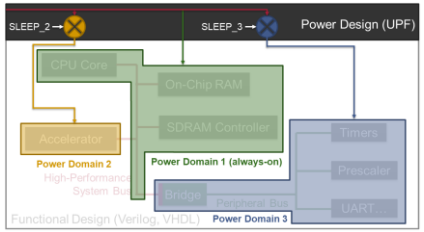
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MATHEMATISCH-
NATURWISSENSCHAFTLICHE FAKULTÄT
Embedded Systems

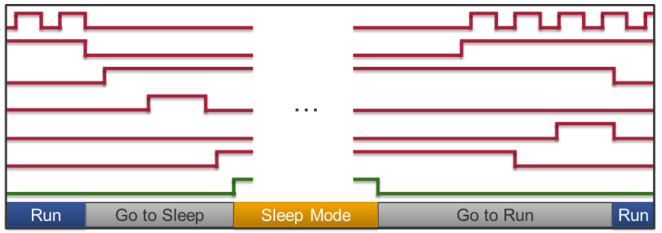
Unified Power Format

```
create_power_domain PD_2 -elements {/core/exec/alu/muldiv} \
    -supply {primary set_primary_PD_2} \
    -supply {default_isolation set_primary_TOP_VA} \
    -supply {default_retention set_primary_TOP_VA}
create_power_switch switch_0_PD_2 -output_supply_port \
    {supply_out net_vvdd_PD_2} -input_supply_port \
    {supply_in net_vdd_TOP_VA} -control_port {control_in} \
    -on_state {IS_ON supply_in {!control_in}} \
    -off_state {IS_OFF {control_in}}
add_power_state -domain PD_2 \
    -state {IS_ON -logic_expr {set_primary_PD_2 == IS_ON}}
add_power_state -domain PD_2 \
    -state {IS_OFF -logic_expr {set_primary_PD_2 == IS_OFF}}
```

Power Design



Power Management



Fully-automated Synthesis of Power Management Controllers from UPF

The Long and Winding Road:
How we can implement power management controllers in UPF…?

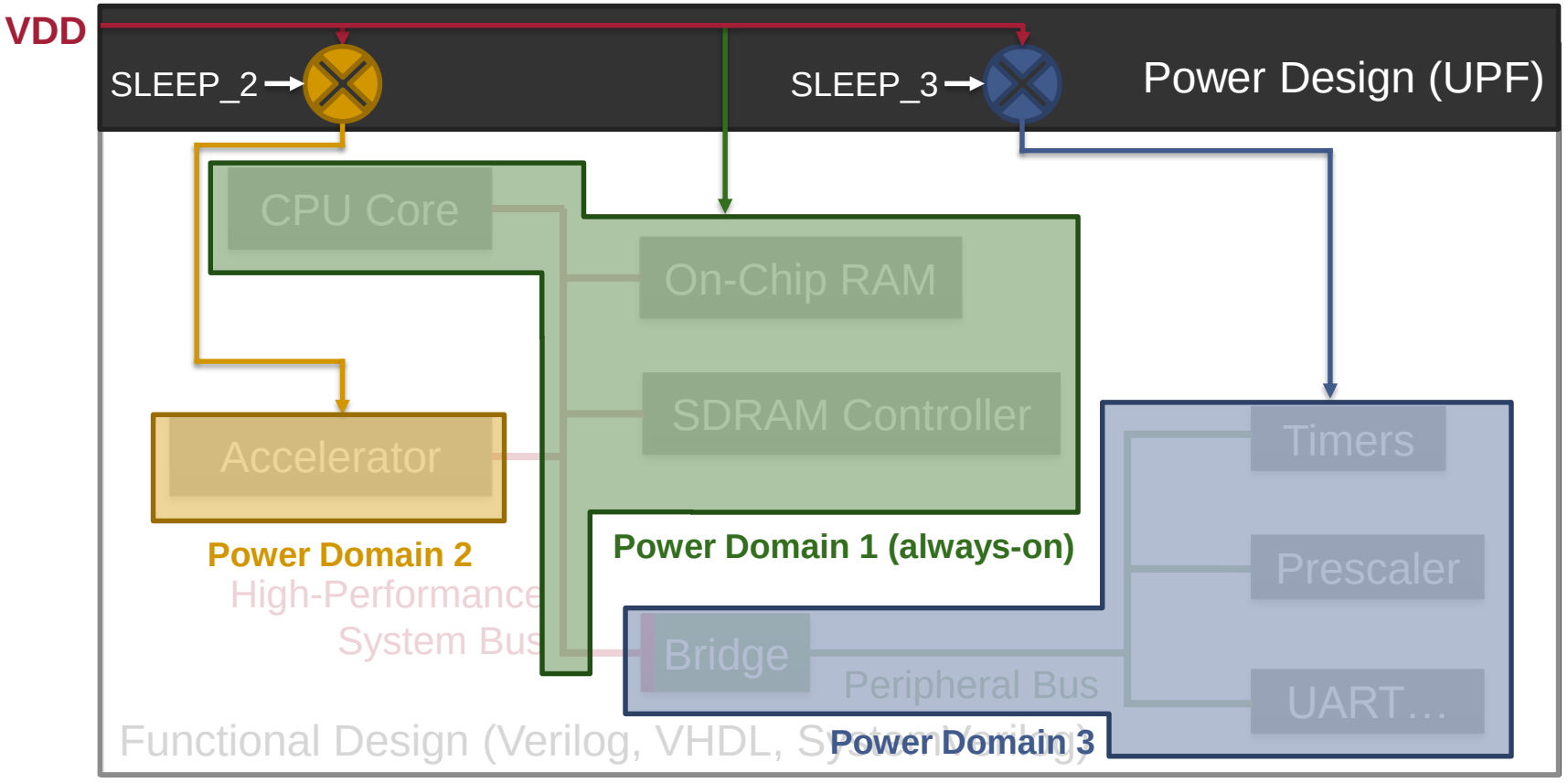
22.01.2019, Dustin Peterson | University of Tuebingen

Agenda

1. Concepts and Limitations of UPF
2. Extensions to the Unified Power Format
3. PMC Synthesis
4. Implementation + Results
5. Summary + Roadmap

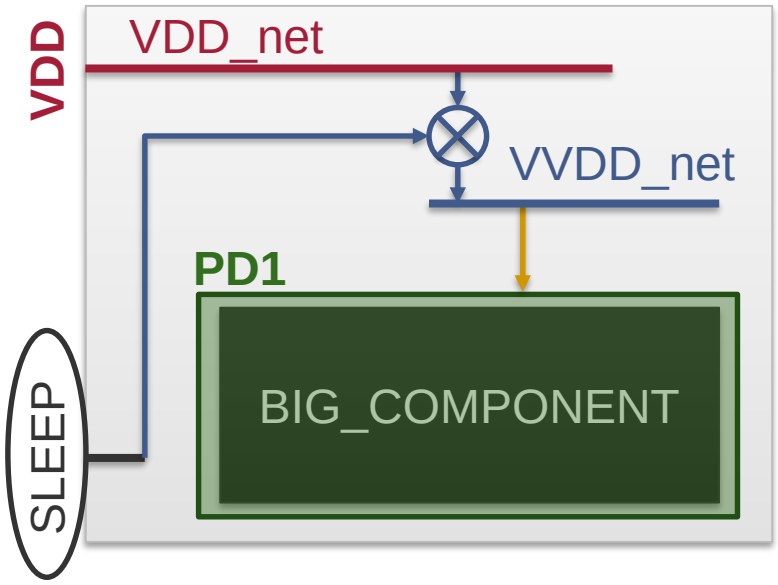
Unified Power Format (UPF)

- Tcl-based power design specification language (IEEE-1801) supported by most commercial synthesis and simulation tools
- **Basic Paradigm:** Separation of Concerns (Power vs. Function)



A Rapid UPF Example…

- **Example:** How can we implement power gating by UPF?



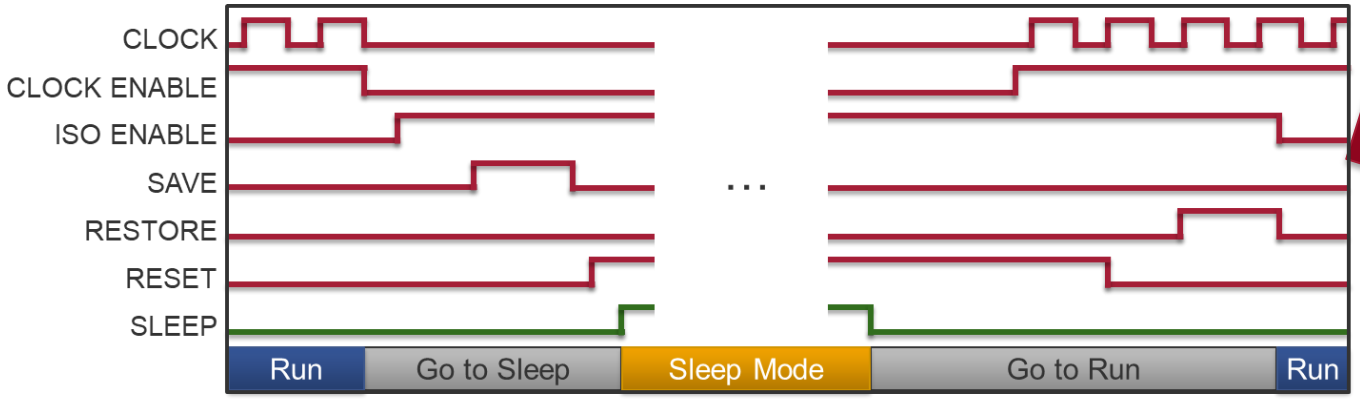
We want this SLEEP signal to control the power gate!

```
create_supply_port VDD
create_supply_net VDD_net
connect_supply_net VDD_net
  -ports {VDD}
create_power_domain PD1
  -elements {BIG_COMPONENT}
create_supply_net VVDD_net
create_power_switch
  -input_supply_port {VDD VDD_net}
  -output_supply_port {VDDG VVDD_net}
  -control_port {SLEEP SLEEP}
  -on_state {ON VDD !SLEEP}
  -off_state {OFF SLEEP}
set_domain_supply_net PD1
  -primary_power VVDD_net
```

A Rapid UPF Example…getting more complex!

- But instead of just switching the sleep signal like this…

…you have to do a lot more stuff for each power domain!



Power Management Controller (PMC) undertakes this task!

Currently, this needs to be implemented in RTL code!

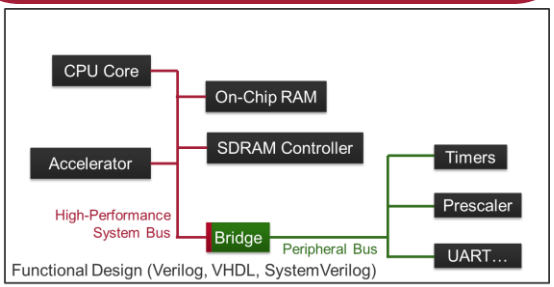
Goals of our Approach

- Keeping RTL clean from power management intent.
- Automatic synthesis of one or multiple power management controllers (PMC) from UPF + integration back into the SoC.
- PMC controlled by software (bus I/O) or hardware.

1.

```
create_power_domain PD_2 -elements {/core/exec/alu/muldiv} \
    -supply {primary set_primary_PD_2} \
    -supply {default_isolation set_primary_TOP_VA} \
    -supply {default_retention set_primary_TOP_VA}
create_power_switch switch_0_PD_2 -output_supply_port \
    {supply_out net_vvdd_PD_2} -input_supply_port \
    {supply_in net_vdd_TOP_VA} -control_port {control_in} \
    -on_state {IS_ON supply_in {!control_in}} \
    -off_state {IS_OFF {control_in}}
add_power_state -domain PD_2 \
    -state {IS_ON -logic_expr {set_primary_PD_2 == IS_ON}}
add_power_state -domain PD_2 \
    -state {IS_OFF -logic_expr {set_primary_PD_2 == IS_OFF}}
```

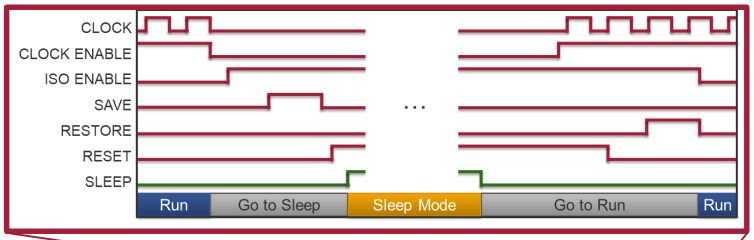
Extended UPF Script

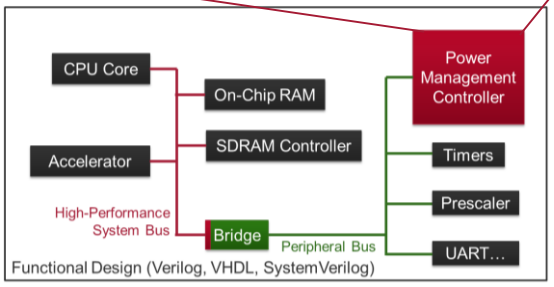


RTL Design

Power Management Synthesis

2.

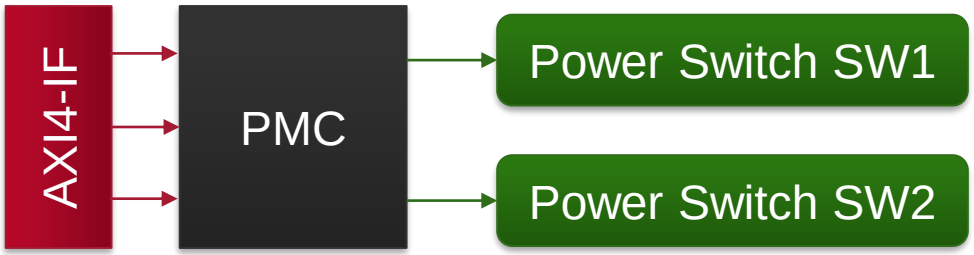




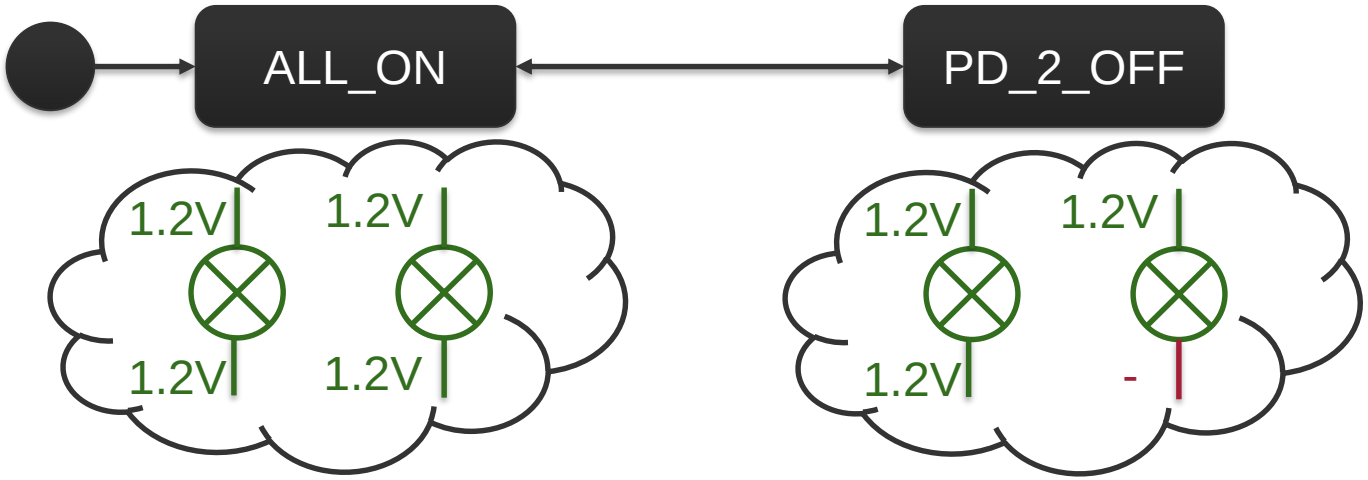
RTL Design with PMC

What UPF Extensions do we need?

- We need a way to specify…
  - …various power management controllers with several interfaces (AXI, APB, …) managing different devices!



  - …power state machines per power management controller!

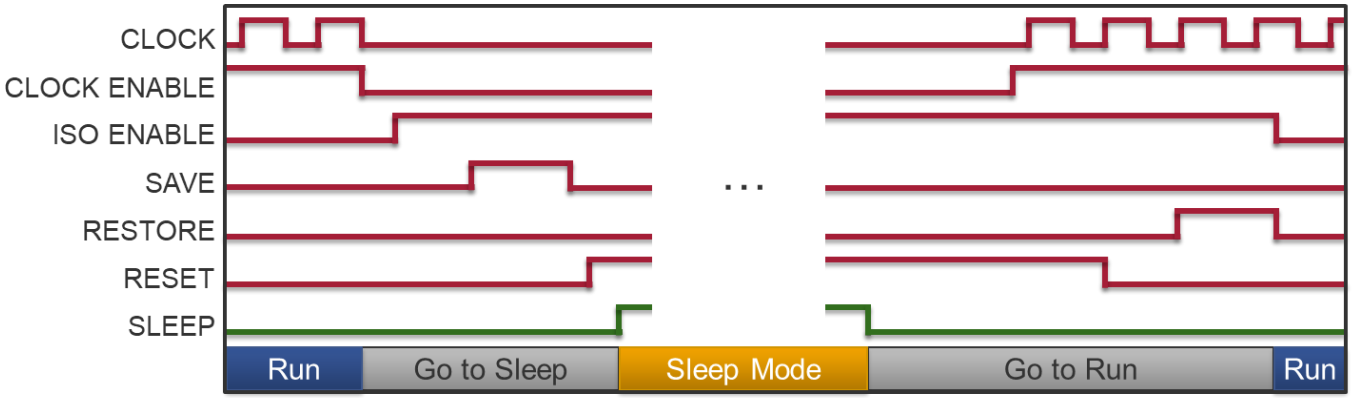


What UPF Extensions do we need?

- We need a way to specify…
  - …which actions we need to trigger during transition?

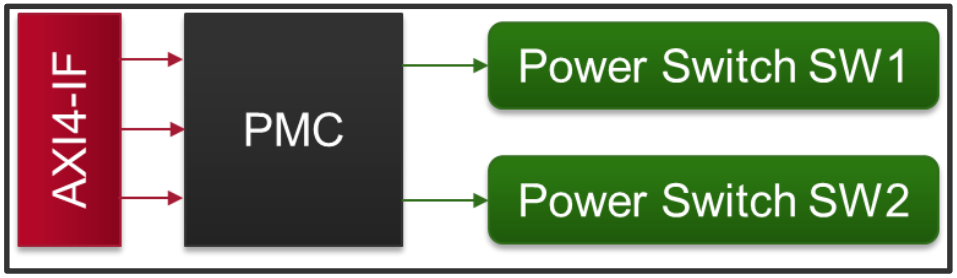
„Disable SW_2, Enable Isolation, Enable Clock Gating, …“

  - …and in what order do we need them to be triggered?



Extended Unified Power Format

- **Starting Point:**
  Basic PMC I/O Specification
  + List of Managed Devices



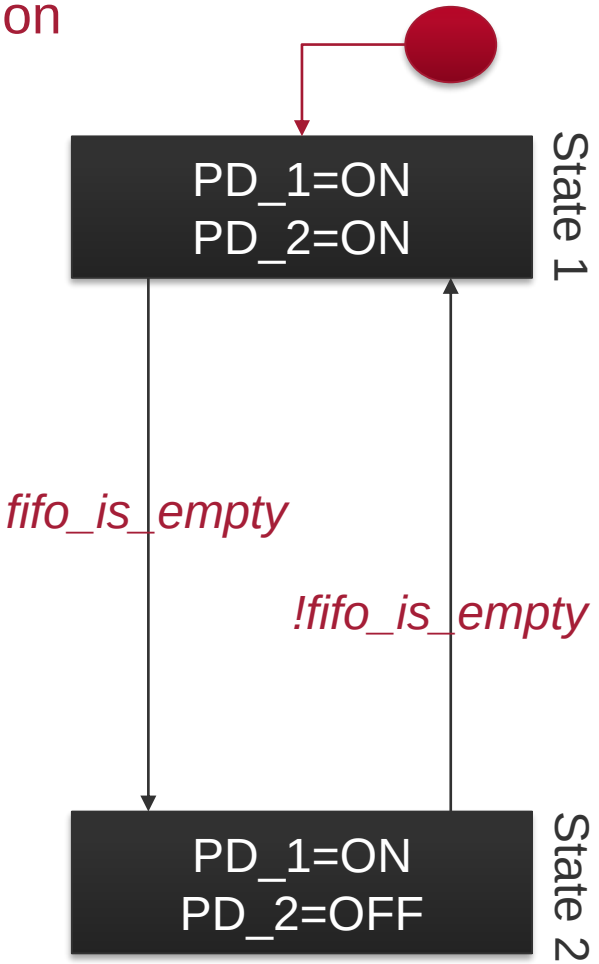
Purpose	Create a new power management controller in the current scope that implements the given power state machine.
Syntax	**create_pmc** *pmc_name* [**-devices** { *power_switch* | *voltage_source* }*] [**-domain** *domain_name*] **-interface** <*plain* | *self* | *apb3* | *axi4lite* | *avalon* | *wishbone*> [**-interface_param** { *name value* }]* [**-extend_interface** {*target_scope*}] **-clock** {*clock_signal* [<*posedge*|*negedge*>]} **-reset** {*reset_signal* [<*sync*|*async*> <*low*|*high*>]} [**-scheduler** <*default* | *custom*>]

Extended Unified Power Format

- Existing commands to specify power state machine for each PMC:
  - add_power_state, describe_state_transition
- **But: Some information is missing!**
  - Initial State? Under which condition shall a transition be taken?

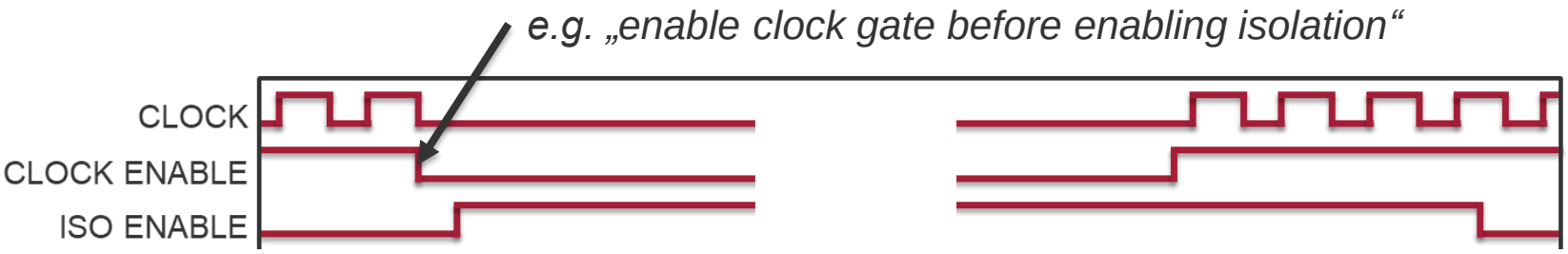
Purpose	Sets the initial state of a power management controller.
Syntax	**set_initial_state** *pmc_name* **-state** *state_name*

Purpose	Enhanced version of IEEE-1801 *describe_state_transition* with support for transition conditions.
Syntax	**describe_state_transition** *transition_name* **-object** *object_name* [**-from** *from_list*] [**-to** *to_list*] [**-paired** *{(from_state to_state)*}*] [[**-legal**|**-illegal**]] [**-condition** *{boolean_expression}*]



Extended Unified Power Format

- Actions are triggered when switching between two power modes!
  - In what order do they need to be triggered?

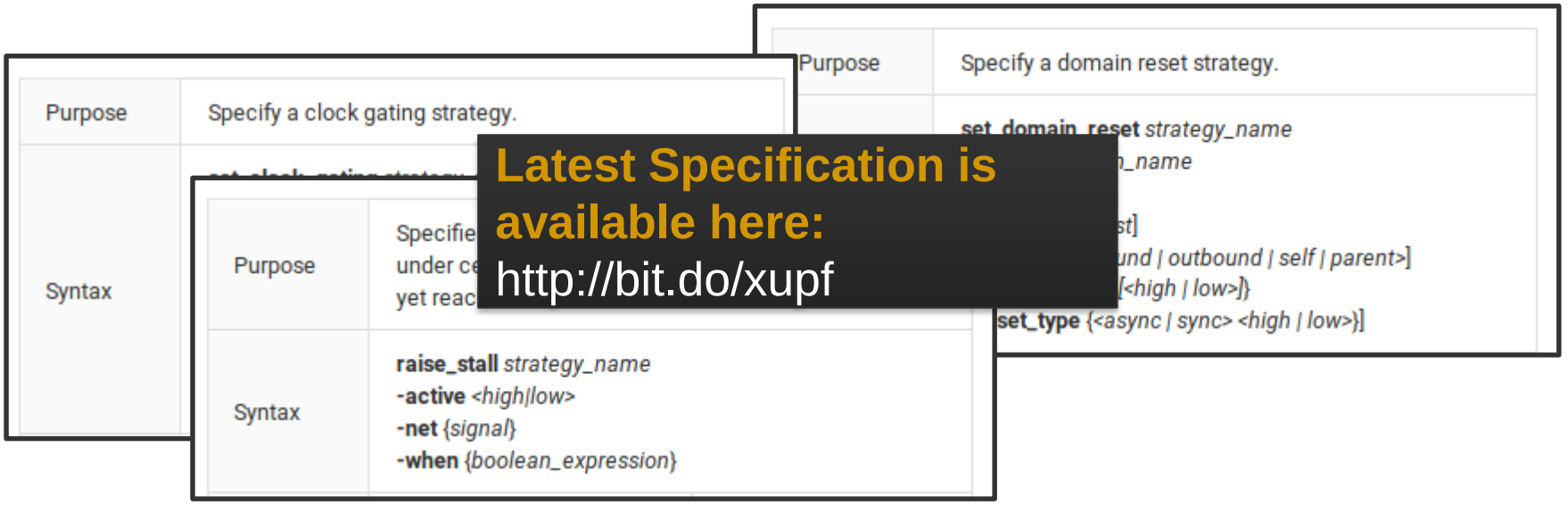


- Three commands are intended for that:
  - **schedule_strategy:** Which strategy needs to be enabled/disabled in which device state / power state?
  - **update_strategy_schedule:** Strategy dependencies
  - **update_device_schedule:** Device dependencies

- Instead of an RTL signal, control handles (-control_port of isolation, retention, … shall reference a PMC object!)

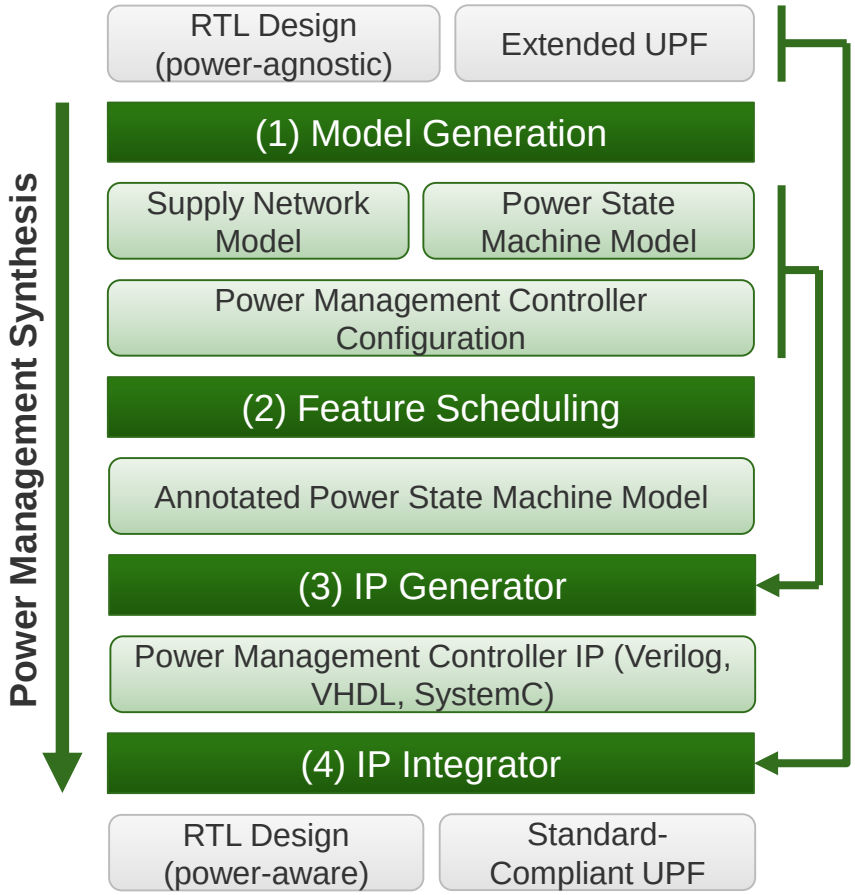
Extended Unified Power Format

- A lot of more commands have been specified to…
  - Specify clock gating strategies.
  - Specify reset strategies.
  - Specify voltage sources.
  - Map voltage sources to IP cells.
  - Specify (pipeline stall) signals and when to trigger them.
  - Specify existing power gating implementations (memory IPs…)



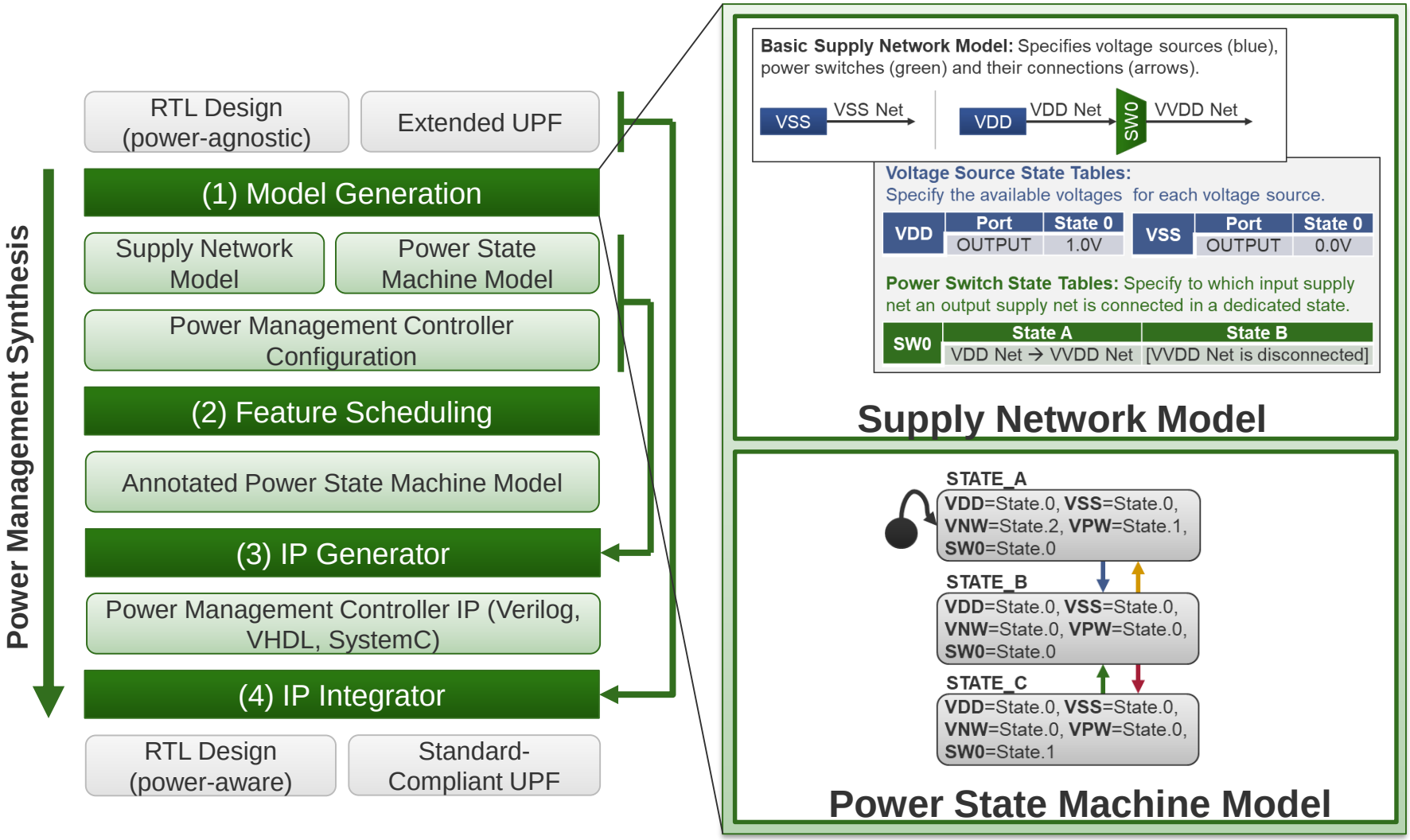
Power Management Synthesis

- **Power Management Synthesis:** Transform an RTL design + xUPF specification into a fully power-managed SoC platform!



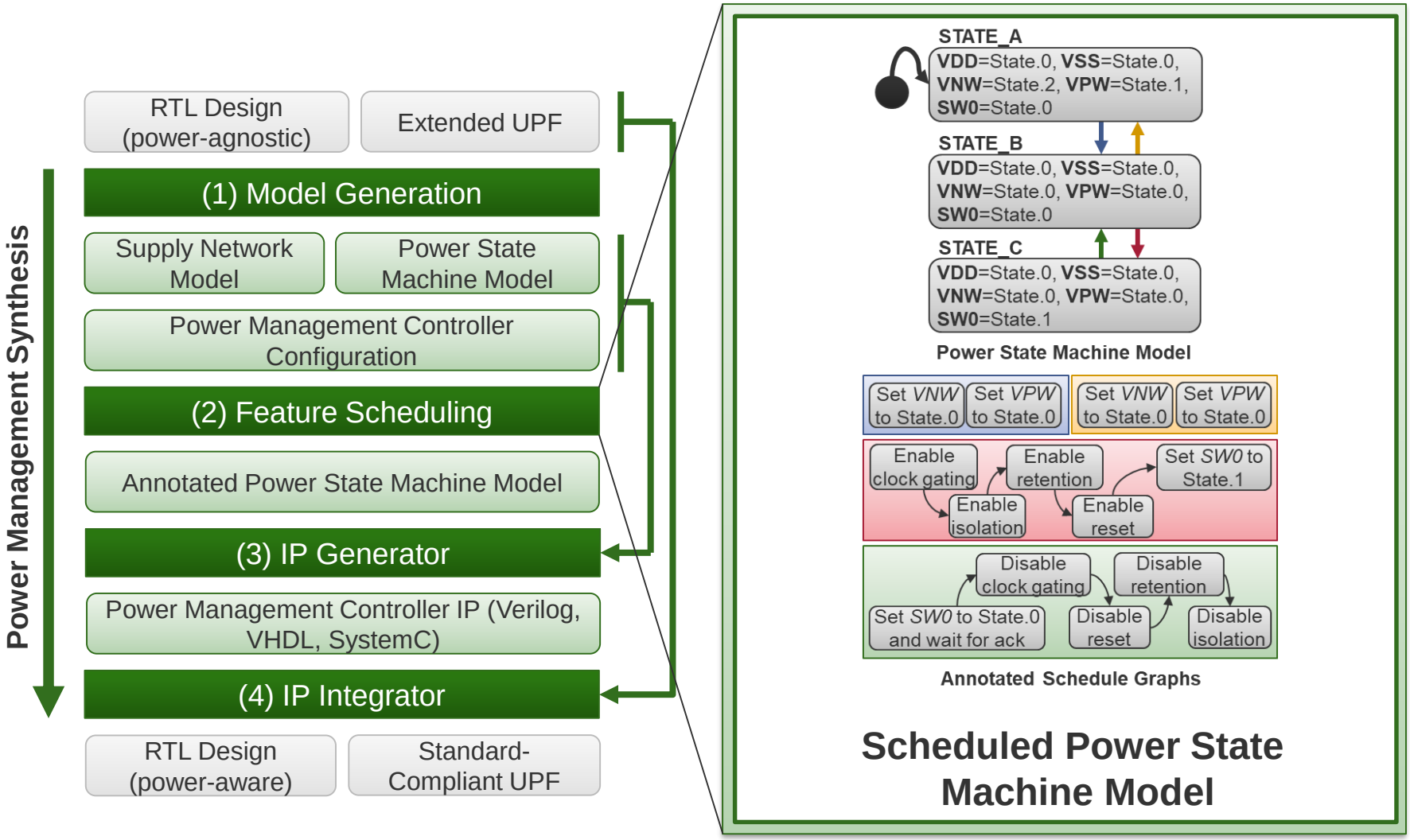
Latest Specification is available here:
http://bit.do/xupf

Power Management Synthesis



Latest Specification is available here:
http://bit.do/xupf

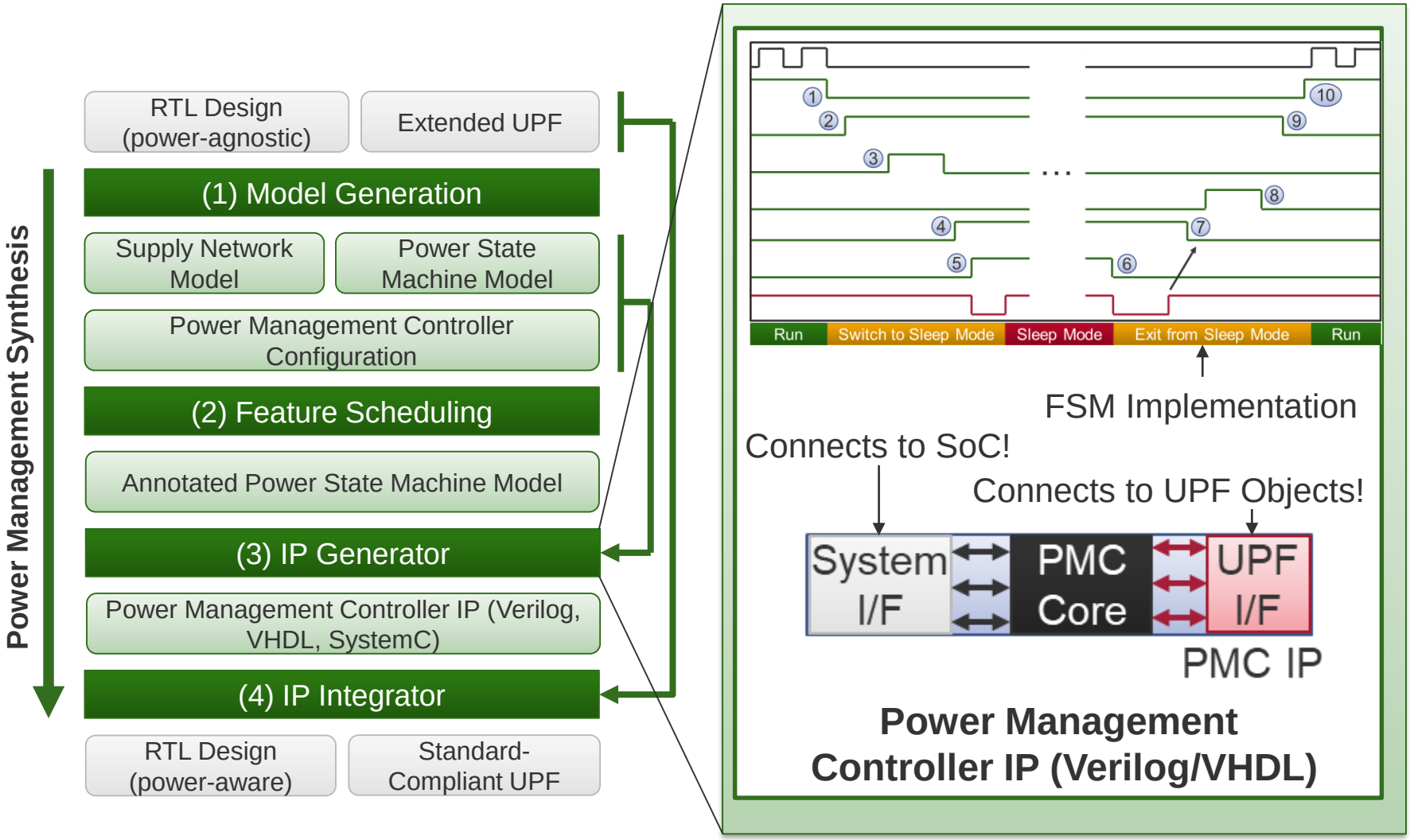
Power Management Synthesis



Latest Specification is available here:
http://bit.do/xupf

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

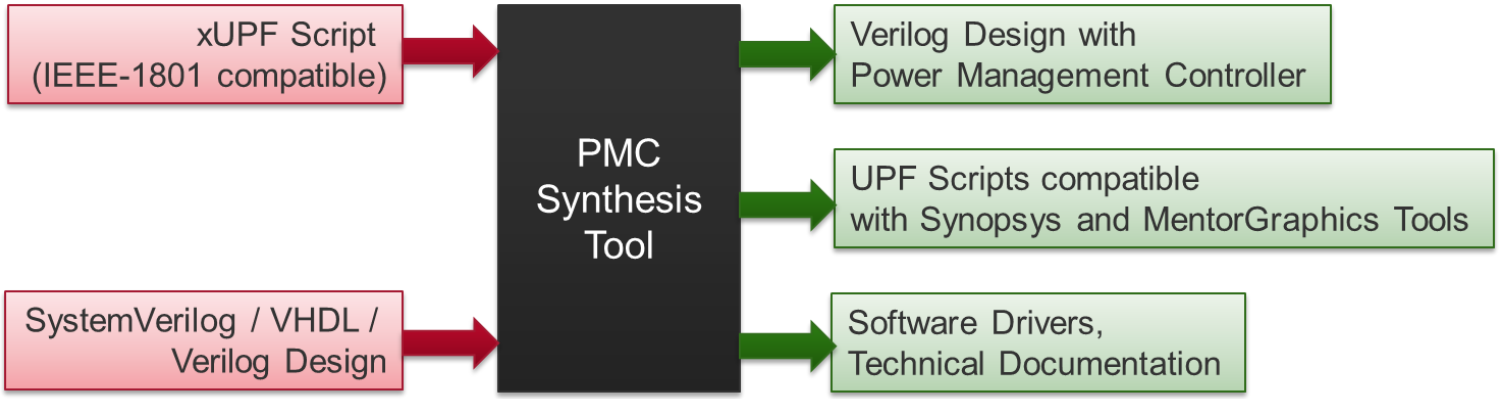
Power Management Synthesis



Latest Specification is available here:
http://bit.do/xupf

Implementation as Synopsys Design Compiler Plugin

- Implementation of a full-blown UPF library in Scala/Java including a power management synthesis tool!



- Integration into Design Compiler using a Tcl interface!

```
analyze -library ${library} -format verilog ${verilog_sources}
elaborate ${top_design}
source -echo ${sdc_input}
uniquify
```

replaces load_upf!

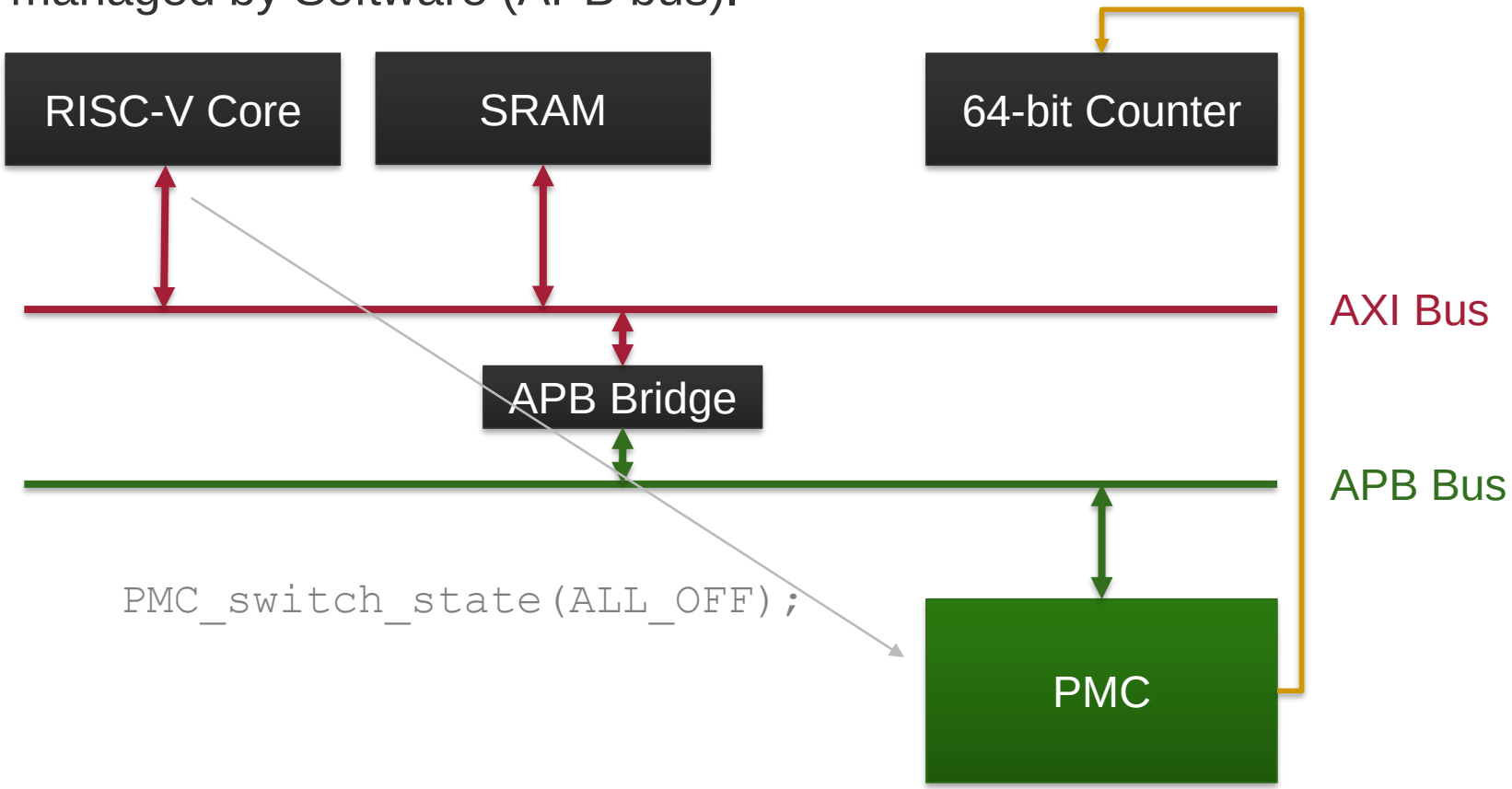
```
# Synthesize power management
##############################
#load_upf ${upf_file}
pmc_compile ${upf_file} -output ${pmc_output_directory} -write_sim_files
```

```
compile_ultra -no_autoungroup
```


Latest Specification is available here:
http://bit.do/xupf

Software-managed Power Management Example in UPF

- **Example:** RISC-V SoC with a PMC managing a counter. PMC managed by Software (APB bus).



Latest Specification is available here:
http://bit.do/xupf

Software-managed Power Management Example in UPF

- **Example:** VexRiscV SoC with a PMC managing a counter. PMC managed by Software (APB bus).

Managed Devices

```
create_pmc PMC -domain TOP_PD -devices {PG_SW}
    -clock {io_mainClk posedge} -reset {io_asyncReset async high}
    -interface apb3
    -interface_param {addr_width 20}
    -interface_param {data_width 32}
    -interface_param {base_address 0xf0030000}
    -interface_param {use_slave_error false}
```

Bus Interface

```
add_power_state TOP_PD/PMC \
    -state {ALL_ON -logic_expr {PG_SS == ON_STATE}} \
    -state {ALL_OFF -logic_expr {PG_SS == OFF_STATE}}
set_initial_state TOP_PD/PMC -state ALL_ON
describe_state_transition -object TOP_PD/PMC \
    PSM_transition0 -from ALL_ON -to ALL_OFF
describe_state_transition -object TOP_PD/PMC \
    PSM_transition1 -from ALL_OFF -to ALL_ON
```


Latest Specification is available here:
http://bit.do/xupf

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Software-managed Power Management Example in UPF

- **Example:** VexRiscV SoC with a PMC managing a counter. PMC managed by Software (APB bus).

```
# Isolation managed by PMC
set_isolation PG_ISO -domain PG_PD -clamp_value 0 \
    -isolation_sense high \
    -isolation_signal {TOP_PD/PMC}
# Retention managed by PMC
set_retention PG_RET -domain PG_PD \
    -save_signal {TOP_PD/PMC low} \
    -restore_signal {TOP_PD/PMC high}
```

Controls driven by PMC!

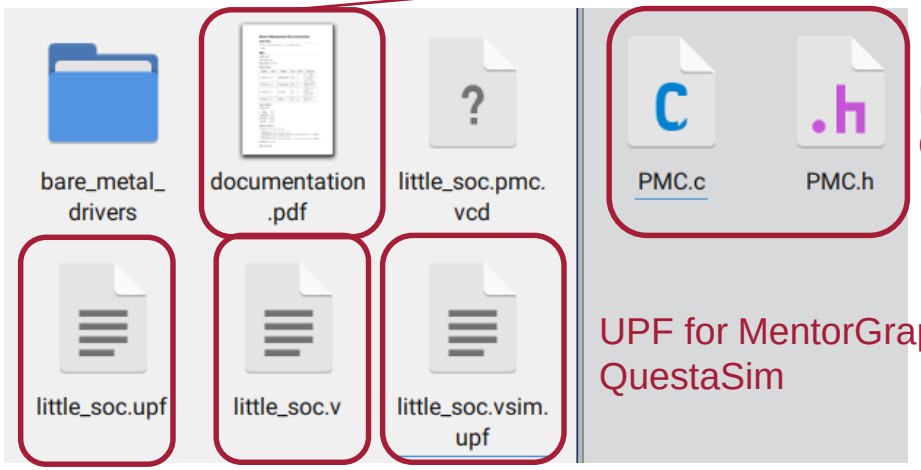
```
# Need to activate strategies when power switch is off!
schedule_strategy TOP_PD/PMC -state PG_SW.PG_SW_OFF \
    -strategy PG_PD/PG_ISO
    -strategy PG_PD/PG_RET
    -strategy PG_PD/PG_CG
    -strategy PG_PD/PG_DR
```

Strategies to enable during power-off!

Latest Specification is available here:
http://bit.do/xupf

Software-managed Power Management Example in UPF

- Results after PMC synthesis:



Bare-Metal C Drivers

UPF for MentorGraphics QuestaSim

UPF for Synopsys Design Compiler

RTL Design with PMC implementation

Overview

The following power management controllers have been created:

- /PMC

PMC

Location: /PMC

Bus Interface: APB3

Base Address: 0xf0030000

Memory Map

Address	Offset	Register	Type	Width	Description
0xf0030000	0x0	RUNNING_MODE	read-write	2	Running PSM mode. Overwrite to trigger a mode change.
0xf0030004	0x4	TARGET_MODE	read-write	2	New target PSM mode.
0xf0030008	0x8	IS_STABLE	read-only	1	Checks if the PSM is in a stable state (1) or if not (0).
0xf003000c	0xc	ERROR	read-only	32	Last occured error.

Power States

State Bit Width: 2

State	ID
[UNDEFINED]	0x0
ALL_OFF	0x1
ALL_ON	0x2

Software Driver

The software drivers are available here:

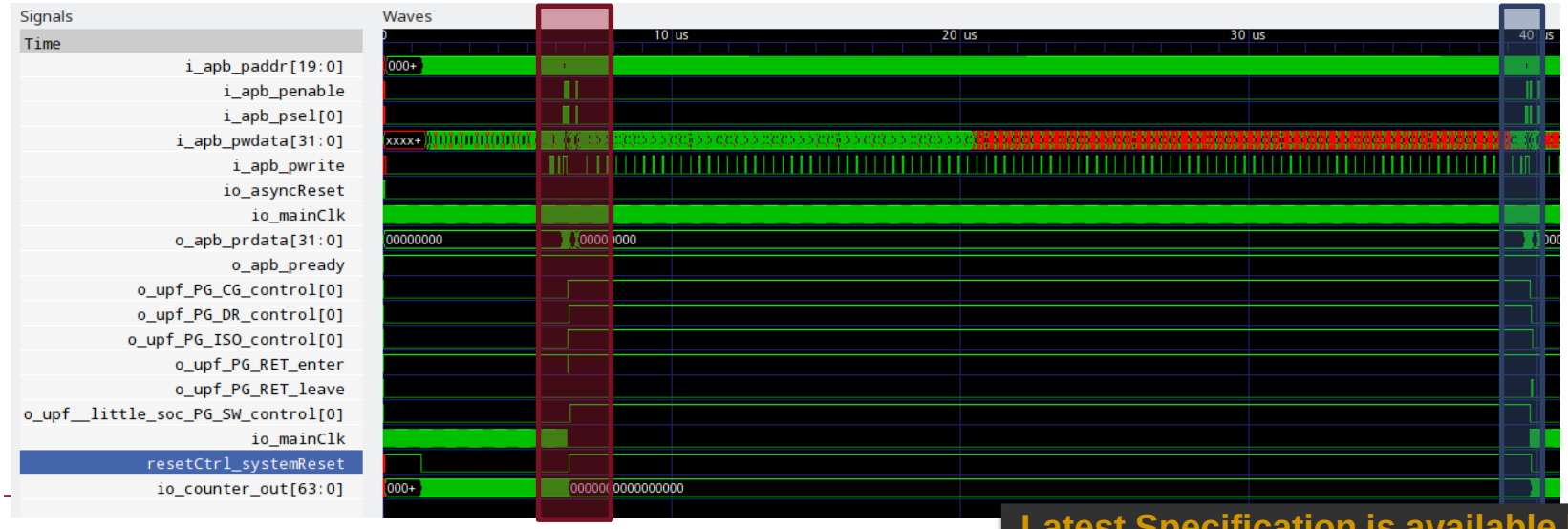
Latest Specification is available here:
http://bit.do/xupf

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Software-managed Power Management Example in UPF

```c
#include <stdint.h>

#include <murax.h>
#include "../../../../syn/output/bare_metal_drivers/PMC.h"

void main() {
    int i = 0;
    PMC_switch_state_b(PMC_ALL_OFF);
    for(i=0; i < 100; i++);
    PMC_switch_state_b(PMC_ALL_ON);
    for(i=0; i < 100; i++);
    PMC_switch_state_b(PMC_ALL_OFF);
    for(i=0; i < 100; i++);

    // Caution: Please don't use Timer B in any code and do
    // not remove this code.

    // The following code is for the testbench which checks that timer B
    // will reach the limit (io_full will go to high).
    prescaler_init(TIMER_PRESCALER);
    TIMER_PRESCALER->LIMIT = 1;
    timer_init(TIMER_B);
    TIMER_B->LIMIT = 8;
    TIMER_B->CLEARS_TICKS = 0x00010002;
    while(1);
}
```

Software uses automatically generated bare-metal drivers!



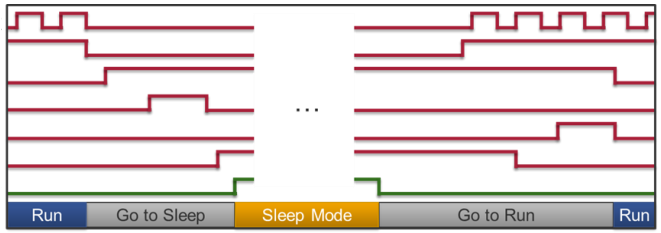
Latest Specification is available here:
http://bit.do/xupf

Summary and Roadmap

Summary:

- UPF is currently not able to capture all aspects of dynamic power management techniques.
- xUPF might be a good hint what we need in future UPF revisions!
- Synthesis tool for xUPF is already available.

```
create_power_domain PD_2 -elements {/core/exec/alu/muldiv} \
    -supply {primary set_primary_PD_2} \
    -supply {default_isolation set_primary_TOP_VA} \
    -supply {default_retention set_primary_TOP_VA}
create_power_switch switch_0_PD_2 -output_supply_port \
    {supply_out net_vvdd_PD_2} -input_supply_port \
    {supply_in net_vdd_TOP_VA} -control_port {control_in} \
    -on_state {IS_ON supply_in {!control_in}} \
    -off_state {IS_OFF {control_in}}
add_power_state -domain PD_2 \
    -state {IS_ON -logic_expr {set_primary_PD_2 == IS_ON}}
add_power_state -domain PD_2 \
    -state {IS_OFF -logic_expr {set_primary_PD_2 == IS_OFF}}
```



Roadmap:

- ASIC-readyness shall be proven in a tapeout in Q1/2020!
- Scala framework planned to be published open-source next year!
- Proposals to IEEE-1801 committee…

Latest Specification is available here:
http://bit.do/xupf

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Thank you.

Dustin Peterson

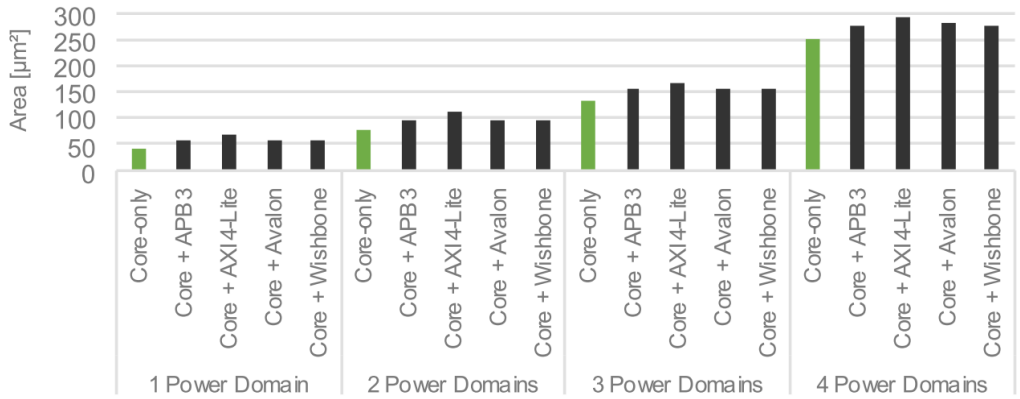
Eberhard Karls Universität Tübingen
Lehrstuhl für Eingebettete Systeme
Fon: +49 7071 - 29 – 75458
Fax: +49 7071 - 29 – 5062
E-Mail: dustin.peterson@uni-tuebingen.de

Latest Specification is available here:
http://bit.do/xupf

BACKUP

Power Management Synthesis

- Implementation evaluated with Synopsys 32nm EDK and GlobalFoundries 22FDX + Invecas 8-Track RVT Library.



Area Results for 22FDX and differently sized power designs with multiple fully-meshed power state machines

- Software-controlled PMC has been integrated into Pulpino, managing a timing tracer and multiple SRAM blocks.
- Our implementation supports dynamic body biasing, so we plan to manage the body bias by our PMC in the tape-out, too.
- Integration of xUPF into power design exploration under development!

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Software-managed Power Management Example in UPF

- **Example:** VexRiscV SoC with a PMC managing a counter. PMC managed by Software (APB bus).

Basic Power Management Specification

```
create_pmc PMC -domain TOP_PD -devices {PG_SW} \
    -clock {io_mainClk posedge} -reset {io_asyncReset async high} \
    -interface apb3 -interface_param {addr_width 20} -interface_param {data_width 32} \
    -interface_param {use_slave_error false} -interface_param {base_address 0xf0030000}
```

Managed devices

Bus Interface

Power State Machine

```
add_power_state TOP_PD/PMC \
    -state {ALL_ON -logic_expr {PG_SS == ON_STATE}} \
    -state {ALL_OFF -logic_expr {PG_SS == OFF_STATE}}
set_initial_state TOP_PD/PMC -state ALL_ON
describe_state_transition -object TOP_PD/PMC \
    PSM_transition0 -from ALL_ON -to ALL_OFF
describe_state_transition -object TOP_PD/PMC \
    PSM_transition1 -from ALL_OFF -to ALL_ON
```

Connect PMC to Bus Interconnect

```
connect_logic_net apb3Router_1/io_outputs_3_PADDR \
    -ports TOP_PD/PMC/i_apb_paddr -reconnect
connect_logic_net apb3Router_1/io_outputs_3_PSEL \
    -ports TOP_PD/PMC/i_apb_psel -reconnect
connect_logic_net apb3Router_1/io_outputs_3_PENABLE \
    -ports TOP_PD/PMC/i_apb_penable -reconnect
connect_logic_net apb3Router_1/io_outputs_3_PREADY \
    -ports TOP_PD/PMC/o_apb_pready -reconnect
connect_logic_net apb3Router_1/io_outputs_3_PWRITE \
    -ports TOP_PD/PMC/i_apb_pwrite -reconnect
connect_logic_net apb3Router_1/io_outputs_3_PWDATA \
    -ports TOP_PD/PMC/i_apb_pwdata -reconnect
connect_logic_net apb3Router_1/io_outputs_3_PRDATA \
    -ports TOP_PD/PMC/o_apb_prdata -reconnect
```

Managed Strategies

```
# Isolation managed by PMC
set_isolation PG_ISO -domain PG_PD -clamp_value 0 \
    -isolation_sense high \
    -isolation_signal {TOP_PD/PMC}
# Retention managed by PMC
set_retention PG_RET -domain PG_PD \
    -save_signal {TOP_PD/PMC low} \
    -restore_signal {TOP_PD/PMC high}
# Clock Gating managed by PMC
set_clock_gating PG_CG -domain PG_PD \
    -type and -location inbound -clock_sense posedge \
    -control {TOP_PD/PMC}
# Power Domain Reset managed by PMC
set_domain_reset PG_DR -domain PG_PD \
    -resets {u_big_block/resetCtrl_systemReset} \
    -location inbound -reset_type {async high} \
    -control {TOP_PD/PMC}

# Need to activate strategies when power switch is off!
schedule_strategy TOP_PD/PMC -state PG_SW.PG_SW_OFF \
    -strategy PG_PD/PG_ISO
    -strategy PG_PD/PG_RET
    -strategy PG_PD/PG_CG
    -strategy PG_PD/PG_DR
```

Latest Specification is available here:
http://bit.do/xupf

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Power Management Synthesis

Power Management Synthesis

RTL Design (power-agnostic)

Extended UPF

(1) Model Generation

Supply Network Model

Power State Machine Model

Power Management Controller Configuration

(2) Feature Scheduling

Annotated Power State Machine Model

(3) IP Generator

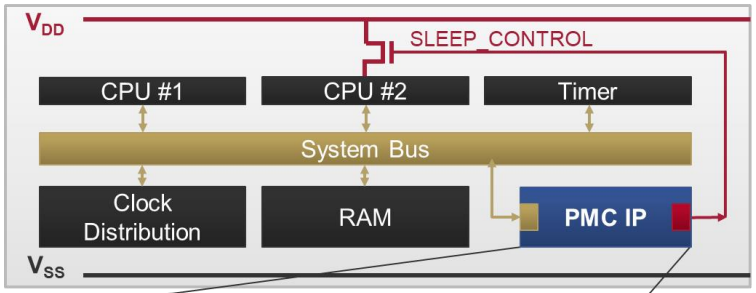
Power Management Controller IP (Verilog, VHDL, SystemC)

(4) IP Integrator

RTL Design (power-aware)

Standard-Compliant UPF

- Instantion and Connection of PMC IP
- Synthesis of Clock Gating, Domain Reset and Stall Propagation Logic
- Rewriting of UPF Script



Interconnected! automatically

Configured in UPF, generated automatically in Verilog***

Power Managed SoC Design

*** Currently supported and generated interfaces: AXI4-Lite, APB3, Avalon-MM, Wishbone

Latest Specification is available here:
http://bit.do/xupf