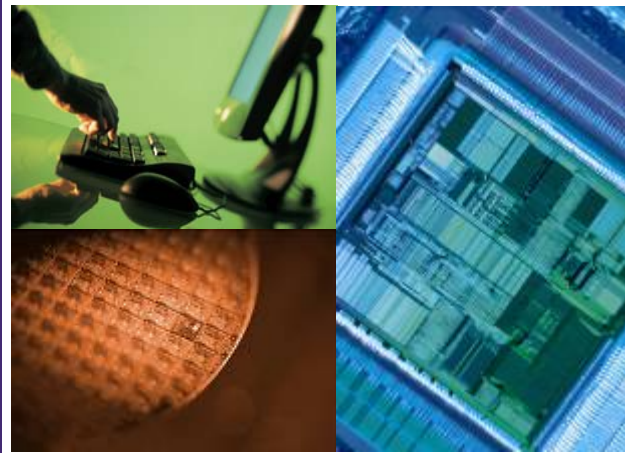


Liberty for Low Power Design



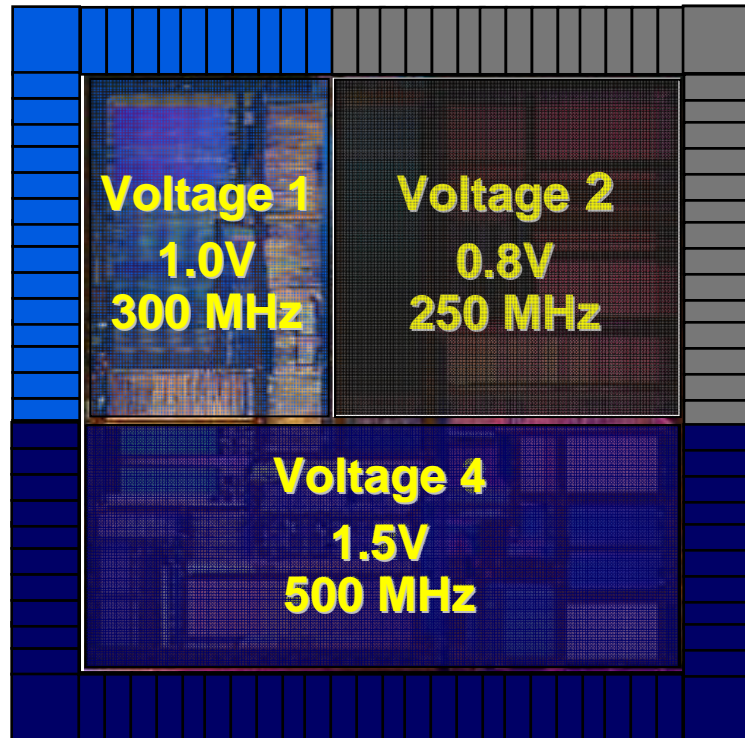
SYNOPSYS[®]
Predictable Success

Agenda

- What's needed for Low Power?
- New power capabilities
 - Approved by Liberty TAB
- Need more ?

Multi-Voltage Design

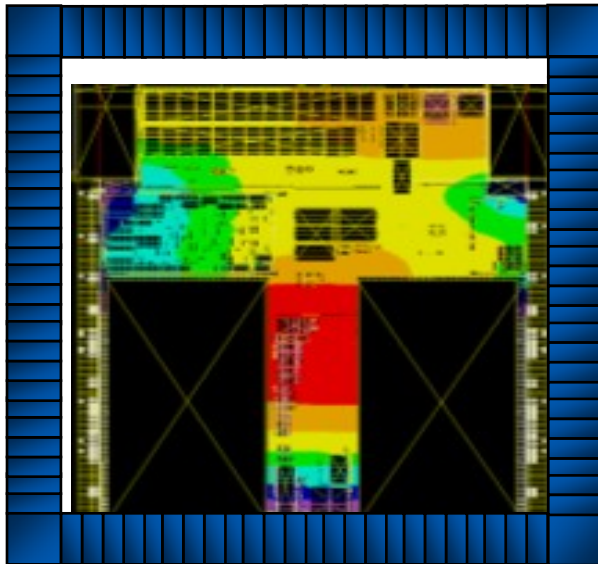
Dynamic and Leakage Power Reduction



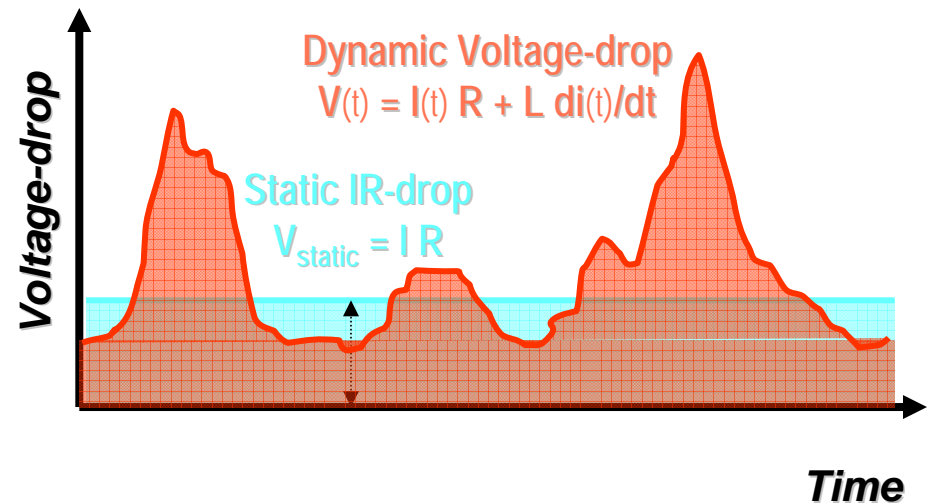
- Multiple Vdd designs are becoming popular
- Dynamic Voltage & Frequency Scaling is another technique

Libraries Need Delay and Power Models for a Wide Range of Vdd

Power Integrity

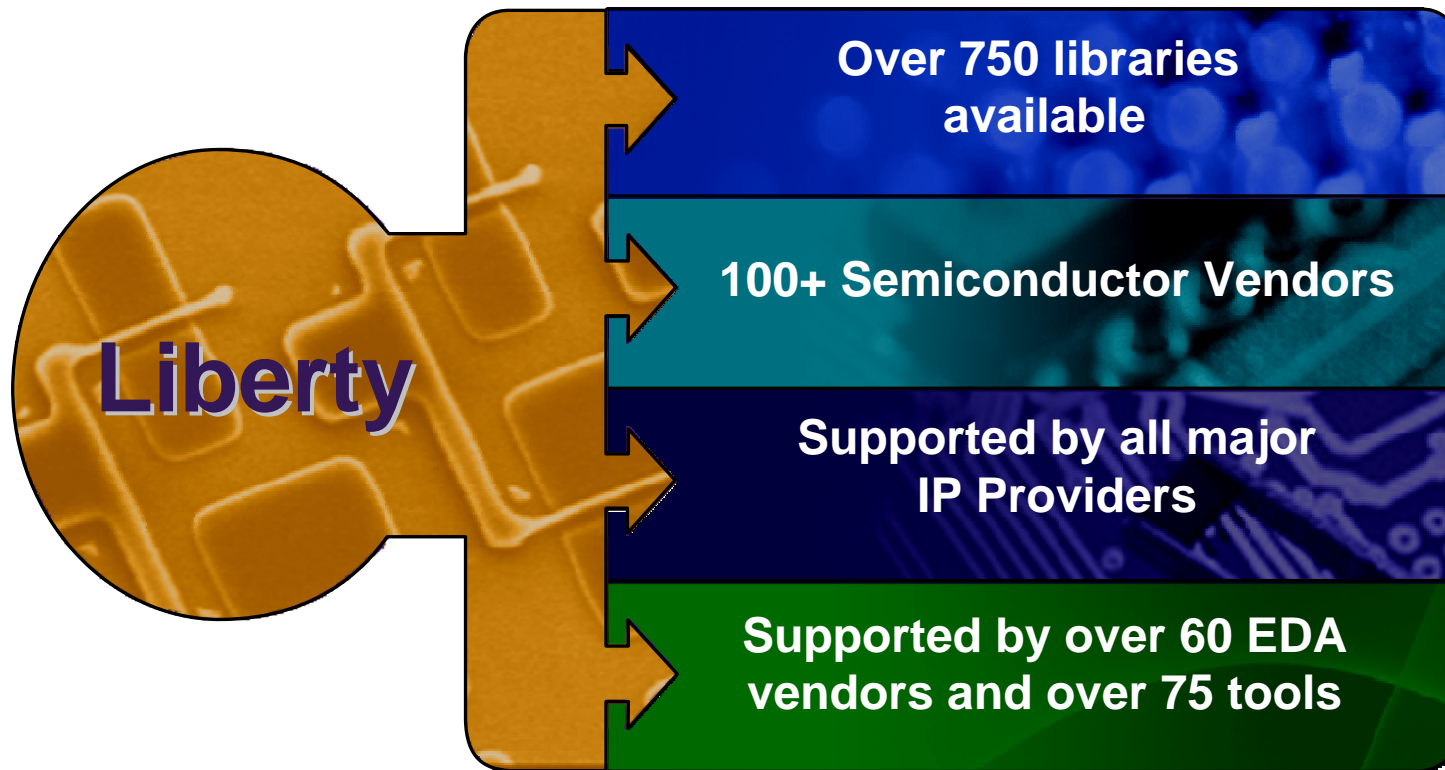


- Tools to compute:
 - Voltage drop on instance rail pins
 - Impact of voltage drop on delay



Libraries Need New Models For Power Integrity Analysis

Liberty™ is an Industry Standard



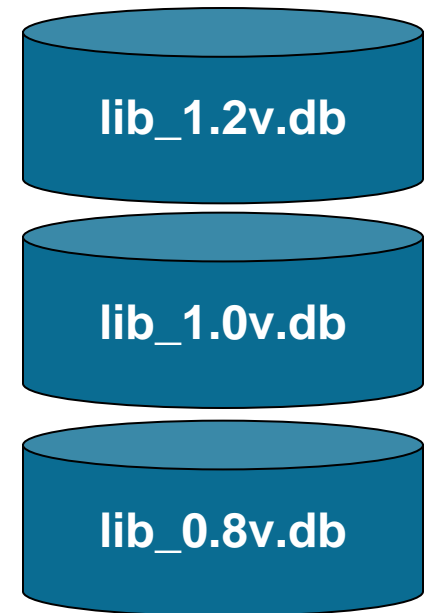
At the heart of Semiconductor Design

What's Needed for Low Power Design ?

- Accurate Modeling of Critical Power Effects
 - Voltage Scaling (timing, power, noise) – accurate across a broad range
 - Cell/rail currents plus equivalent parasitics
 - Gate leakage and channel leakage
 - Macro modeling capabilities
- Power / Ground (PG) pins
 - A place to attach rail/currents and equivalent parasitics
 - Automated power & control connection and checking, especially for special cells:
 - Level Shifters, Isolation, Retention
- Switch Cells
 - Coarse-grained and Fine-grained
 - DC I/V Curve Voltage Drop
 - Automated power & control connection and checking

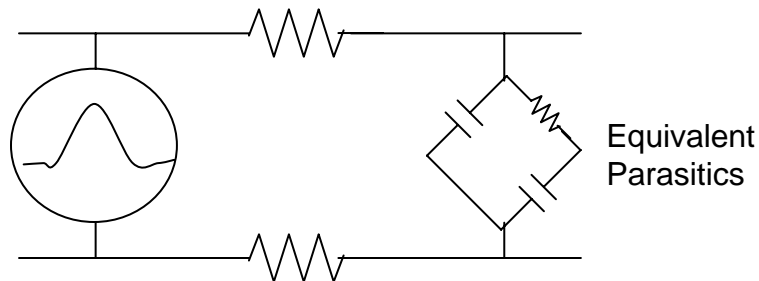
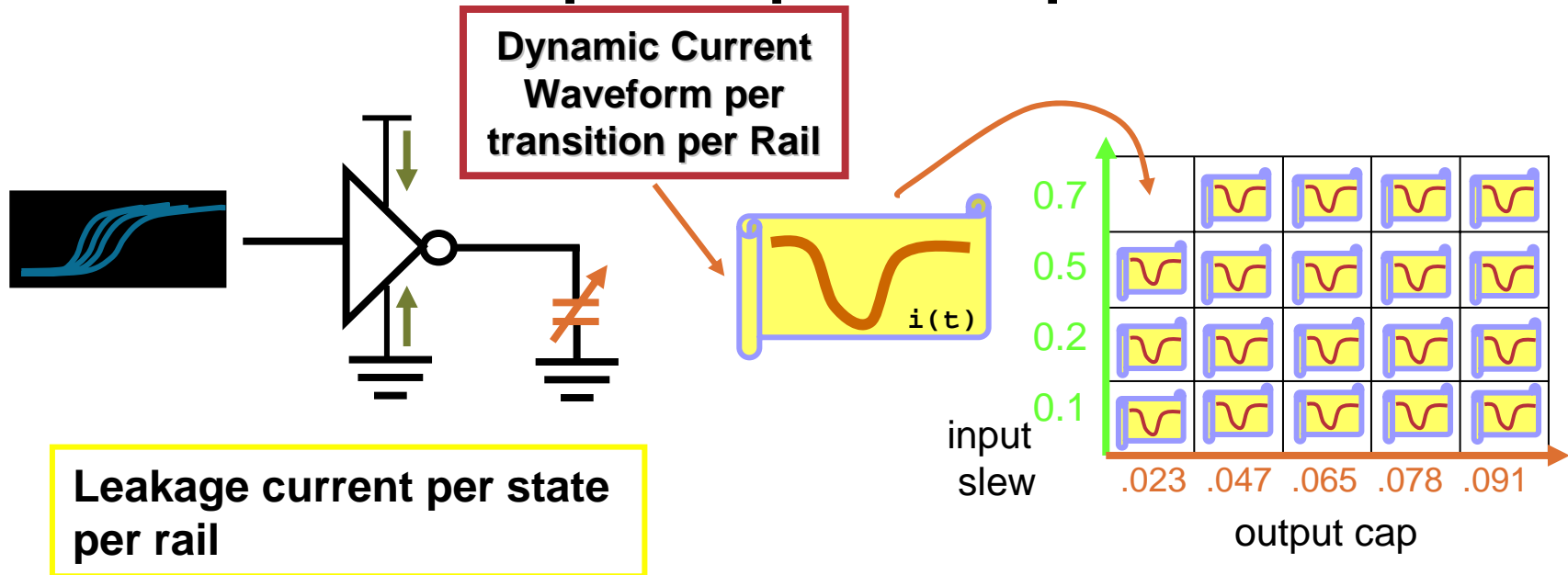
Accurate Modeling of Critical Power Effects: Voltage and Temperature Scaling

- **CCS enables high accuracy analysis for wide range of Voltages and Temperatures**
 - For power-aware design styles:
 - Single V_{dd}
 - Multiple V_{dd}
 - Dynamic voltage & frequency scaling (DVFS)
 - Accounting for IR Drop impact on delay
- **What is scaled:**
 - Driver model (timing)
 - Receiver model (timing)
 - Timing constraints:
 - setup, hold, recovery, removal, MPW
 - Noise models
 - Power analysis
- **Library Characterization is Straightforward**



**Separate CCS
Libraries**

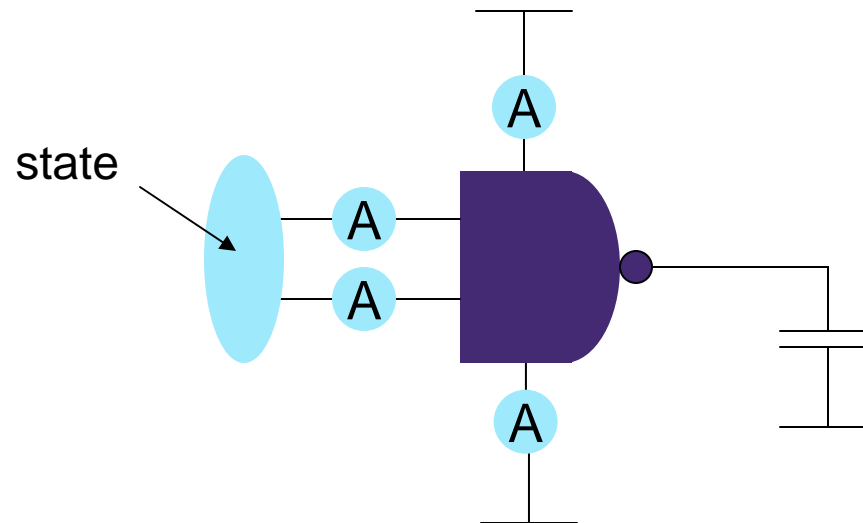
Accurate Modeling of Critical Power Effects: Cell/rail currents plus equivalent parasitics



Parasitics per state

Accurate Modeling of Critical Power Effects: Gate Leakage & Channel Leakage

- Channel leakage Modeling
 - Static current from source to drain
 - State dependent



- Gate leakage Modeling
 - Static current from driver cell to transistor gates of load cell
 - State dependent

Accurate Modeling of Critical Power Effects: Gate Leakage & Channel Leakage

```
cell(cell_name) {  
  ...  
  leakage_current() {  
    when : state_1;  
    pg_current(pg_pin_1) {  
      value : val_1_1;  
    }  
    ...  
    pg_current(pg_pin_n) {  
      value : val_1_n;  
    }  
    gate_leakage(input_1) {  
      input_low_value : val_low_1_1;  
      input_high_value : val_high_1_1;  
    }  
    ...  
  }  
  gate_leakage(input_n) {  
    input_low_value : val_low_1_n;  
    input_high_value : val_high_1_n;  
  }  
  leakage_current() {  
    when : state_2;  
    ...  
  }  
  leakage_current() {  
    /* default state */  
    ...  
  }  
}
```

2 levels of state dependency for Gate Leakage
– based on PG pin state and gate input

Accurate Modeling of Critical Power Effects: Macro modeling capabilities

- Asynchronous Operation Modeling

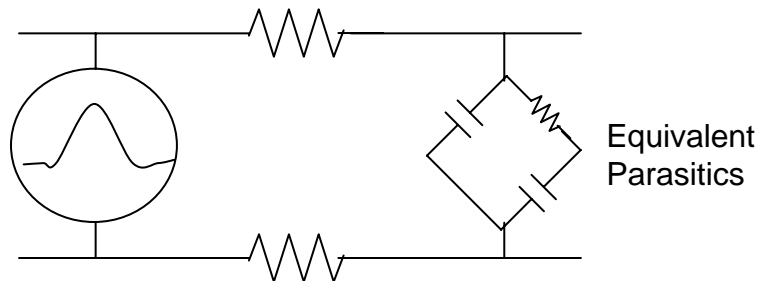
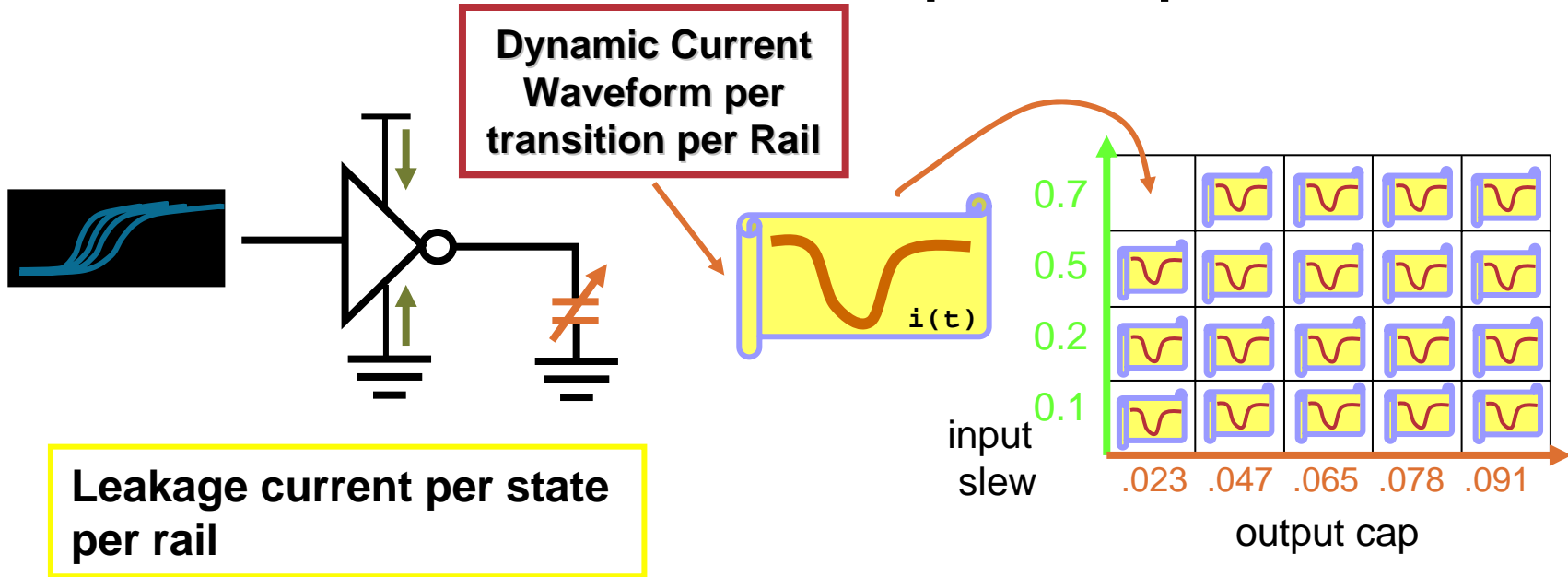
- Asynchronous Read access
- Model multi-input events

```
cell(cell_name) {  
    power_cell_type : macro;  
    dynamic_current(event_name) { /* the label is optional */  
        when : state_1;  
        related_inputs : related_input_list;  
        switching_group() {  
            input_switching_condition(input_sense);  
            min_input_switching_count: min_val;  
        }  
    }  
}
```

- Consolidated equivalent parasitic

Power/Ground (PG) Pin Support

Place to attach rail/currents and equivalent parasitics

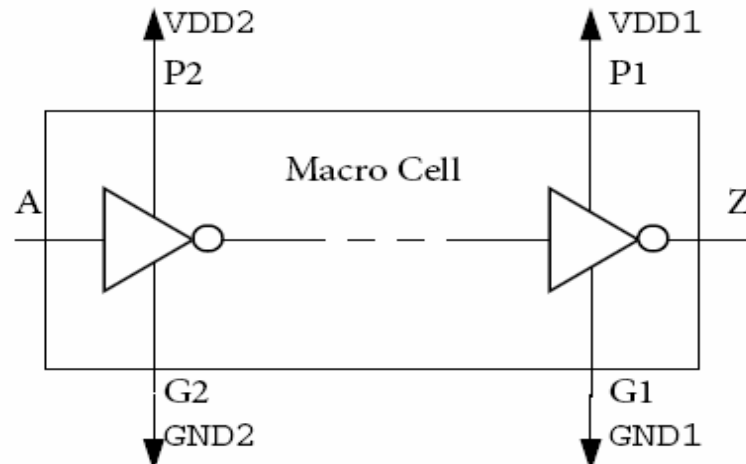


Parasitics per state

PG Pin Support Special Cells

- Provides support for real PG library pins where:
 - power pin means current source pin and
 - ground pin means current sink pin
- Provides mapping from:
 - PG pins to related inputs & outputs
 - PG pins to automatically connected power nets

Level Shifter



PG-Pin syntax example

```
library(sample) {  
  voltage_map(VDD1, 3.0);  
  voltage_map(VDD2, 3.1);  
  voltage_map(GND1, 0.3);  
  voltage_map(GND2, 0.0);  
  ...  
  cell(test) {  
    pg_pin(P1) {  
      voltage_name : VDD1;  
      pg_type : primary_power;  
    }  
    ...  
    pg_pin(G2) {  
      voltage_name : VDD2;  
      pg_type : primary_ground;  
    }  
    ...  
    leakage_power() {  
      when : "!A";  
      value : 1.5;  
      related_pg_pin : P1;  
    }  
  }  
}
```

Referenced
by pg_pin

```
pin(A) {  
  direction : input;  
  related_power_pin : P2;  
  related_ground_pin : G2;  
  input_signal_swing() {  
    low : 2.0;  
    high : 2.8;  
  }  
  ...  
}
```

Associate power & ground
with signals

Valid values of the pg_type attribute are : primary_power, primary_ground,
backup_power, backup_ground, internal_power and internal_ground

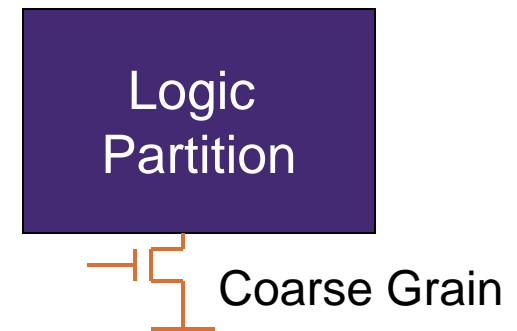
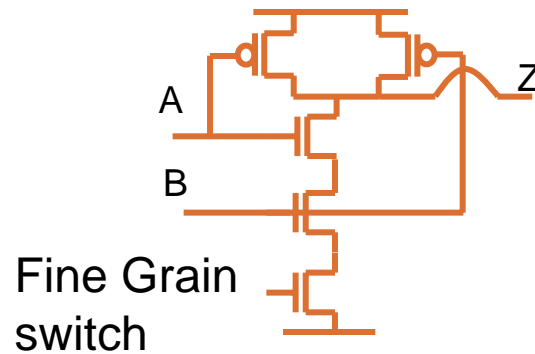
PG Pins: Automated Connection and Checking

connect_power_domain automatically connects cells in four different ways:

- 'primary' power or ground net for a power domain
 - 'backup' (always on) power or ground net for a power domain
 - 'internal' (switched) power or ground net for a power domain
 - Via use-defined `power_connection_type` when more automated connections are needed.
-
- Checking based on ranges defined in the min/max/hi/low values in input/output voltage groups
 - Note: May be adjusted as UPF is developed

Switch Cell Support

- Liberty extended to support switch cells
 - Power down logic to save power
 - Fine grain (per cell)
 - Coarse grain (per block)

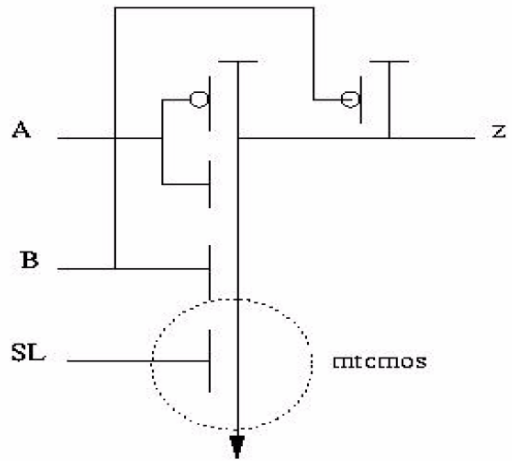


Switch_cell_type: *fine_grain* or *coarse_grain*.

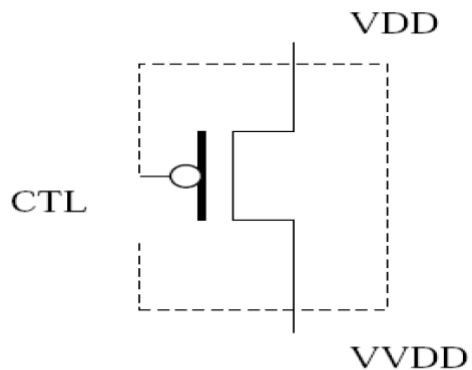
Switch_function: defines the condition to turn off switch/partition

Switch_pin: identifies pin as switch pin

Switch Cells: Coarse-Grained & Fine-Grained

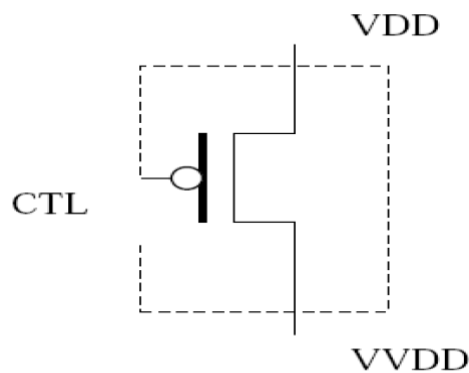


```
cell(NAND2SLP) {
  switch_cell_type : fine_grain;
  leakage_power () {
    when : SL;
    value : 0.008;}
  pin(Z) {
    function : "!(A*B)"; /* switch pin not in here */
    switch_function : "!SL";
    direction : output;
  }
}
```



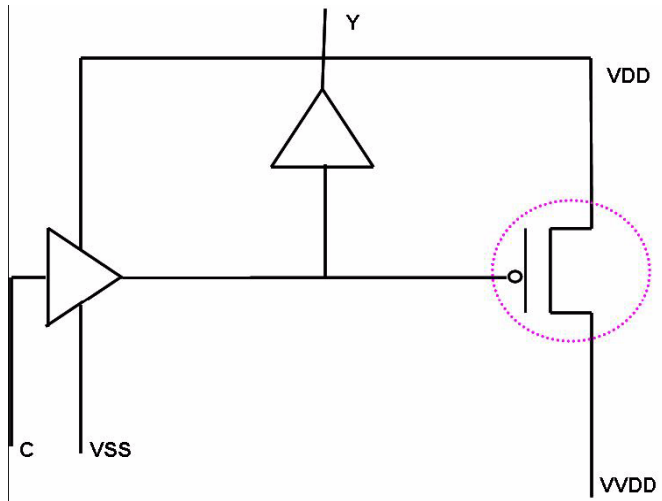
```
cell ( CG_SWITCH ) {
  switch_cell_type : coarse-grain;
  pg_pin ( VDD ) {
    type : primary_power;
    voltage_name : VDD;}
  pg_pin ( VVDD ) {
    type : internal_power;
    switch_function : "CTL"}
}
```

Switch Cells: DC I/V Curve Voltage Drop



```
pin ( CTL ) {
  switch_pin : true;
  capacitance: 0.034;}
/* I/V curve information */
dc_current ( ivt1 ) {
  related_switch_pin : CTL; /* control pin */
  related_pg_pin : VDD; /* source */
  related_virtual_pg_pin : VVDD; /* drain */
  /* index_1 is used to specify the input_voltage
  values
of the switch pin */
  /* index_2 is used to specify the voltage
  difference
values between the related_pg_pin and
related_virtual_pg_pin */
  values("0.010, 0.020, 0.030, 0.030, 0.030", \
"0.011, 0.021, 0.031, 0.031, 0.031", \
"0.012, 0.022, 0.032, 0.032, 0.032", \
"0.013, 0.023, 0.033, 0.033, 0.033", \
"0.014, 0.024, 0.034, 0.034, 0.034");
```

Switch Cells: Automated power & control connection and checking



```
cell ( CG_SWITCH ) {  
  ...  
  switch_cell_type : coarse-grain;  
  pg_pin ( VDD ) {  
    type : primary_power;  
    voltage_name : VDD;  
  }  
  pg_pin ( VVDD ) {  
    type : internal_power;  
    switch_function : "CTL"  
  }  
  pin ( CTL ) {  
    switch_pin : true;  
    capacitance: 0.034;  
  }  
}
```

Need more information ?

- Full Liberty Documentation, Parser & Messageboards
 - Tap-in website & Si2 Liberty TAB websites
 - http://www.synopsys.com/partners/tapin/tapin_program.html
 - <http://si2.org/?page=701>
- Using Liberty with Low Power commands
 - Accellera UPF Technical Subcommittee
 - <http://www.accellera.org/activities/upf>