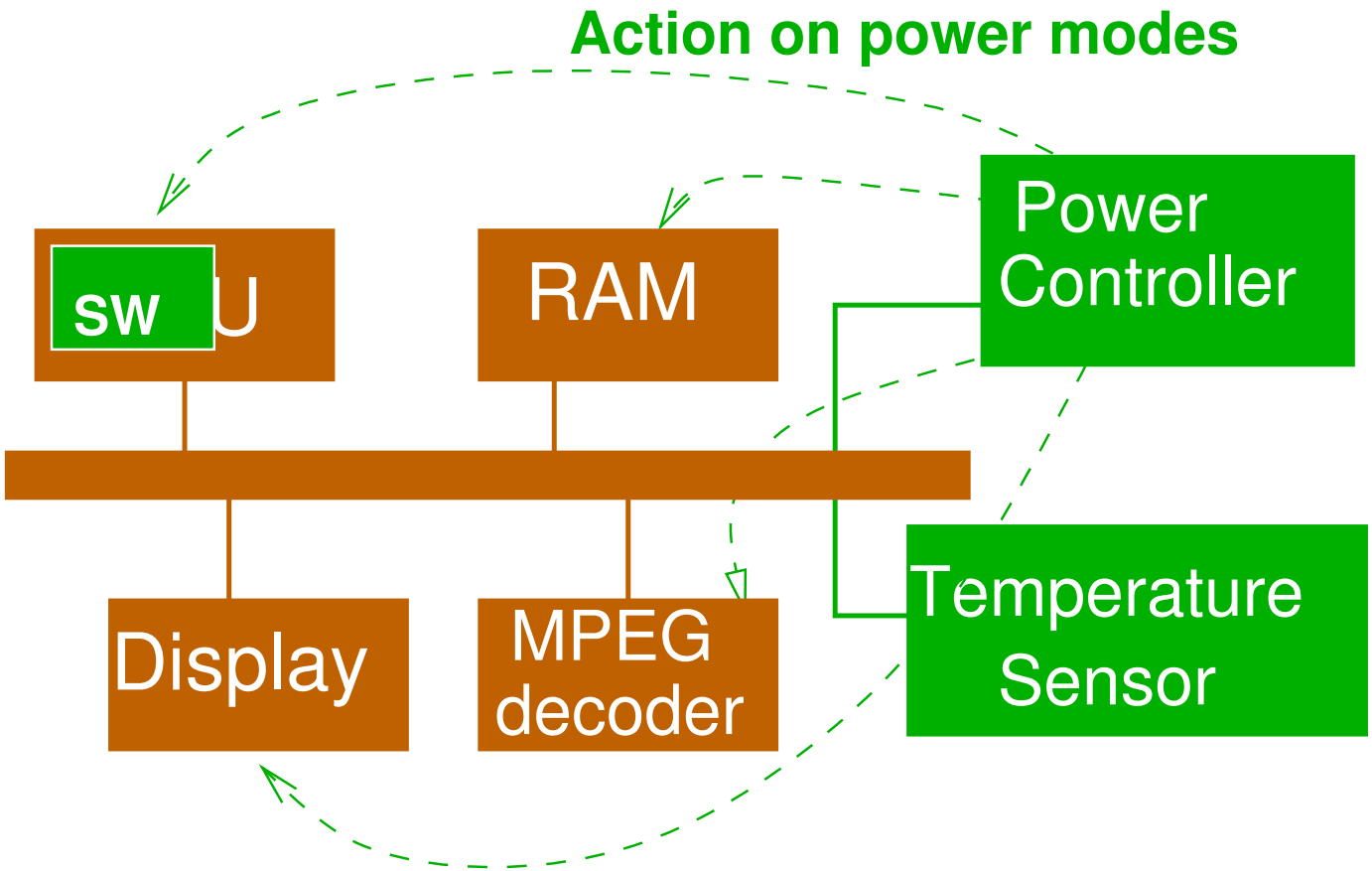
Example System



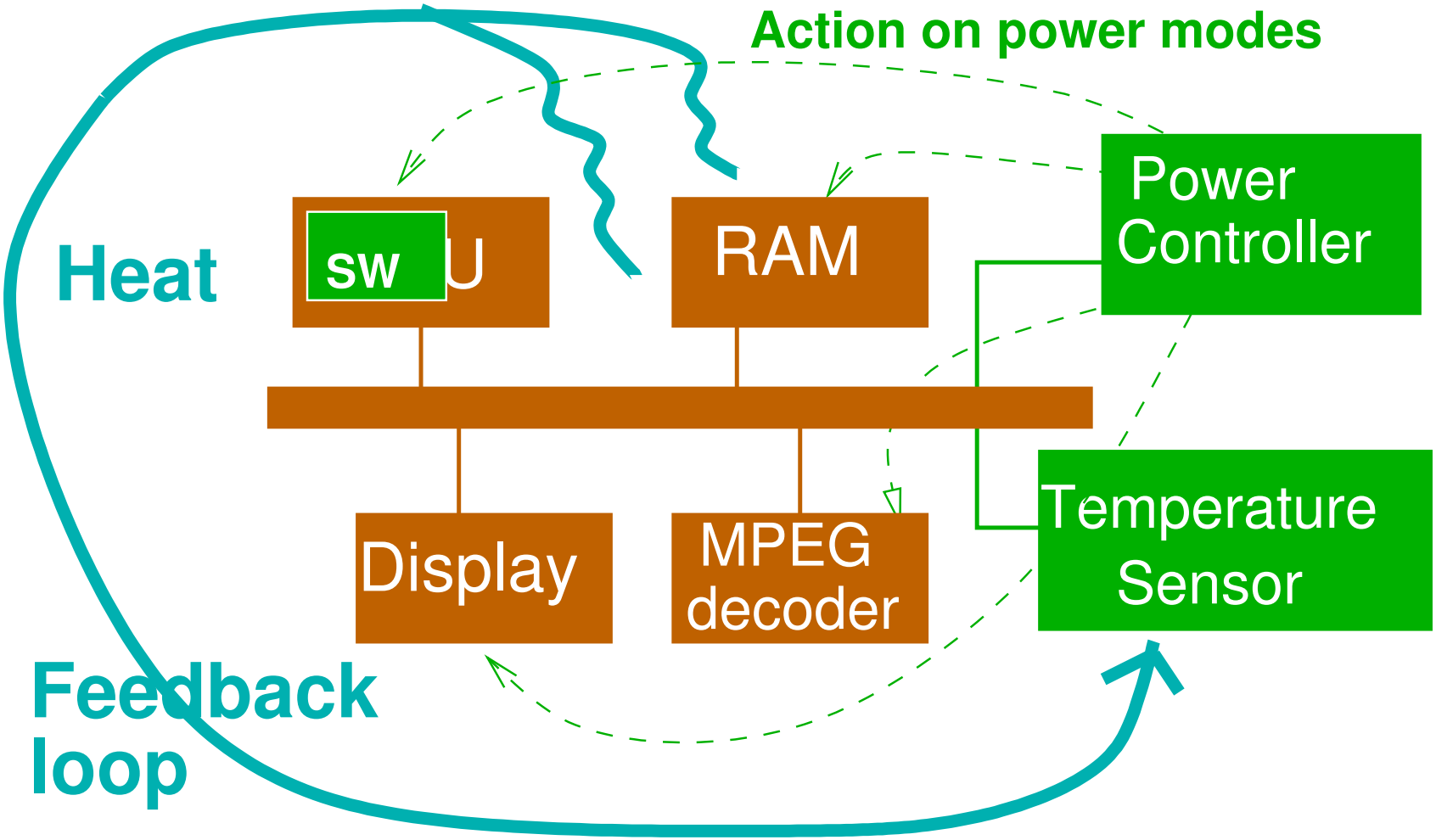
Example System



Example System



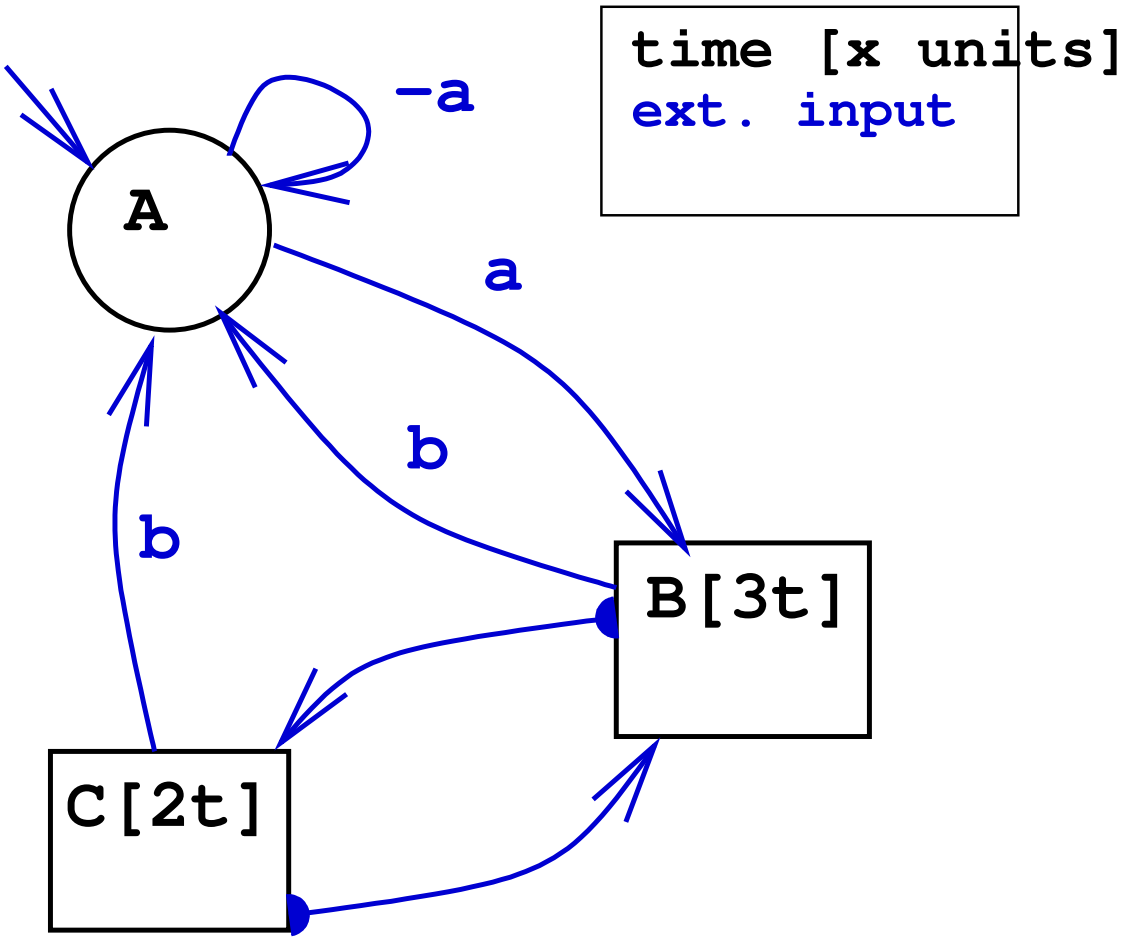
Example System



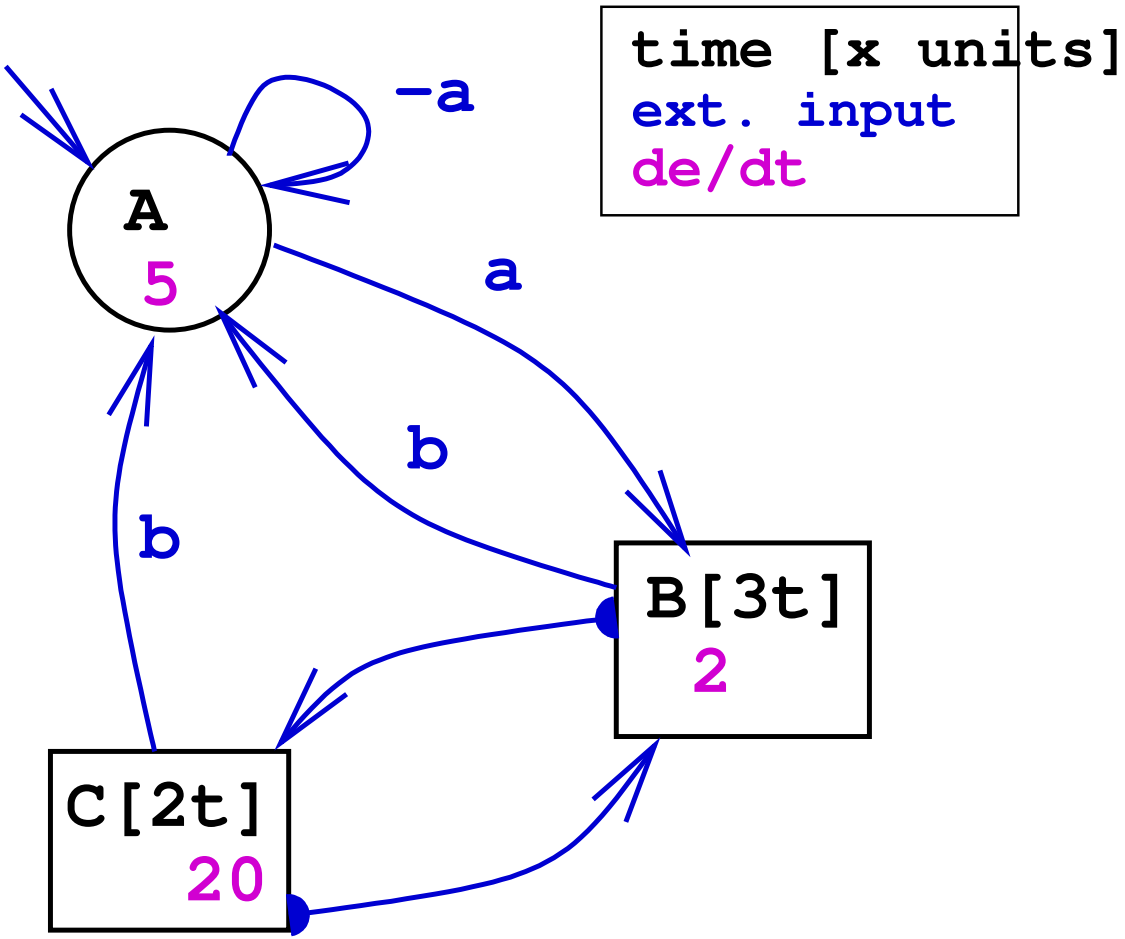
The HELP Approach

- Gather information on the consumption of components (as power-state models)
- Build a Transaction-level model of the SoC, and simulate it, including the SW

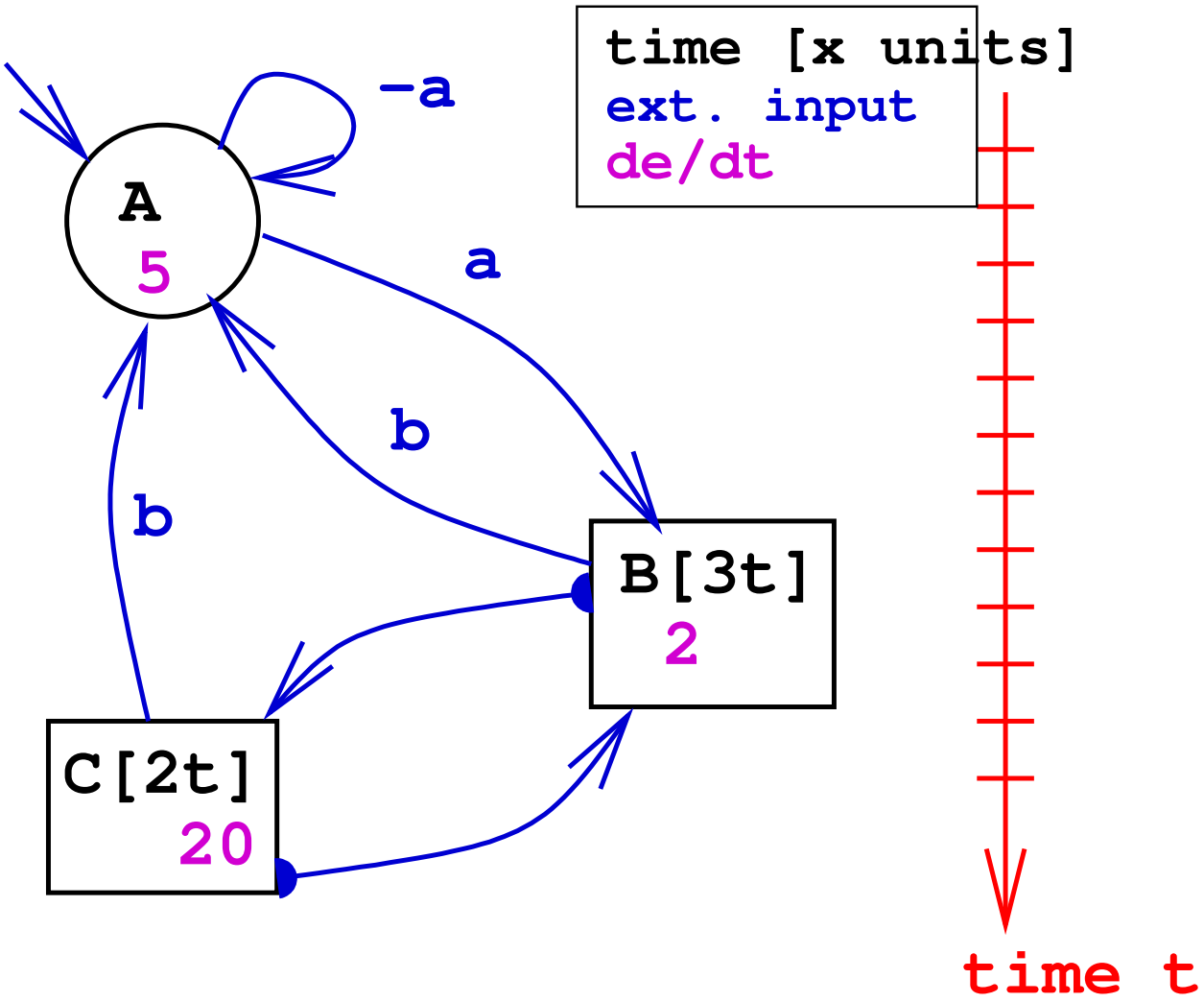
Power-State Models



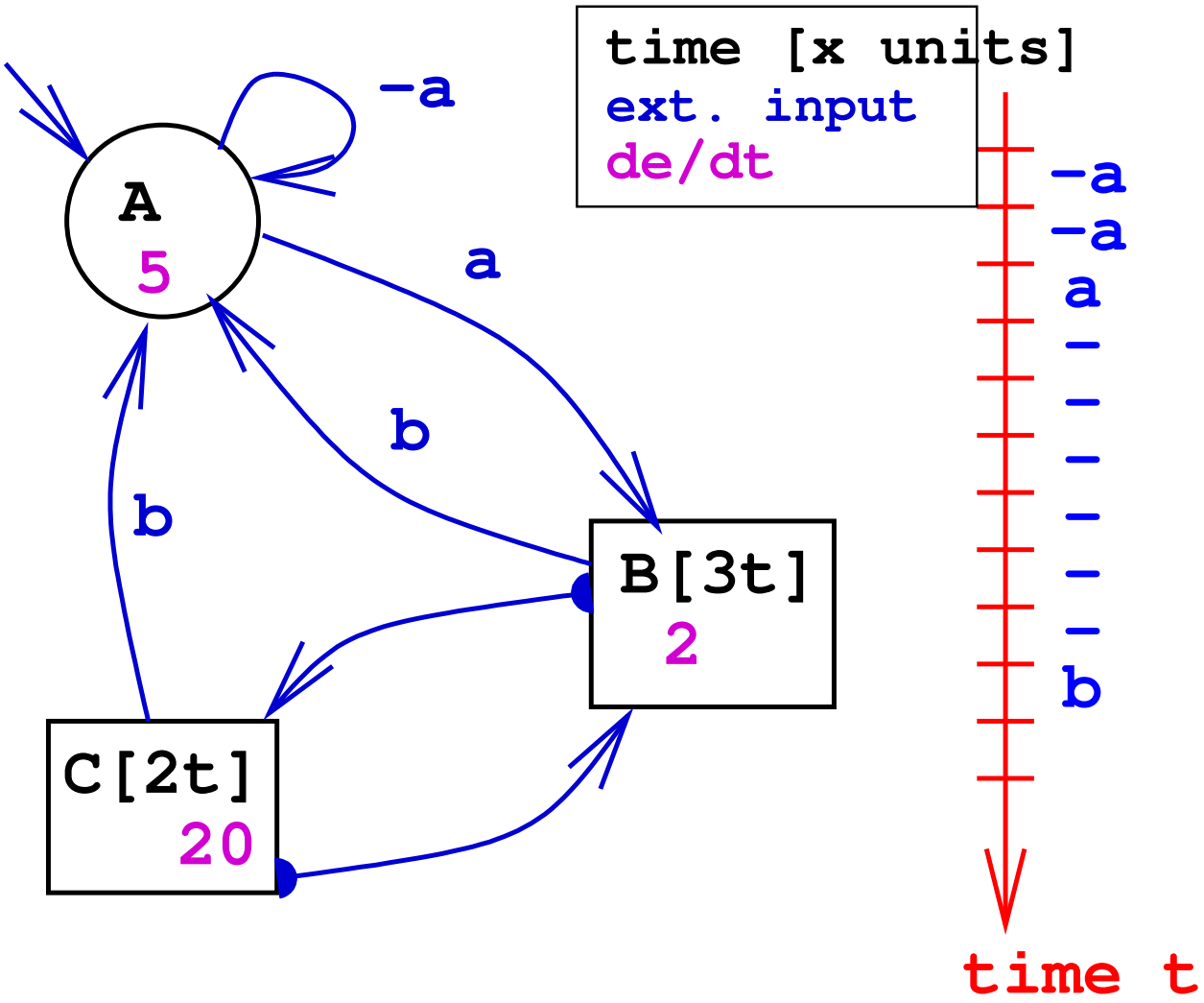
Power-State Models



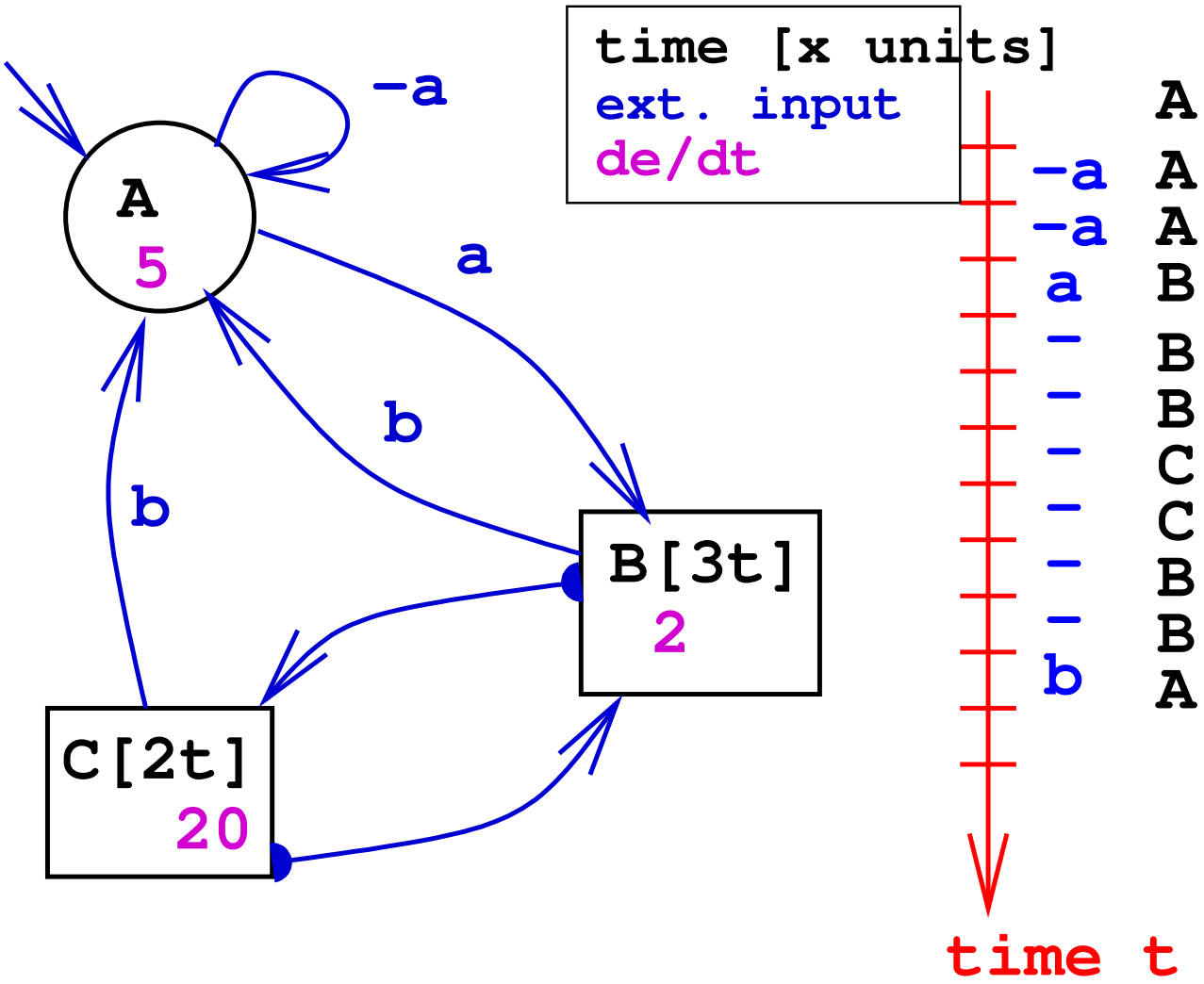
Power-State Models



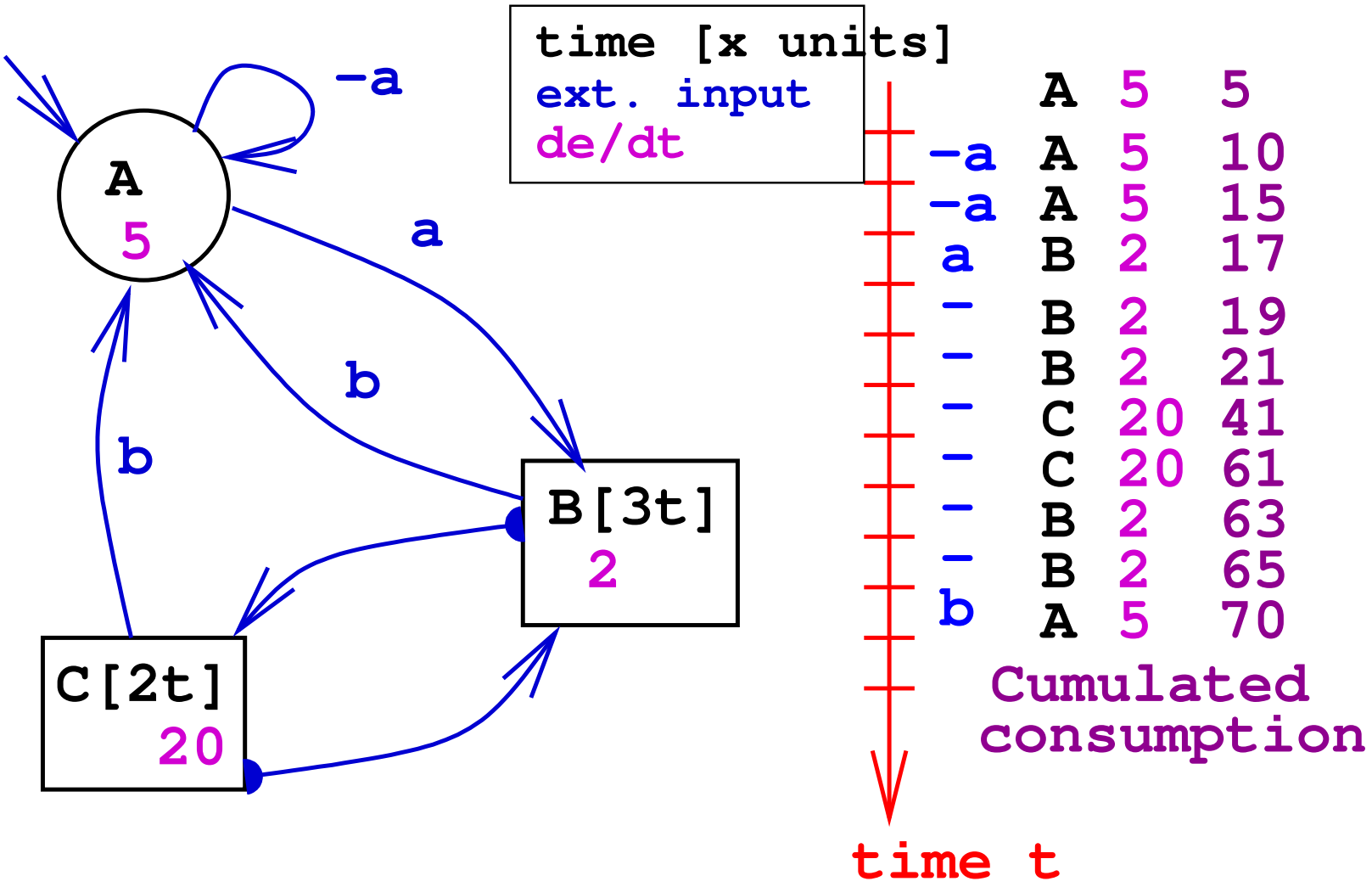
Power-State Models



Power-State Models



Power-State Models

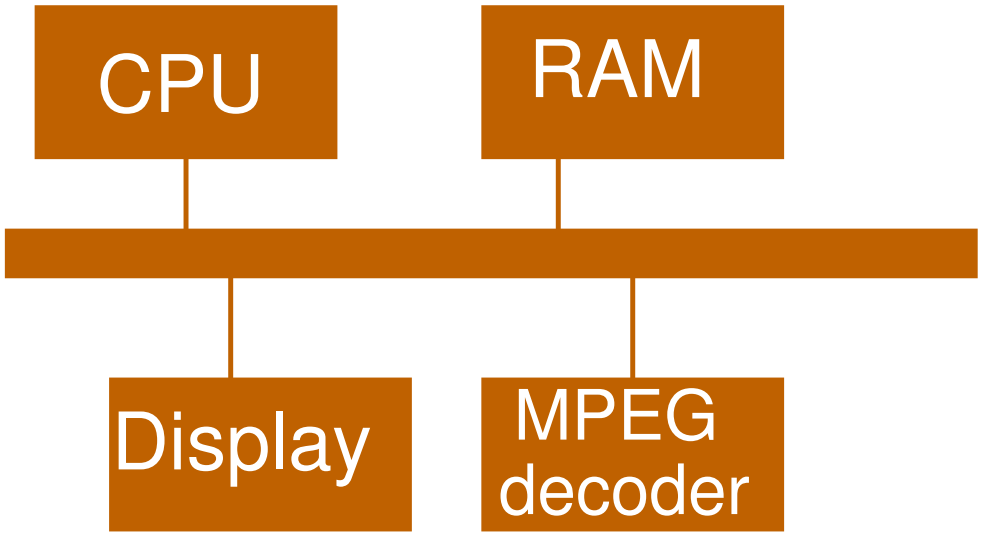


Modeling Time (when not cycle-accurate)

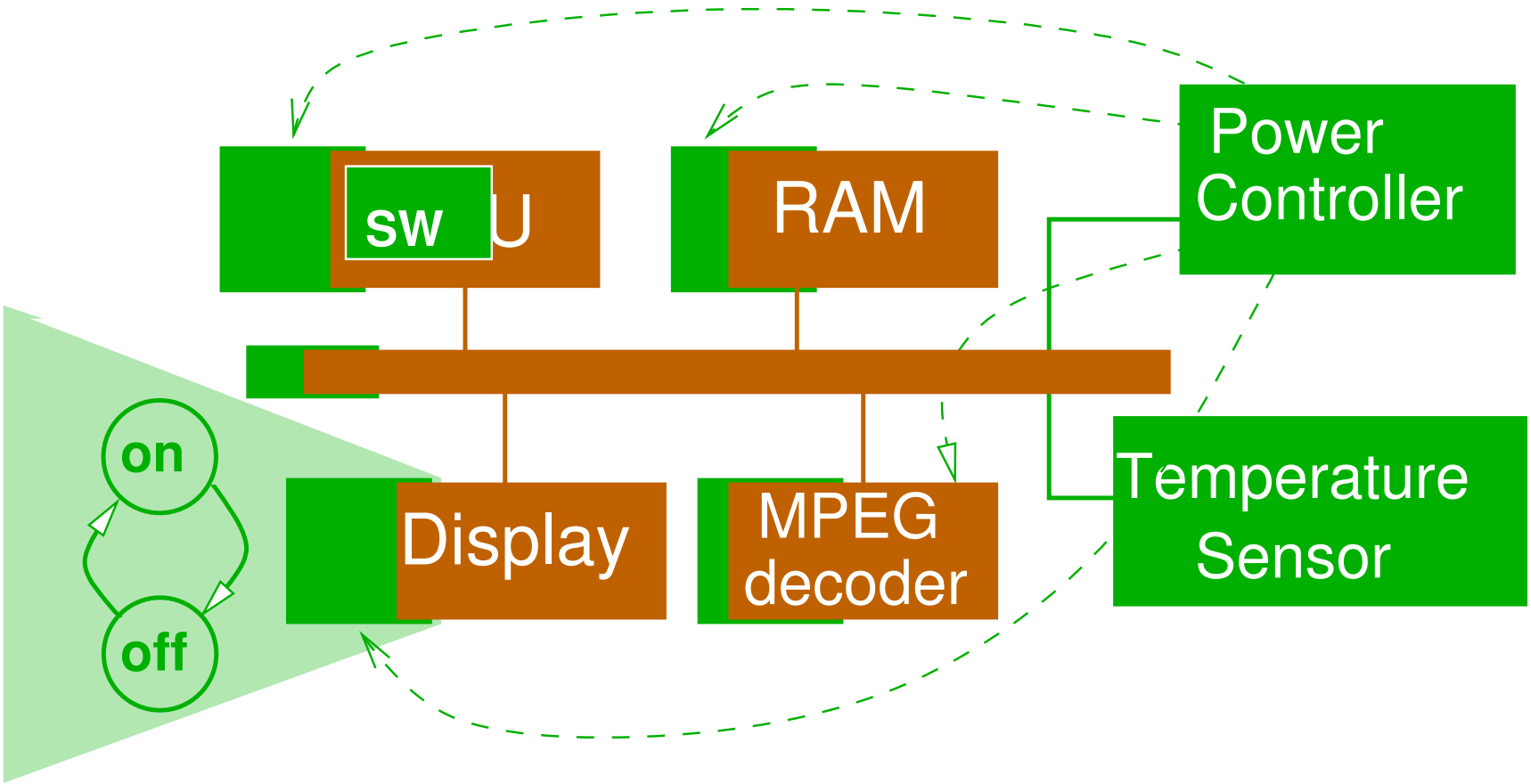
Active part (thread) in the model of a component:

```
...
f() ;                    // do some functional
                         //   effect locally
wait ([T1,T2]) ;         // pretend it takes some time
port.write (...); // communicate with another
                         // component by
                         // initiating a transaction
...
```

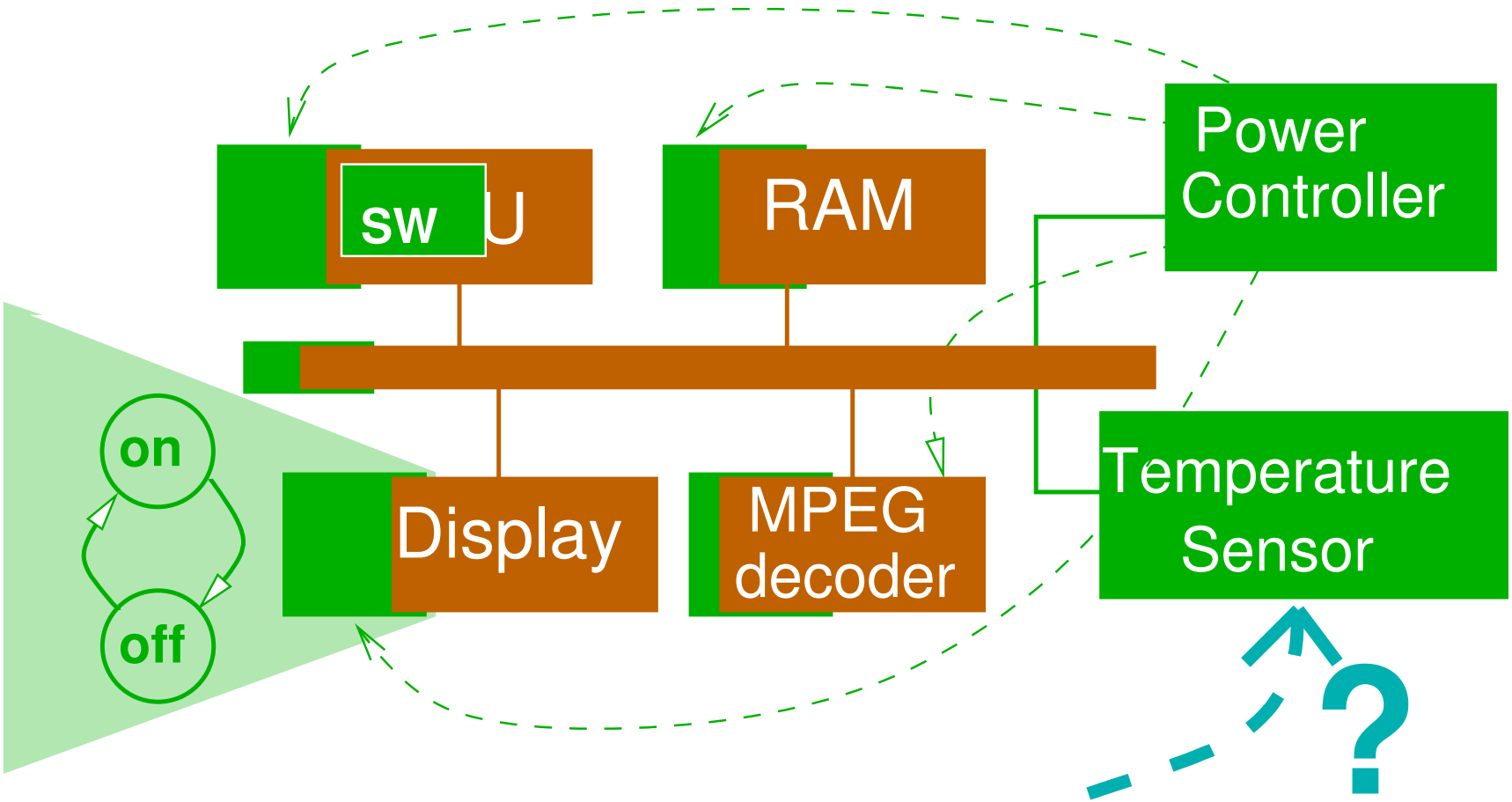
The HELP Generic Modeling Principles (1)



The HELP Generic Modeling Principles (1)



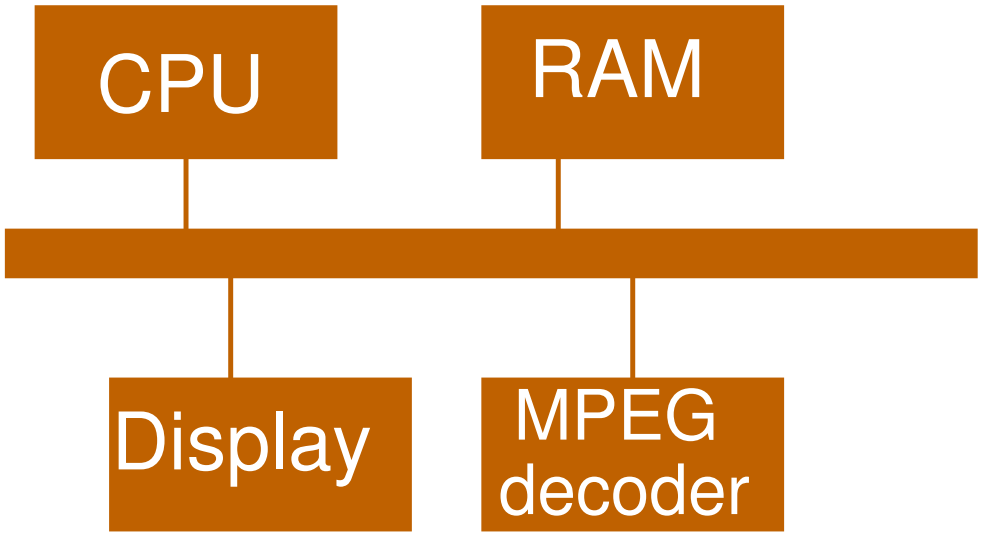
The HELP Generic Modeling Principles (1)



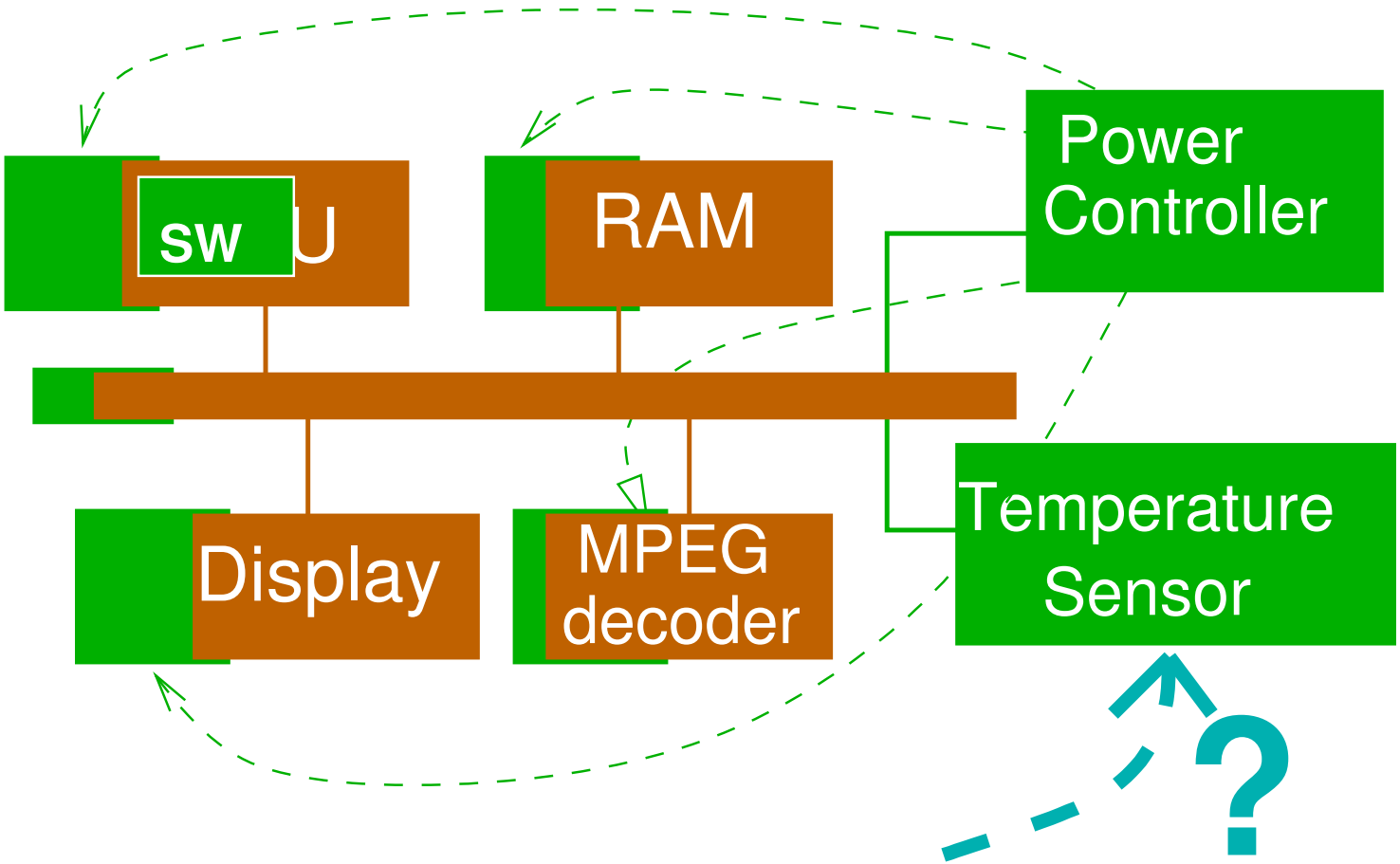
The HELP Generic Modeling Principles (1)

- Brown: the functional+approximate time TLM model of the chip, able to run the real software.
- Green: one "consumption automaton" per component, describing the current state of operation. The states of these automata are driven by the power manager actions. Examples:
  - retention mode for the RAM
  - several low-power modes for the CPU
  - ON/OFF for the display
  - ...

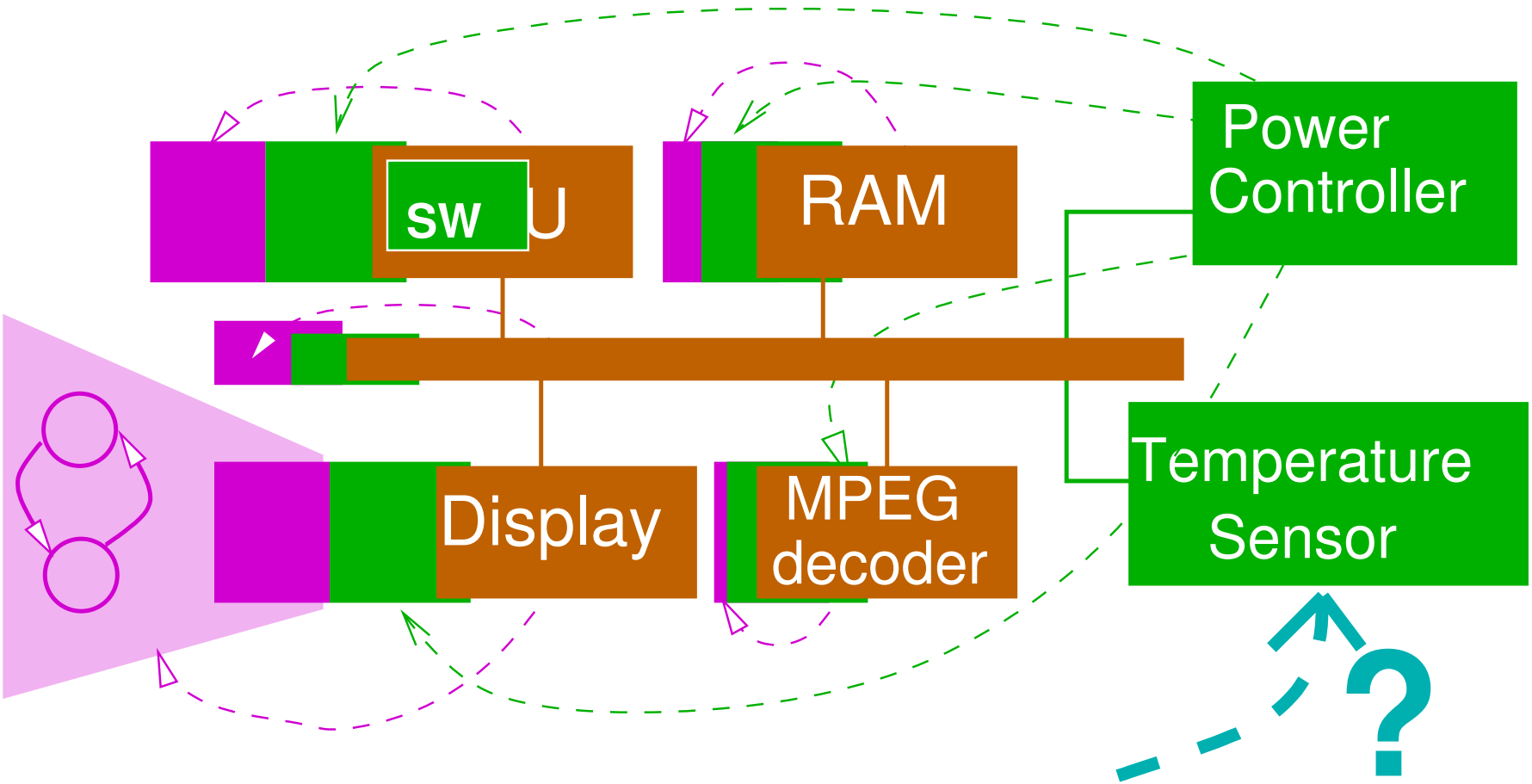
The HELP Generic Modeling Principles (2)



The HELP Generic Modeling Principles (2)



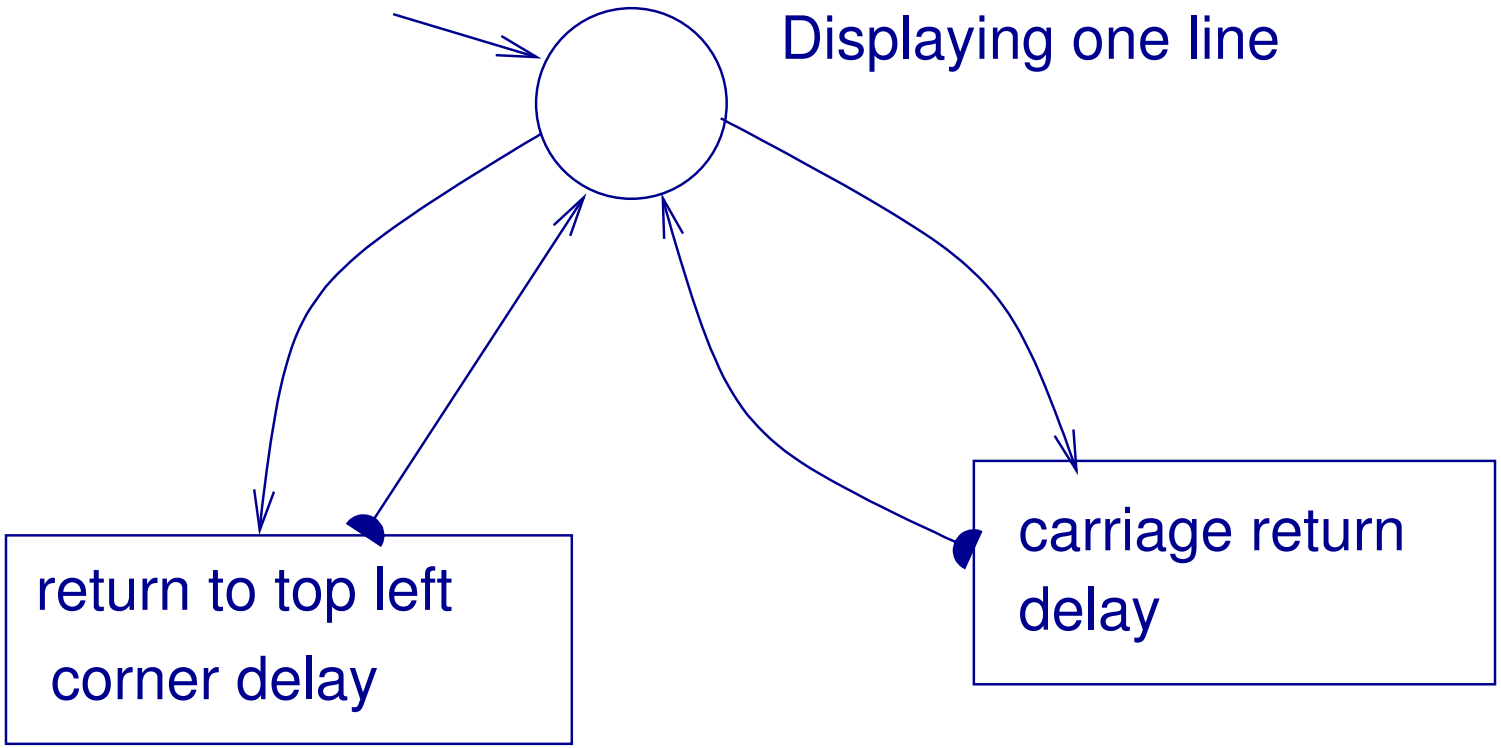
The HELP Generic Modeling Principles (2)



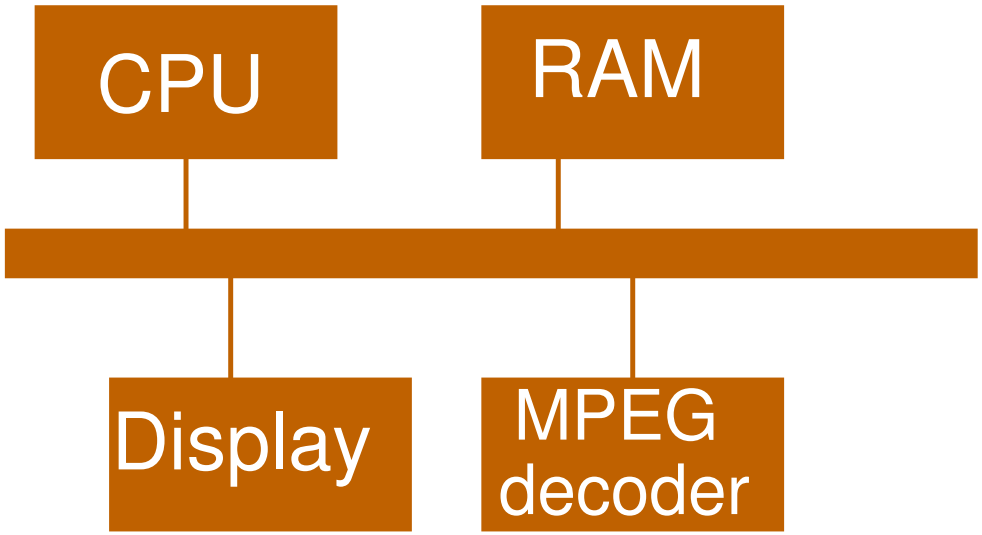
The HELP Generic Modeling Principles (2)

- Brown: the functional+time TLM model of the chip, able to run the real software.
- Green: one "consumption automaton" per component
- Purple: another consumption automaton per component, whose states are driven by the functional+time part. Example: the LCD display.

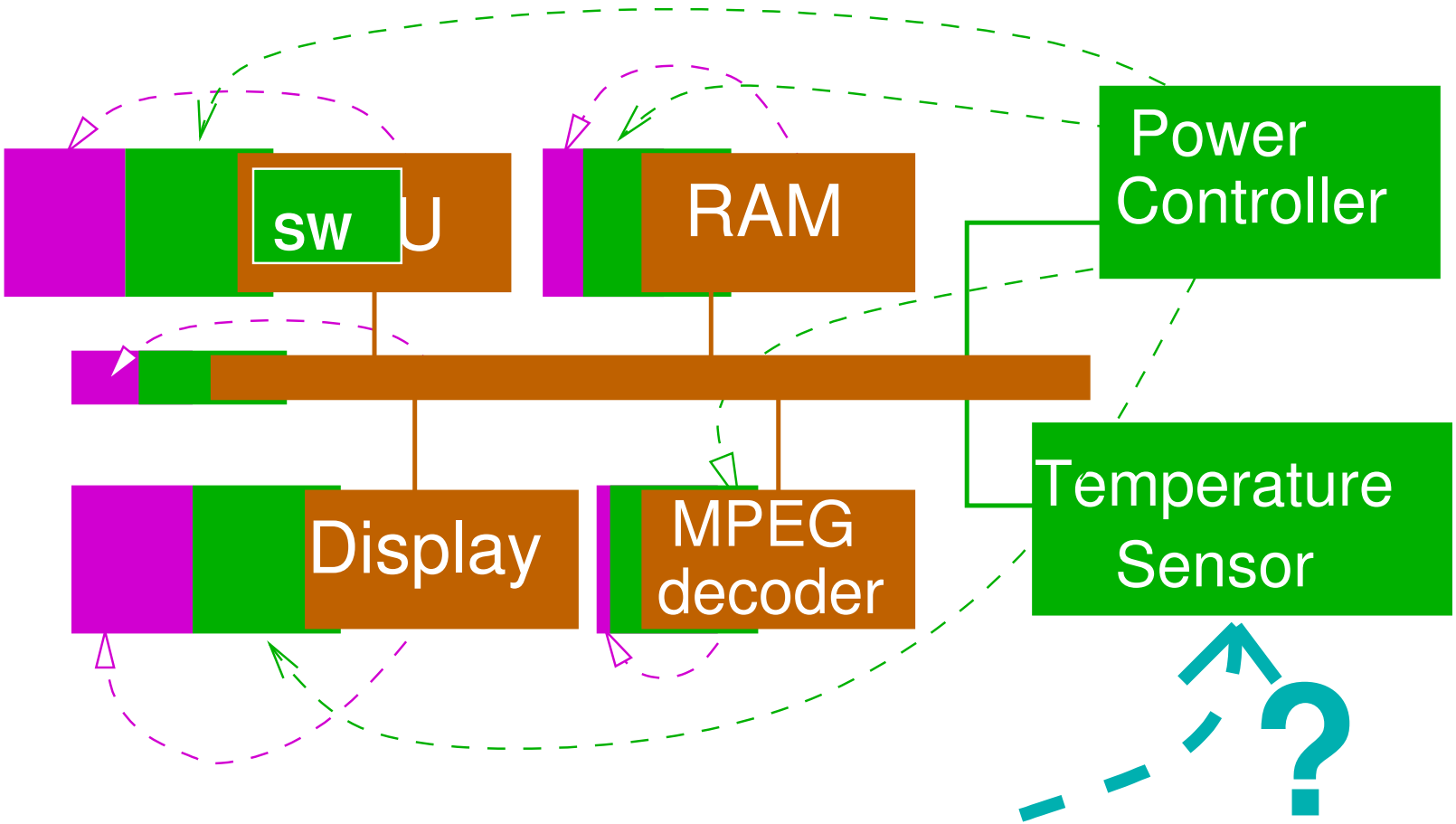
"Purple" consumption automaton for the LCD



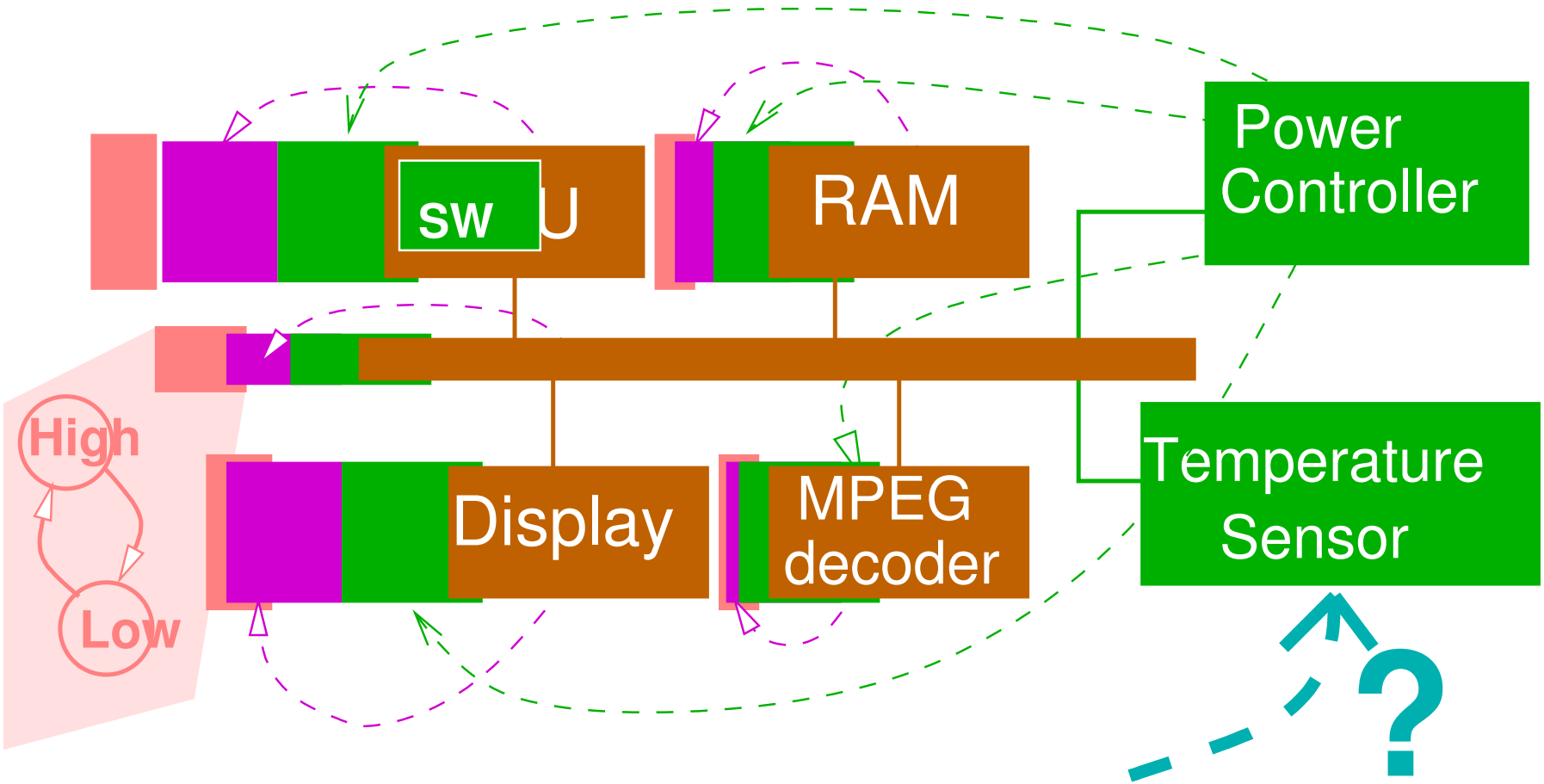
The HELP Generic Modeling Principles (3)



The HELP Generic Modeling Principles (3)



The HELP Generic Modeling Principles (3)



The HELP Generic Modeling Principles (3)

- Brown: the functional+time TLM model of the chip, able to run the real software.
- Green: one "consumption automaton" per component
- Purple: another consumption automaton per component
- Pink: yet another consumption automaton per component, whose states are driven by the *traffic*. Example: the bus

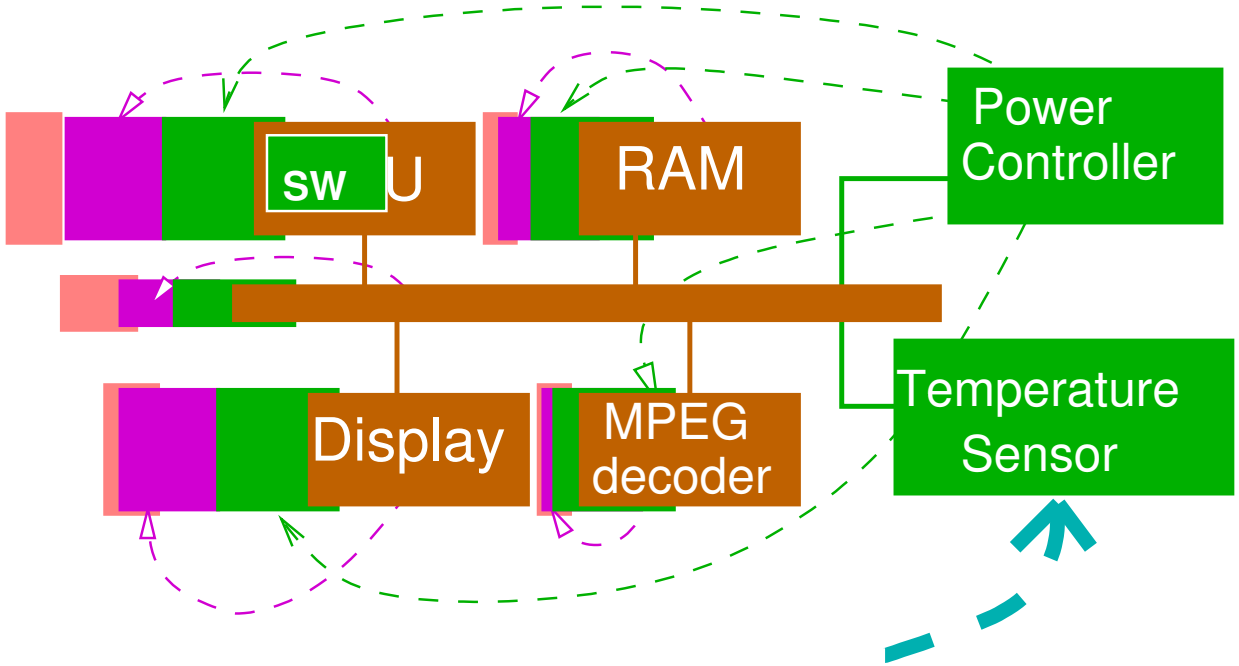
"Pink" consumption automaton for the Bus

For the bus, deducing the current consumption state from the functional behavior requires a very detailed functional model.

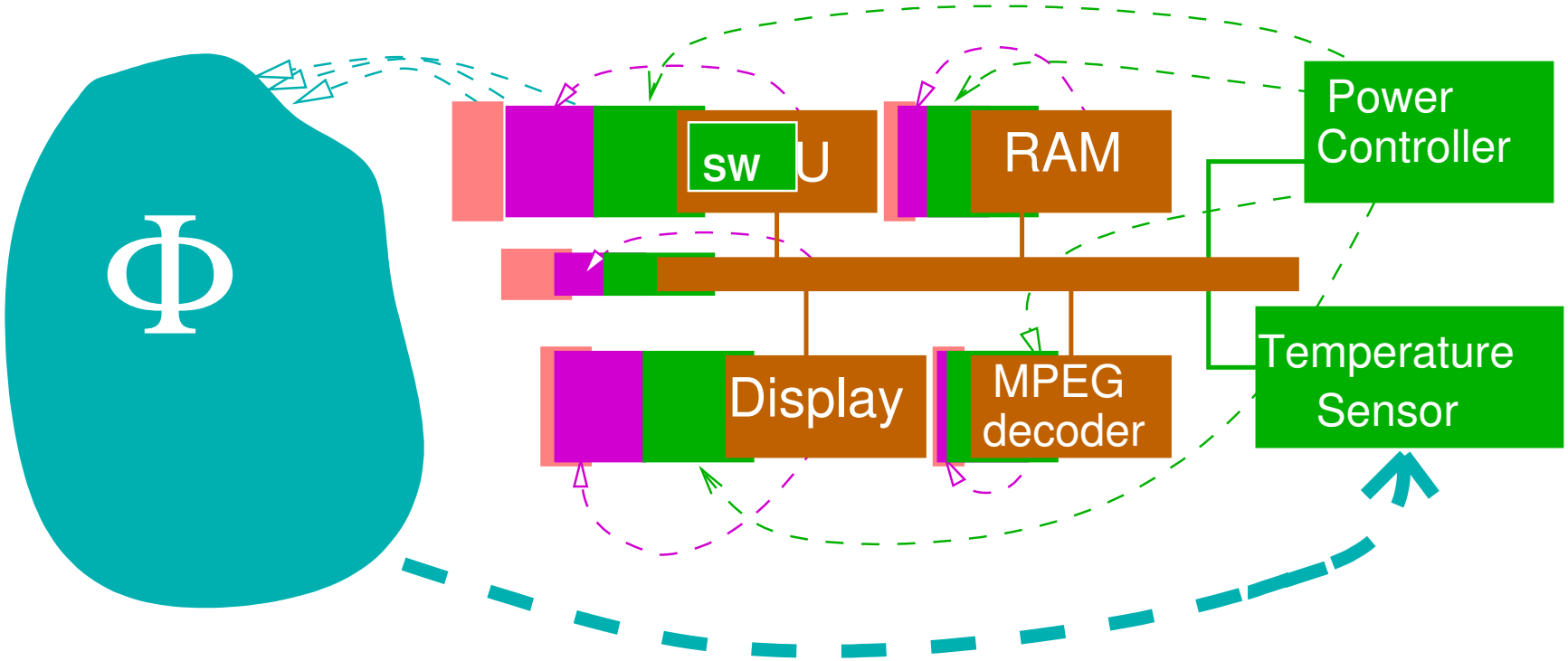
The significant consumption states are known from measures on existing busses, and can be related to the traffic.

In the model, the "Pink" consumption automaton is driven by an observation of the traffic (the input/output ports of the component).

The HELP Generic Modeling Principles (4)



The HELP Generic Modeling Principles (4)



1 Context

2 The HELP Generic Modeling Principles

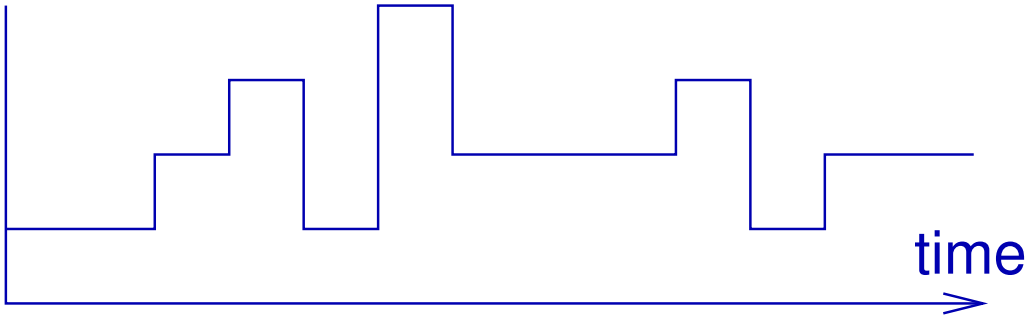
3 A Question Related to the Discrete/Continuous Parts

4 Ongoing Work

Discrete/Continuous Parts

Discrete

consumption



Continuous

Temp =
F (Consumption)

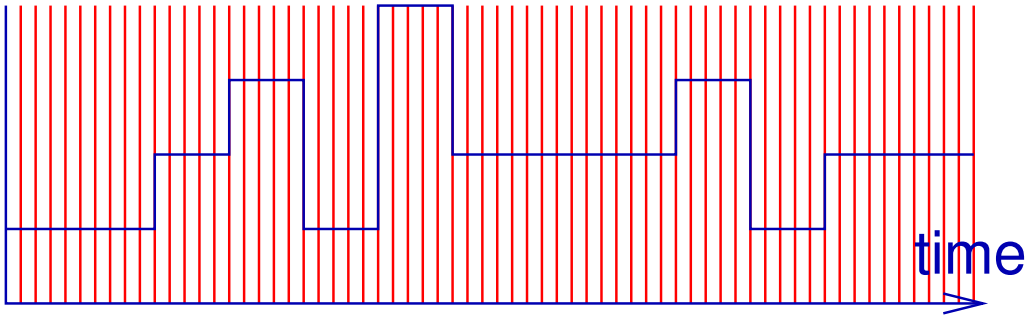
Tools:

DOCEA Aceplorer
ATMI
HotSpot

Discrete/Continuous Parts

Discrete

consumption



Model in a synchronous language

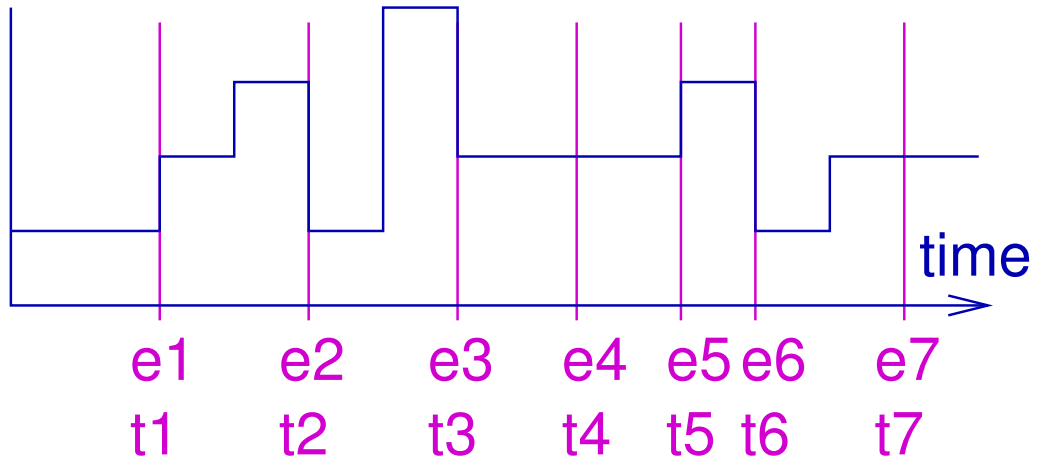
Continuous

Temp =
F (Consumption)

Discrete/Continuous Parts

Discrete

consumption



Model in a discrete–event modeling language.

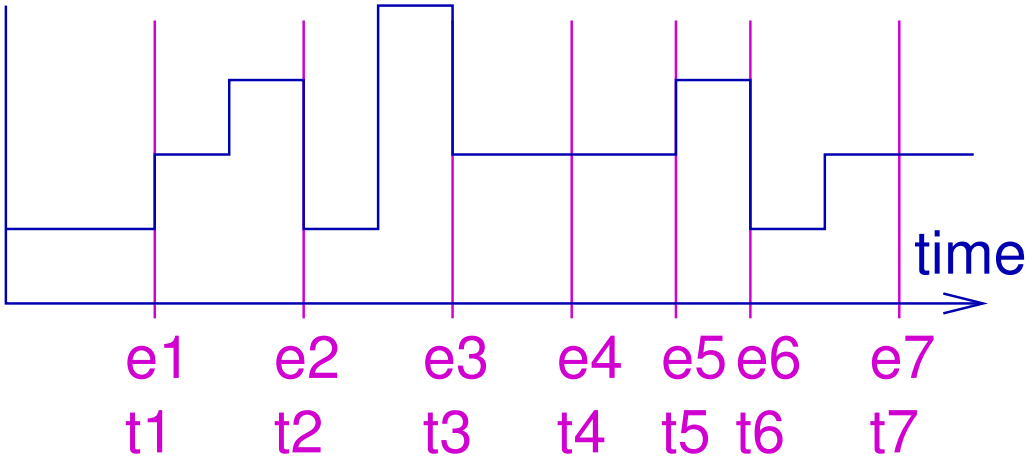
Continuous

Temp =
F (Consumption)

Discrete/Continuous Parts

Discrete

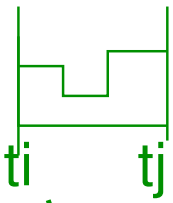
consumption

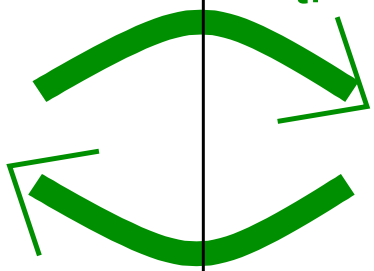


Model in a discrete–event modeling language.

Continuous

Temp =
F (Consumption)

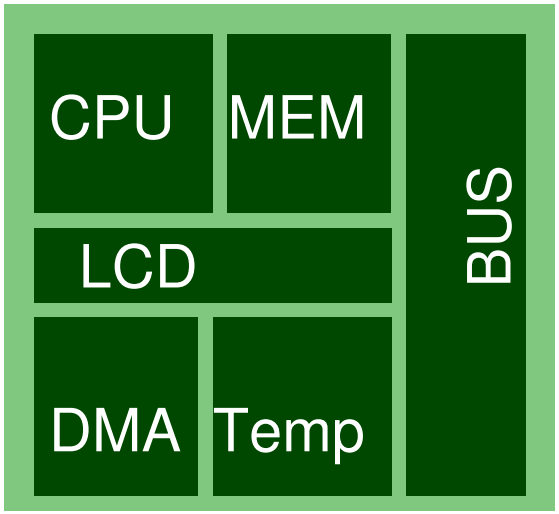




Temp at tj
(for the temp sensor)

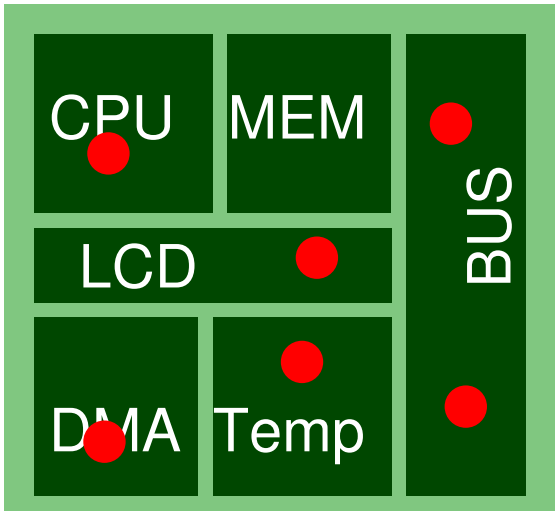
Temperature Model

Floorplan of
the SoC



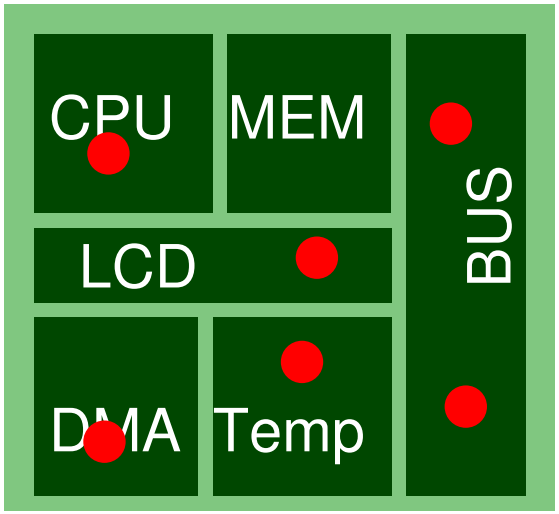
Temperature Model

Floorplan of
the SoC

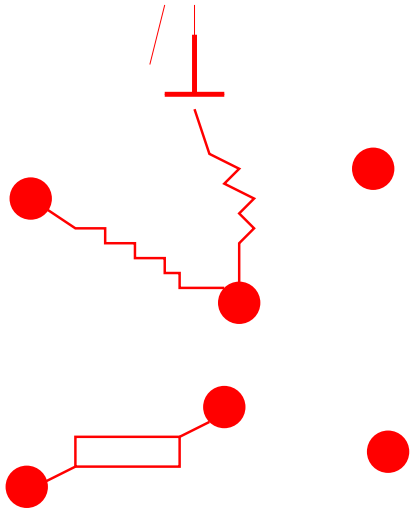


Temperature Model

Floorplan of the SoC



Electrical View of Temperature Propagation



1 Context

2 The HELP Generic Modeling Principles

3 A Question Related to the Discrete/Continuous Parts

4 Ongoing Work

Ongoing Work

- Decisions on when to switch between the two parts
- Formalization of the traces and errors that occur when "sampling" is slow
- Technical work on the connection (with thrift) between SystemC/TLM & DOCEA Aceplorer
- *Case study: comparison with measures on the real system*