TC2
Design for Power Gating –
and what UPF can and cannot do for you

David Flynn

David.Flynn@arm.com – ARM Fellow, R&D

David.Flynn@ecs.soton.ac.uk – Visiting Professor

Purpose of this user session

Propose a basic “best practice” for power gating:

- Now that Synopsys EDA tools are fully multi-voltage aware…
- … and Unified Power Format, UPF1.0, ‘power intent’ support is in place

Need to understand how

- UPF infers enhancements onto verified ‘golden RTL’

And therefore

- Develop a disciplined sub-set of UPF constructs and understand the interaction with RTL design
  - notably clocks, resets and retention state

Outline

1. The basics of power gating for the RTL designer
2. UPF “intent” for power gating and interface clamping
3. Practical power gating for safe turn-on and testability
4. The problems managing resets, clocks/clock gating
5. Complications of state retention with power gating
6. Bringing it all together with a worked example
7. Conclusions
   - Best-practice approaches
   - Pitfalls to avoid.

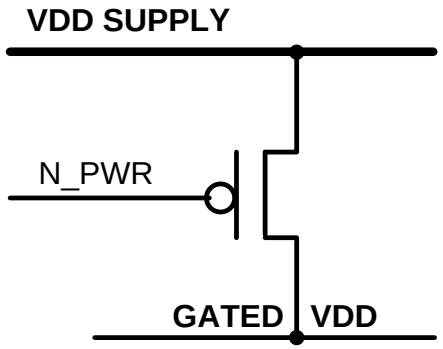
Power gating basics

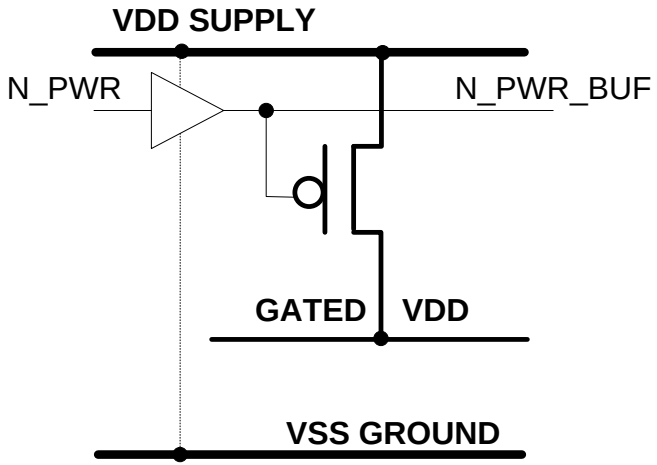
- Given that product “standby” life is critical to many designs, especially where battery powered…
- … and that at 90nm and below, CMOS technology leakage power grows in significance
- Power Gating is one key leakage mitigation technique
  - “MTCMOS” introduces (High-Vth) transistor switches in series with the power rail(s) with conventional circuits (built up of one or more High/Standard/Low Vth cells)
  - But introduces the complication that outputs float/collapse when the circuit is power gated off
  - And register state of course is lost, so will need re-initializing or saving and restoring…

Power gates

- Power gates are high-current transistor switches placed in series with the standard power supply or ground.
- “Header” PMOS switches shown
  - Basic un-buffered switch
  - Internally buffered switch
- “Footer” NMOS switch alternatives
  - Unbuffered/buffered switches to VSS
- Typically many switches in parallel

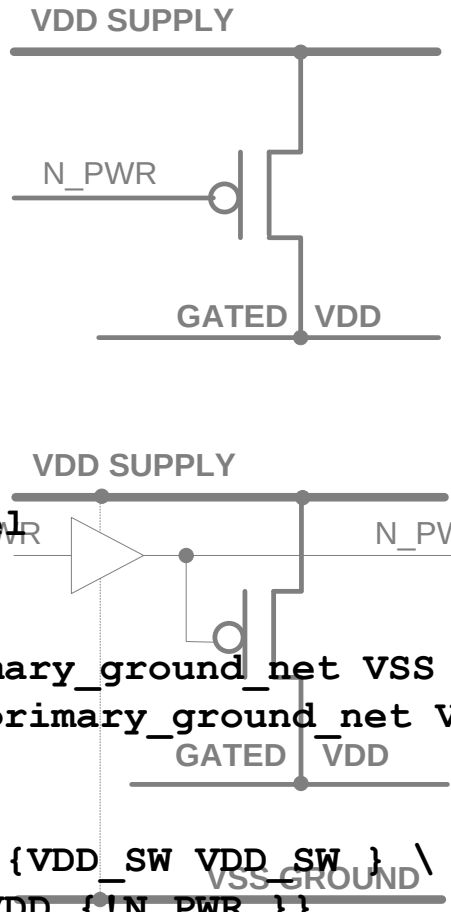
Good “off” leakage – but “on” IR drop





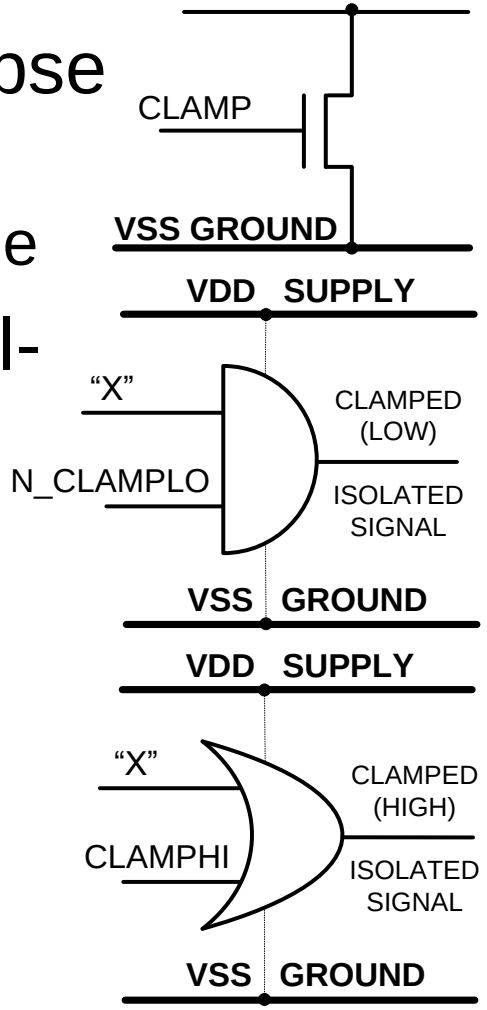
UPF and Power gates

```
create_power_domain TOP
create_power_domain PD -elements { uBlock }

create_supply_port  VDD -domain TOP
create_supply_net   VDD -domain TOP
connect_supply_net  VDD -ports VDD

create_supply_port  VSS -domain TOP
create_supply_net   VSS -domain TOP
connect_supply_net  VSS -ports VSS

create_supply_net   VDD_SW -domain PD -resolve parallel
create_supply_net   VSS -domain PD -reuse

set_domain_supply_net TOP -primary_power_net VDD -primary_ground_net VSS
set_domain_supply_net PD  -primary_power_net VDD_SW -primary_ground_net VSS

create_power_switch PG -domain PD \
    -input_supply_port {VDD VDD } -output_supply_port {VDD_SW VDD_SW } \
    -control_port {N_PWR N_PWR } -on_state {on_state VDD {!N_PWR }}
map_power_switch PG -domain PD -lib_cell HEADBUF16_X1M_A12TL
```



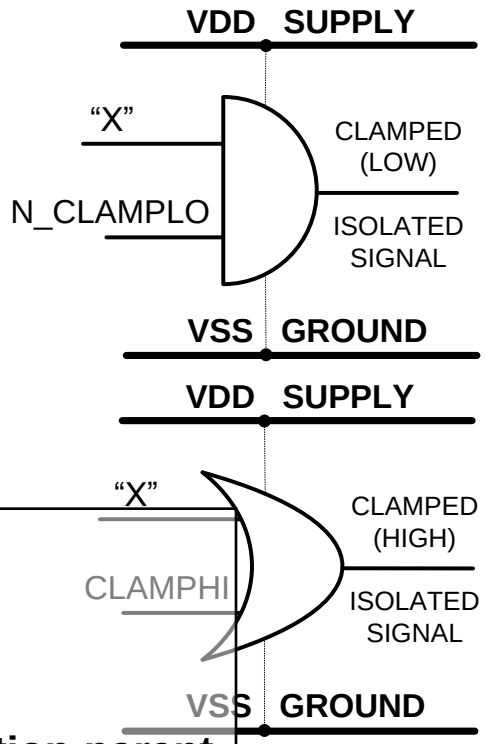
Output Clamping

- Outputs of power-gated blocks collapse
  - To a voltage where leakage through off switch is equal to residual circuit leakage
- For header switched systems a "pull-down" transistor is the minimum
- To avoid "multiple" driver issues special CLAMP cells are introduced to protect downstream inputs. e.g.
- CLAMPLO (AND-style function)
- CLAMPHI (OR-style function)



UPF and Output Clamping

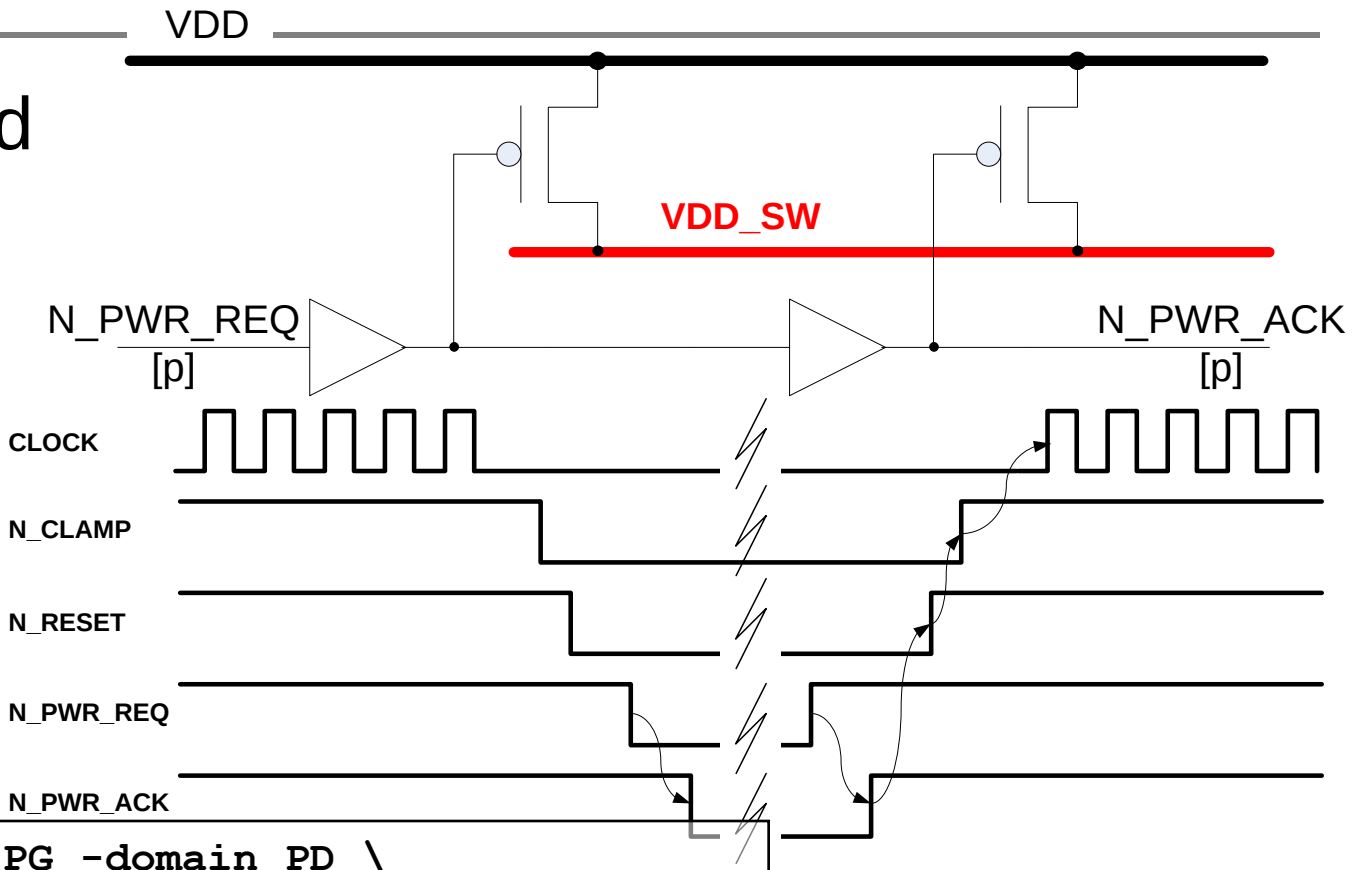
- For protocol-sensitive output signals the clamped values need to be architected
  - No longer in the RTL
  - An example of where UPF changes may break system verification
- UPF standard chooses isolation as the name for output clamping




```
set_isolation vsoc -domain PD -isolation_power_net VDD \
   -isolation_ground_net VSS \
   -clamp_value 0 -applies_to outputs
set_isolation_control vsoc –domain PD \
   -isolation_signal N_CLAMPLO  -isolation_sense low -location parent
```

UPF & Power Gating chains

- Recommended “chained” request / ack switch control scheme
- Helps “phase” turn-on, and testability




```
create_power_switch PG -domain PD \
    -input_supply_port {VDD VDD } \
    -output_supply_port {VDD_SW VDD_SW } \
    -control_port {N_PWR N_PWR } \
    -ack_port {N_PWR_ACK N_PWR_ACK {N_PWR_REQ}} \
    -on_state {on_state PD {!N_PWR }}
```

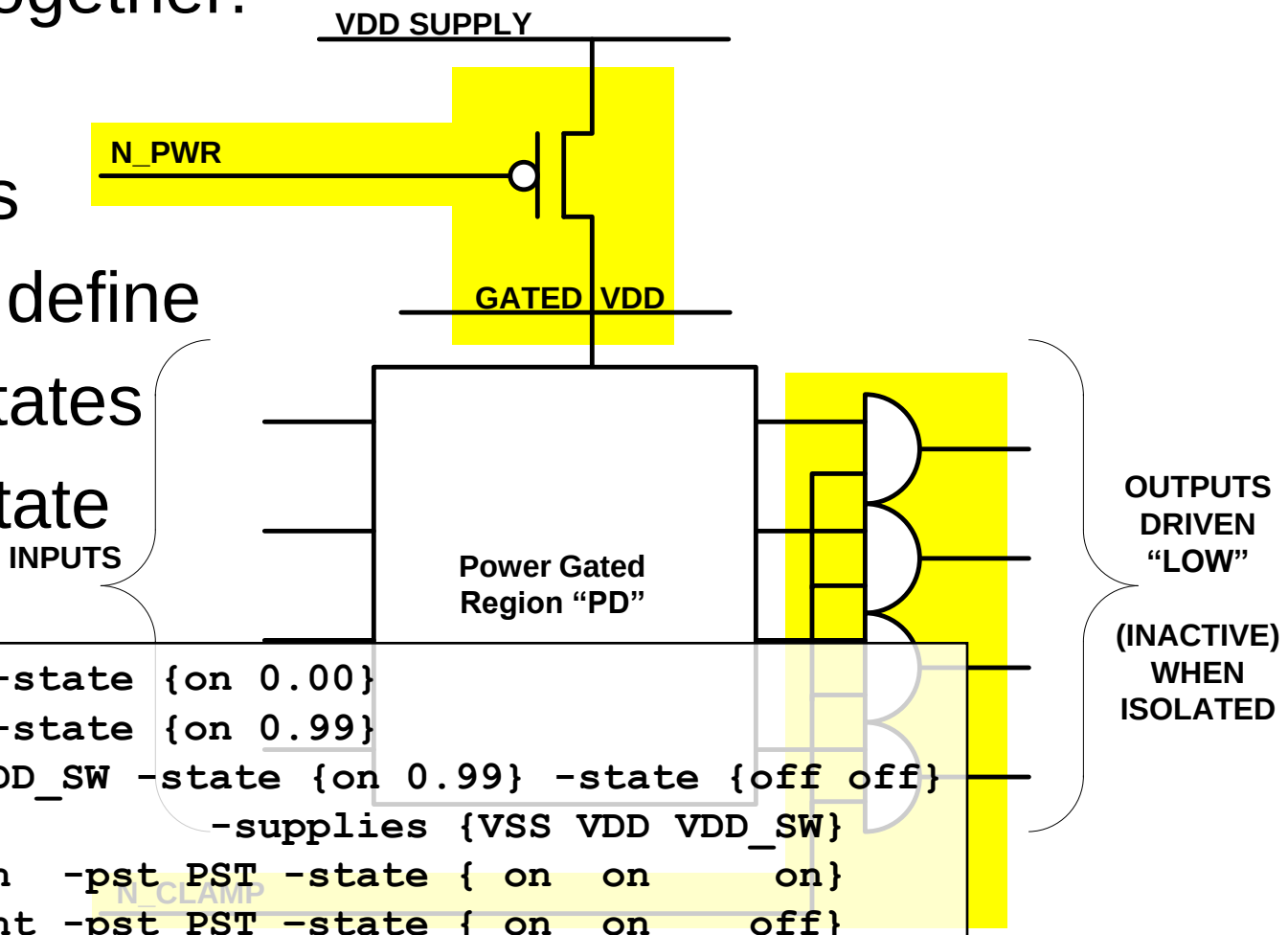
RTL+ UPF power gating

Putting this all together:

- Power gates
- Output clamps

Finally, need to define

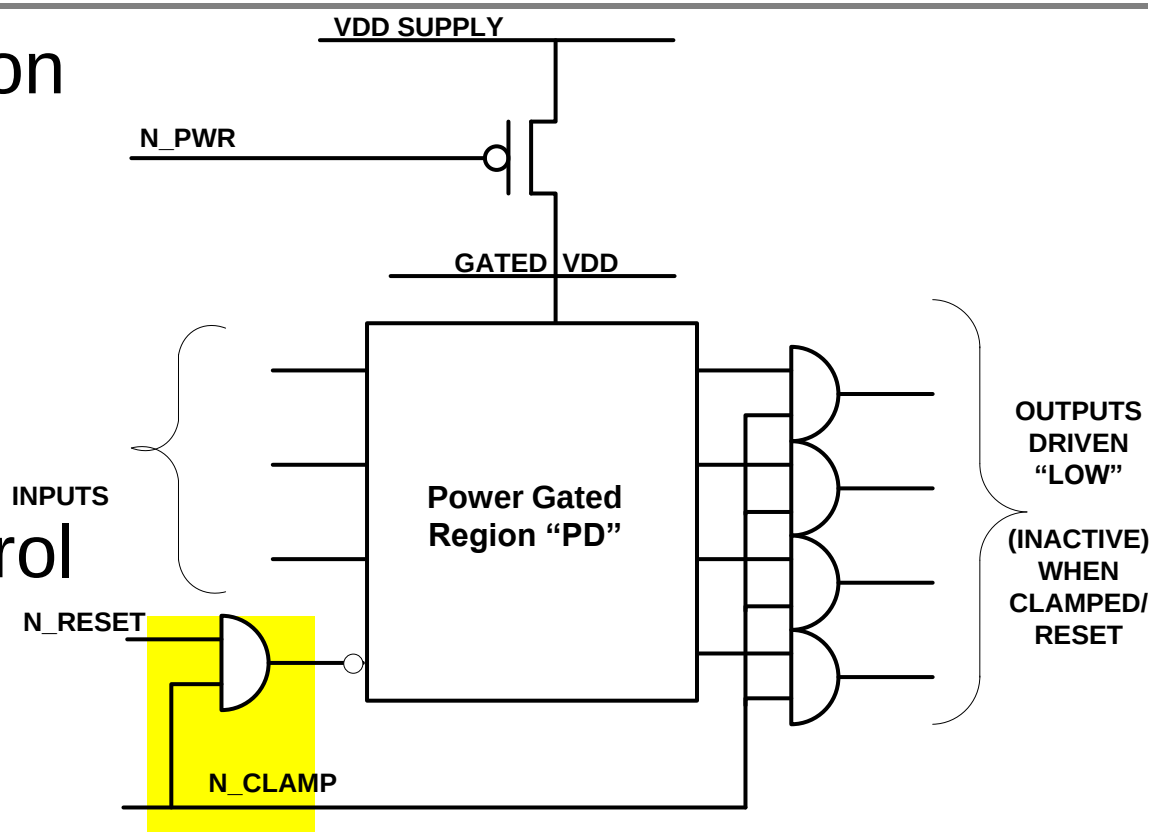
- Valid power states
- UPF Power State Table

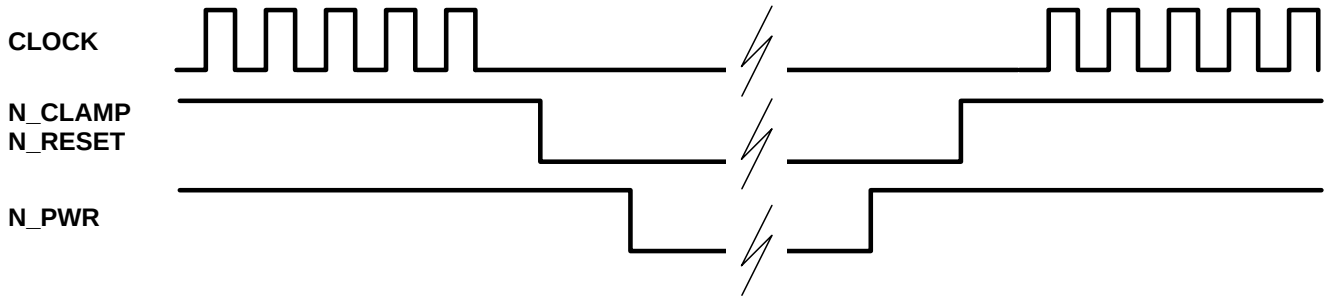



```
add_port_state VSS -state {on 0.00}
add_port_state VDD -state {on 0.99}
add_port_state PG/VDD_SW -state {on 0.99} -state {off off}
create_pst     PST                -supplies {VSS VDD VDD_SW}
add_pst_state fullon  -pst PST -state { on   on      on}
add_pst_state dormant -pst PST -state { on   on     off}
```

UPF and Resets

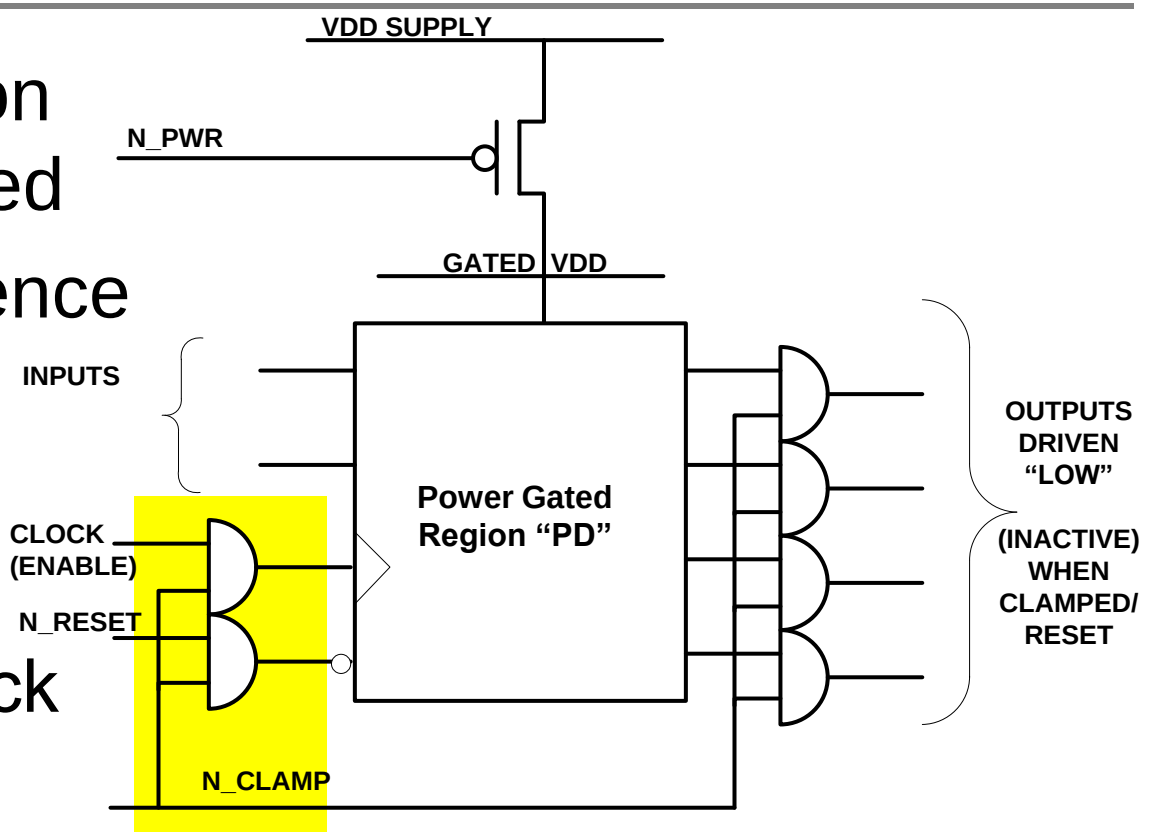
- A power gated region loses state
- Reset sequence required afterwards
- Ideally explicit individual RTL control
- "N_CLAMP" on asynch reset input needs care
- i.e. separate UPF clamps

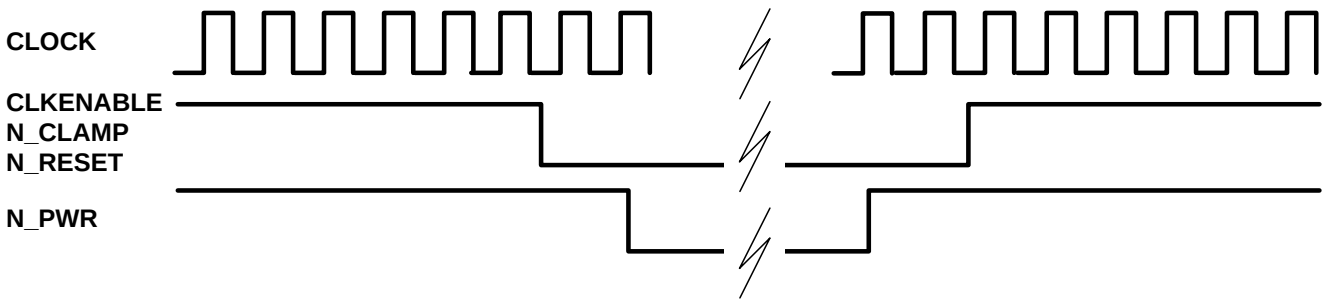




UPF and Clocks

- A power gated region should not be clocked
- Inhibit/enable sequence needs design/verify
- Explicit RTL control ideally
- “N_CLAMP” on clock enable needs (even more) care!





UPF and State Retention

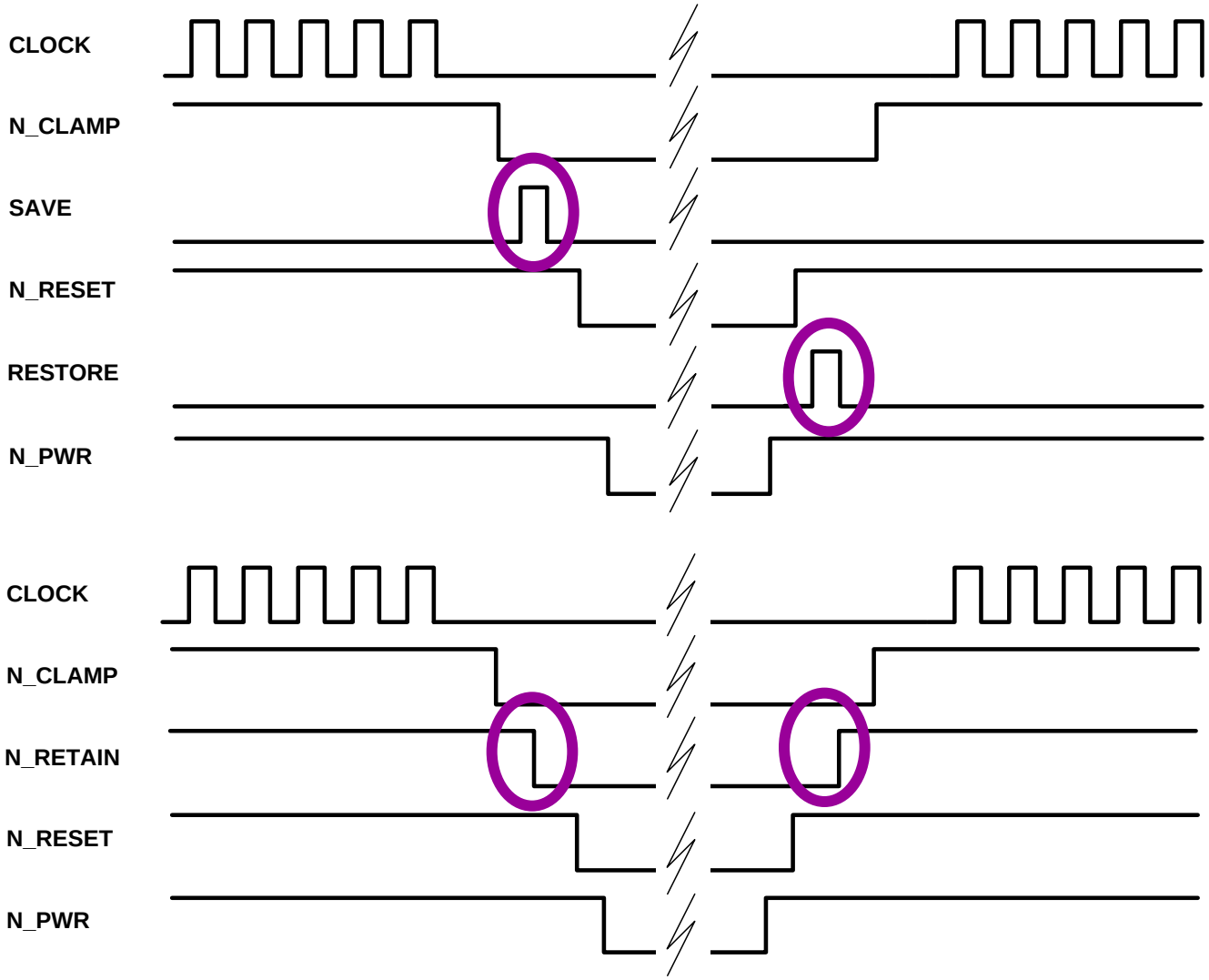
- UPF allows highly flexible (and dangerous) inference of retained state
- Safe options are:
  - ALL state
  - or NO state!
- Validation is key
  - As well as from reset (easy!)
  - Need to verify all pathological non-reset state interaction
  - To prove no deadlock etc.
- Clock gates as well
  - Explicit or inferred

```
set_retention RET -domain PD \
    -retention_power_net VDD \
    -retention_ground_net VSS
set_retention_control RET -domain PD \
    -save_signal {N_RETAIN low}
    -restore_signal {N_RETAIN high}
map_retention_cell RET -domain PD \
    -lib_cell_type DRFF
```

```
map_retention_cell RET -domain PD \
    -lib_cell_type RLAT \
    -elements {. . .}
```

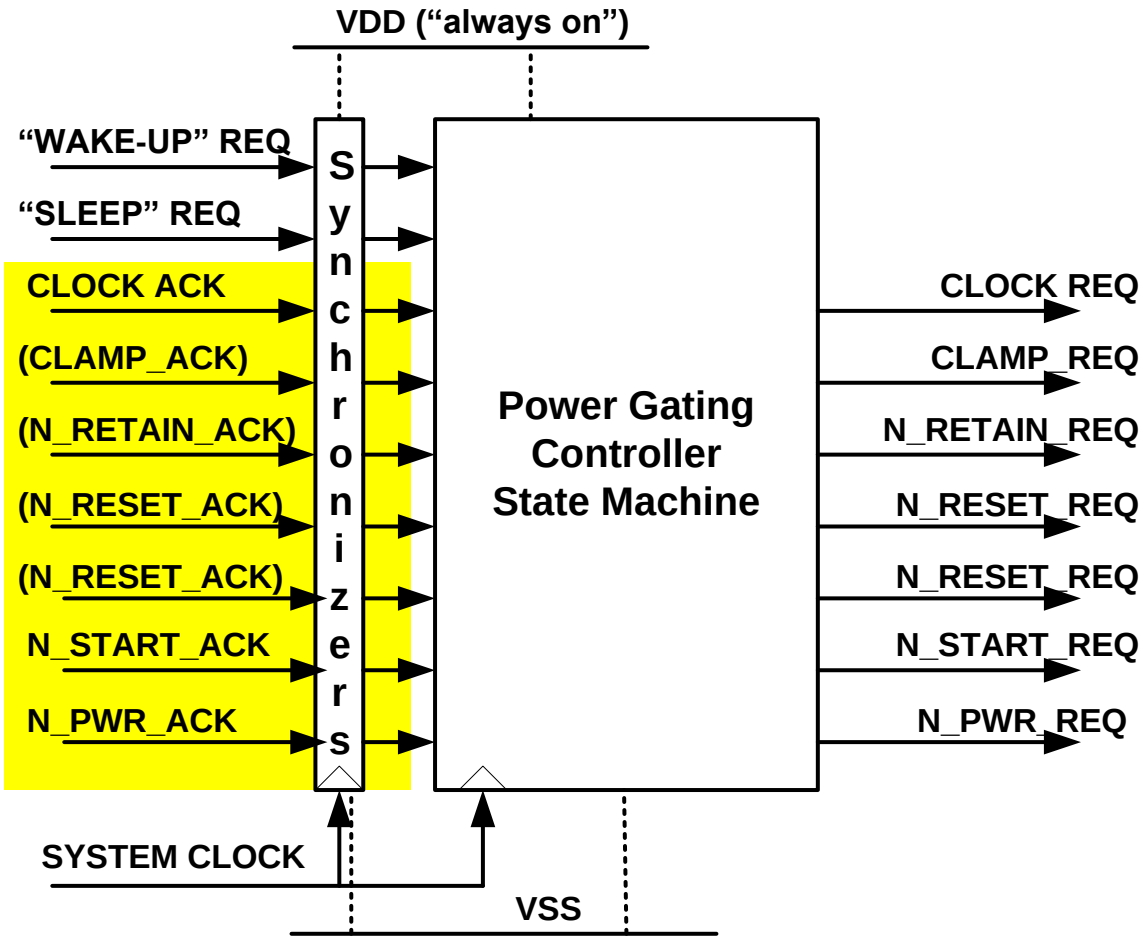
SRPG Sequencing

- SRPG needs explicit RTL control
- Depends on the RET FLOP style
  - Library



SRPG controller

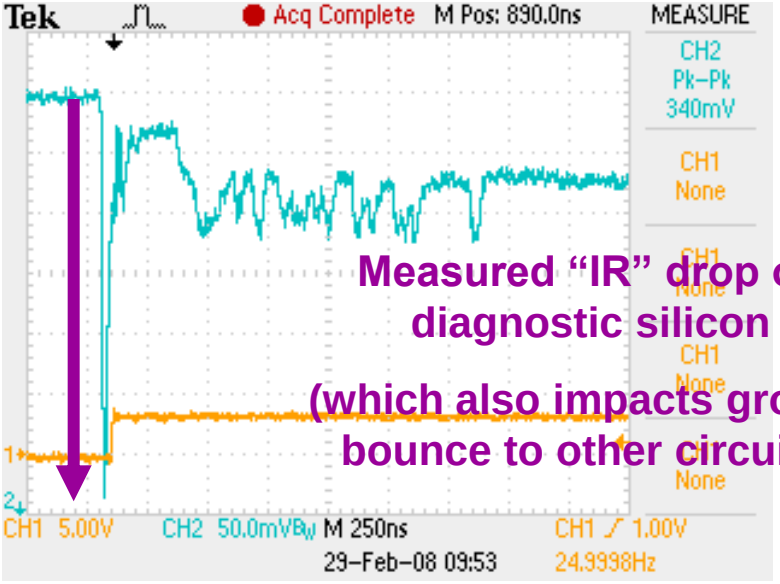
- RTL coded state machine
- Explicit drives to the UPF and RTL subsystem ports
- And to clock and reset controller
- For design-reuse across generations consider generic asynch handshakes



Multiple Power-Gate chains

- In order to safely avoid large power transients when turning on power gated regions consider “weak” and “strong” power gate chains (inferred in UPF)
- Measured benefits for TSMC 90nm G “SALT1” silicon

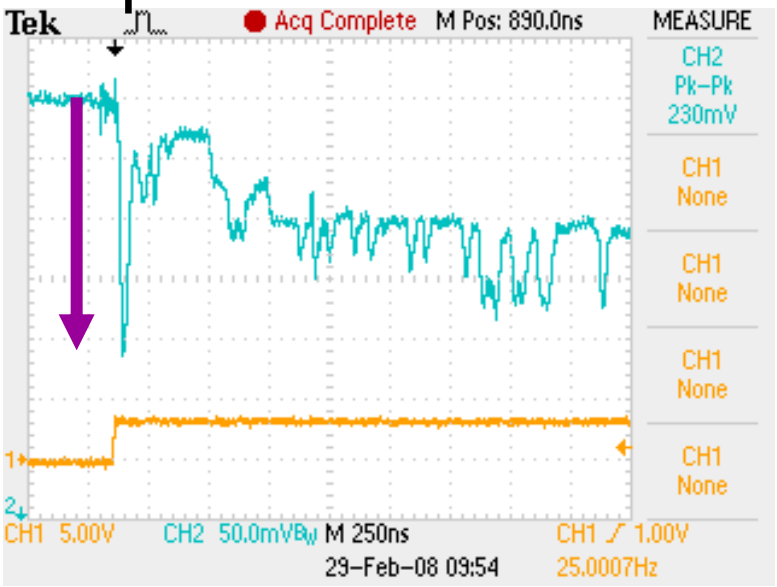
Forced main turn-on



Measured “IR” drop on diagnostic silicon

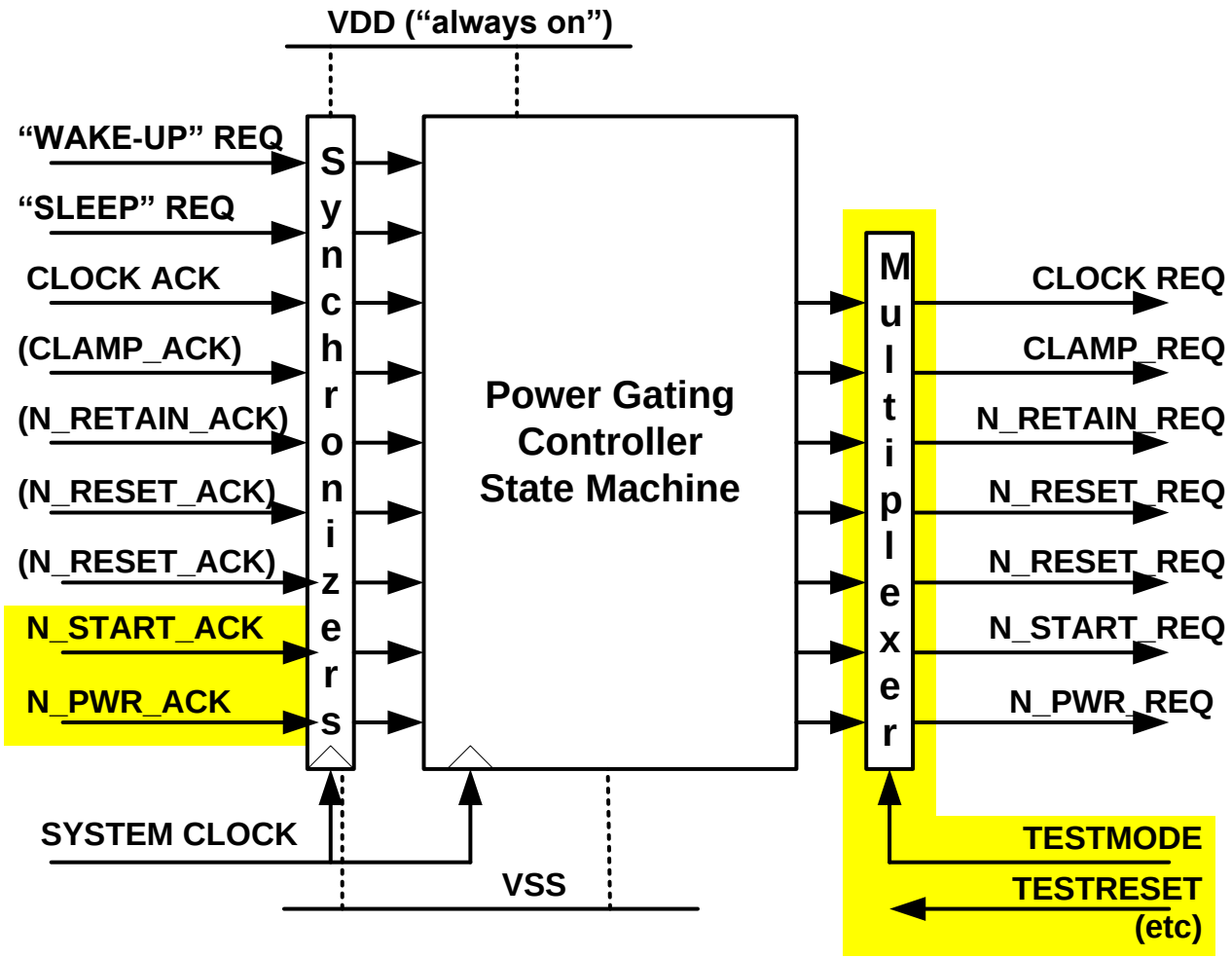
(which also impacts ground bounce to other circuits)

Sequenced start/main



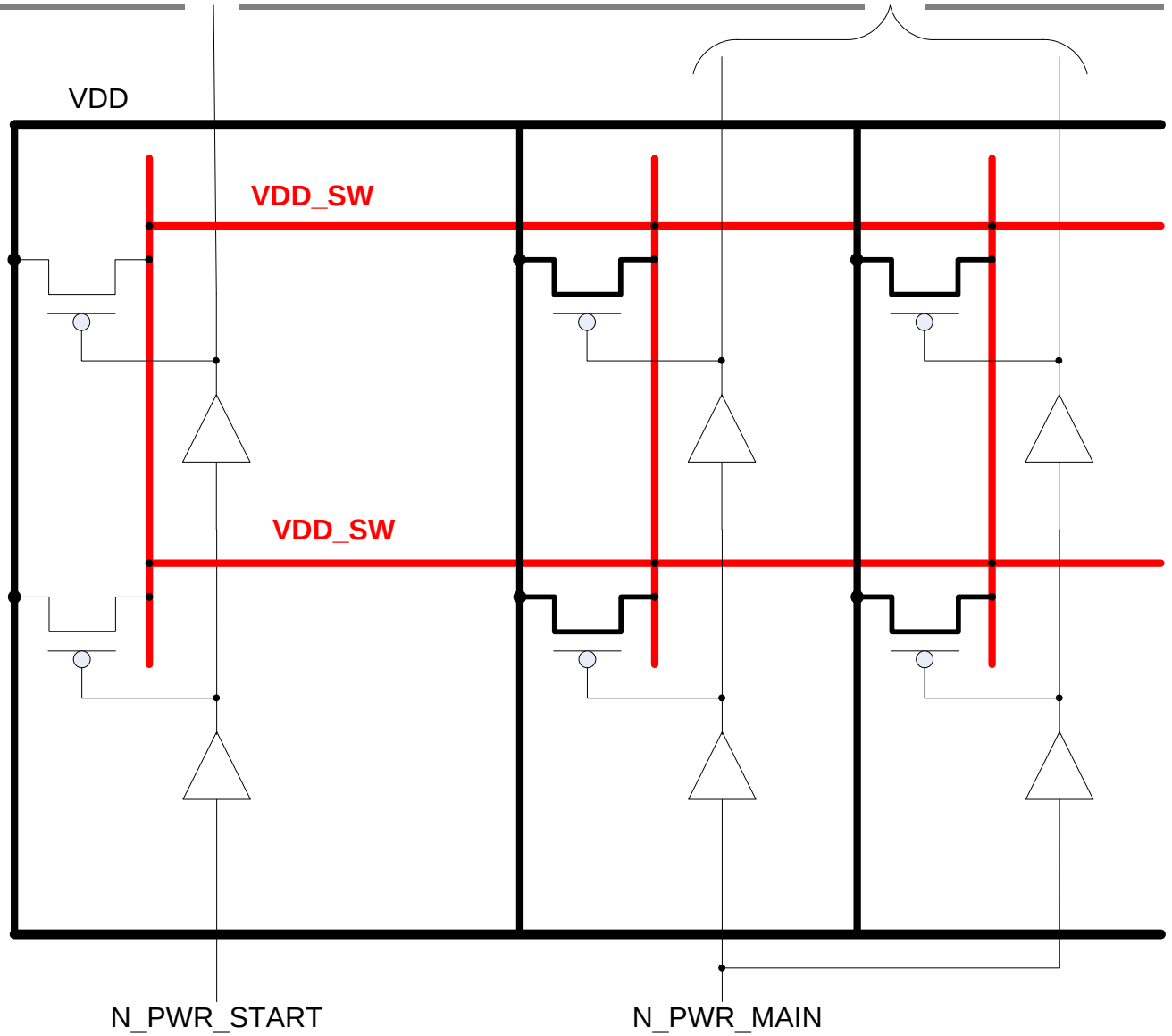
Do not forget testability

- Well understood that Clocks and resets are made RTL controllable
- Same true for retention and power gating controls
- PG “ACK” chains allow buffer chain integrity testing



Multi-chain virtual power grid

- When using distributed power gating need to ensure good switched rail grid
- Ensure good current sharing across switches
- Suit “soft start”



Conclusions – PG/UPF

- UPF can infer basic power gating, but design-time analysis is required to factor in appropriate power-up control strategies
- UPF infers basic interface output clamping in a straightforward manner but the designer needs to “architect” and specify the quiescent/safe values to be used
- How fast a power gated network can be turned-on requires attention to both the UPF inferred power gating structures and the RTL controller design
- Designing in handshake protocols across the UPF and RTL divide can facilitate design reuse and technology portability
- Power gating is not instantaneous or as transparent as clock-gating. Clocks cannot be restarted until power rails are safely stabilized and valid. Resets can either be handled explicitly or require careful asserting with UPF-inferred input clamping.

Conclusions – SRPG/UPF

- Arbitrary state retention inference through UPF is strongly to be discouraged
- For legacy IP, ‘total state’ or no state retention are the options that will verify and work
- UPF state integrity relies on not being compromised by power-gating switch-on transients
  - Not only of the block in question – but neighboring blocks on the SOC that share a common ground or power rail
- Single-edge clocking is strongly advocated otherwise clock gating constructs will have to infer retention latches for functional clock gating.

Questions?

Thank You

With thanks also to

- Alan Gibbons, Synopsys Inc, Reading UK
- John Biggs, James Myers, ARM Ltd, Cambridge UK
- Sachin Idgunji, Leah Schuth, ARM Inc, San Jose
- Electronics and Computer Science faculty
  Southampton University, UK