UPF – A Cooperative Effort Under Accellera
Agenda

• Introduction (00:15)
  – Kevin Kranen, Synopsys
• UPF Low Power Design Basics (00:30)
  – Stephen Bailey, Mentor Graphics
• Low Power Design Implementation (00:30)
  – Arvind Narayanan, Magma Design
• Comprehensive MV Verification (00:15)
  – Anand Iyer, ArchPro
• Q & A
Agenda

• Introduction (00:15)
  – Kevin Kranen, Synopsys
• UPF Low Power Design Basics (00:30)
  – Stephen Bailey, Mentor Graphics
• Low Power Design Implementation (00:30)
  – Arvind Narayanan, Magma Design
• Comprehensive MV Verification (00:15)
  – Anand Iyer, ArchPro
• Q & A
UPF is the New Industry Standard
Built from Silicon-proven Technologies

Technology donations to UPF TSC
- Mentor
  - External power configuration file for verification
- Magma
  - Power Management commands
- Vast
  - System level modeling methodology and format
- Synopsys
  - RTL constructs (Verilog and VHDL)
  - Power Management commands
  - Switching activity format – SAIF
- TI
  - Retention cell semantics
- Atrenta, Synchronous DA

UPF Participating Companies
- AMD
- ArchPro
- ARM
- Atrenta
- Azuro
- Cadence
- ChipVision
- FreeScale
- IBM
- Infineon
- Intel
- LCDM Eng
- LSI Logic
- Magma
- Mentor
- Nokia
- Nordic Semi
- Novas
- NXP
- Qualcomm
- Si2
- STARC
- STM
- Synchronous DA
- Synopsys
- TI
- Toshiba
- VaST
- Virage Logic
- Xilinx

• Feb 07 - UPF 1.0 standard approved by Accellera !
• March 07 – IEEE P1801 PAR / Study Group underway
UPF 1.0 Industry Endorsement & Support

- **Infineon** - The quick development and release of the UPF 1.0 standard is based on our close partnership relations with EDA suppliers who share the same vision and attitude in making things happen. We are convinced that UPF will support us in achieving zero-defect quality and our productivity objectives, which both are key for Infineon's World class Automotive Product Portfolio.

  Hartmut Hiller, Senior Director Design Methodology Automotive, Industrial & Multimarket

- **Synopsys** - Applauds Accellera for approving the UPF standard for low power design and verification. We plan to deliver our UPF 1.0-based implementation and verification solution during 2007. In response to customer demand for a standard that enables consistent and interoperable end-user low power flows and methodologies, Synopsys - together with Magma Design Automation, Mentor Graphics, leading end-customers and IP companies - has made strong contributions to UPF 1.0 based on our proven technologies. UPF 1.0 is ready for industry use.

  Rich Goldman, Vice President, Synopsys, Strategic Market Development
UPF 1.0 Industry Endorsement & Support

- **Magma** - The speed at which the UPF standard has been developed and approved demonstrates the power of one open, inclusive and cooperative industry-wide effort. Users will realize significant improvements in productivity and quality of results by having a single, portable file and format with which they can specify, modify and maintain design data. Accellera, Magma, Mentor, Synopsys and all the companies that donated technology and expertise should be commended.

  Kam Kittrell, General Manager, Design Implementation Business Unit, Magma Design Automation

- **Mentor** - Designers want a single format that is simple to use, extensible, and capable of describing complex power behavior. The Unified Power Format (UPF) 1.0 standard achieves this by being open and comprehensive enabling support from leading EDA vendors and customers for industry-wide adoption. Mentor is committed to Accellera's UPF 1.0 standard as we are a leading contributor of our proven technology to this open standard for low power design and verification.

  Robert Hum, Vice President & General Manager, Mentor Graphics Design Verification & Test Division
Digital Design

COT/ASIC/FPGA Synthesis + Physical Implementation + DFT + Signoff

(TTM Revenue: $1,042.0M)

Based on Q4 05 through Q3 06 EDAC MSS data plus other publicly available market data
Digital Verification

RTL Verification + Formal Verification
(TTM Revenue: $579.8M)

Based on Q4 05 through Q3 06 EDAC MSS data plus other publicly available market data
Digital Verification 😊

Based on 2007 John Cooley DeepChip Verification Survey “Mindshare” – 818 Respondents
What is UPF?

• Unified Power Format
• UPF provides the ability for electronic systems to be designed with power as a key consideration early in the process.

• Why UPF?
  – No existing HDL adequately supports the specification of power distribution and management
  – Vendor-specific formats are non-portable and create opportunities for bugs via inconsistent specifications
Consistent Commands to:

• Define power distribution architecture
  – Power domains
  – Supply rails
  – Switches

• Create power strategy
  – Power state tables

• Set up and map
  – Retention
  – Isolation
  – Level shifters
  – Switches
Why has Power Become THE Dominant Constraint?

- Low power is key for many (most) applications
  - Mobile, processing, communications, consumer
- Driver: Process technology
  - >100nm:
    - Switching dominates power consumption
  - <100nm
    - Static leakage consumes >50% of power

Intel 45nm Test Chip

Intel 45nm Memory Cell
<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 Sep 06</td>
<td>Accellera TSC formation</td>
</tr>
<tr>
<td>18 Sep 06</td>
<td>Design Objectives Document; Weekly meetings start</td>
</tr>
<tr>
<td>5 Oct 06</td>
<td>Si2 / Accellera Workshop on Low Power</td>
</tr>
<tr>
<td>30 Oct 06</td>
<td>First drafts available for review</td>
</tr>
<tr>
<td>30 Nov 06</td>
<td>Submission to Accellera Board for Approval</td>
</tr>
<tr>
<td>23 Jan 07</td>
<td>Accellera Technical committee approves standard</td>
</tr>
<tr>
<td>22 Feb 07</td>
<td>Accellera Board approves UPF – V1.0 released</td>
</tr>
<tr>
<td>23 Feb 07</td>
<td>IEEE study group formed</td>
</tr>
<tr>
<td>7 May 07</td>
<td>(expected) Establish IEEE P1801 Working Group to develop a proposed standard for Low Power</td>
</tr>
</tbody>
</table>

Public Download [http://www.accellera.org](http://www.accellera.org)
Synopsys UPF Support

• Based on existing, proven capabilities
  – 2 years prior experience
  – Over 20 Multi-voltage tapeouts
  – Numerous power-gating tapeouts

• Broad Product Support in 2007
  – Verification
  – Synthesis
  – Physical Implementation
  – Checking
  – Signoff
  – Low Power IP
UPF in a Nutshell

• What is UPF?
  – Abstract supply distribution and control network specification
  – Power-aware design intent
  – Used throughout design flow

• Key Concept: **UPF extends without changing the logic design specification**
  – Golden source is not touched
  – No re-verification of logic-only
  – UPF augments the HDL specification

• Key Concept: **Matching simulation & implementation semantics**
Agenda

• Introduction (00:15)
  – Kevin Kranen, Synopsys
• UPF Low Power Design Basics (00:30)
  – Stephen Bailey, Mentor Graphics
• Low Power Design Implementation (00:30)
  – Arvind Narayanan, Magma Design
• Comprehensive MV Verification (00:15)
  – Anand Iyer, ArchPro
• Q & A
UPF Value Points

• TCL
  – Exploit all TCL scripting capabilities

• IP Accommodating
  – Specify how separate from what
    • Which registers require retention
    • Specifics of retention (supplies, control signals)

• Default, general application
  – Retention and isolation strategies
  – Recursive inclusion, etc
  – With well-defined precedence semantics
    • More specific has higher precedence

• Legacy methodology friendly
  – Set_power_switch
  – User-defined supply net state conversions

• Semantic integration with logic design
UPF / Logic Design Relationship

• Key Concept:
  Everything defined in UPF exists in the Logic Hierarchy
UPF / Logic Design Relationship

• Key Concept:
  *Everything defined in UPF exists in the Logic Hierarchy*

```plaintext
create_power_domain pdA
  -include_scope A
```

Module A

Logic Design
create_power_domain

create_power_domain  domain_name

  [-elements  list]
  [-include_scope]
  [-scope  instance_name]
UPF / Logic Design Relationship

• Key Concept:
  Everything defined in UPF exists in the Logic Hierarchy

```create_supply_port spAOn -domain pdA```

Module A

Logic Design
create_supply_port

create_supply_port  port_name  
-domain  domain_name  
[ -direction  <in  |  out> ]
UPF / Logic Design Relationship

- **Key Concept:**
  
  *Everything defined in UPF exists in the Logic Hierarchy*

  ```
  create_supply_net RET
  -domain pdA
  ```

---

**Logic Design**

- **Module A**
- **Lp1**
- **Lp2**
- **Ln1**
- **Ln2**
- **Ln3**
- **spAOn**
- **pdA**
- **RET**
**UPF / Logic Design Relationship**

- **Key Concept:**
  
  *Everything defined in UPF exists in the Logic Hierarchy*

```plaintext
create_supply_net PR
-domain pdA
```

![](image)

- **Module A**
  - Logic Design
  - pdA
  - PR
  - RET
  - spAOn
  - Lp1
  - Ln1
  - Ln2
  - Ln3
  - Lp2
  - Lp3
create_supply_net

create_supply_net  net_name
   -domain  domain_name
   [-reuse]
   [-resolve < unresolved
       |  one_hot
       |  parallel >]
UPF / Logic Design Relationship

• Key Concept:

*Everything defined in UPF exists in the Logic Hierarchy*

```plaintext
create_power_switch SW1 -domain pdA
 -input_supply_port {inp RET}
 -output_supply_port {outp PR}
```

---

**Module A**

**Logic Design**
create_power_switch

create_power_switch switch_name
  -domain domain_name
  -output_supply_port { port_name supply_net_name }*
  {-input_supply_port { port_name supply_net_name }}*
  {-control_port { port_name net_name }}*
  {-on_state { state_name input_supply_port
    {boolean_function}}}
  [{-on_partial_state { state_name input_supply_port
    {boolean_function}}]*
  {-ack_port { port_name net_name [{boolean_function}] }}]*
  {-ack_delay { port_name delay}]*
  {-off_state { state_name {boolean_function} }}]*
  {-error_state { state_name {boolean_function} }}]*
The UPF / Logic Design Relationship

- Key Concept:
  Everything defined in UPF exists in the Logic Hierarchy

```plaintext
connect_supply_net RET
-ports {spAOn}
```

Module A

Logic Design
connect_supply_net

connect_supply_net \textit{net\_name} \\
\textbf{-ports} \textit{list} \\
\textbf{-pins} \textit{list} \\
\textless \textbf{-cells} \textit{list} \mid \\
\textbf{-domain} \textit{domain\_name} \rangle \\
\textless \textbf{-rail\_connection} \textit{rail\_type} \mid \\
\textbf{-pg\_type} \textit{pg\_type} \rangle * \\
\textbf{-vct} \textit{vct\_name}
The UPF / Logic Design Relationship

• Key Concept:
  Everything defined in UPF exists in the Logic Hierarchy

  set_domain_supply_net pdA
  -primary_power_net PR
  -primary_ground_net VSS

Module A

Logic Design
Automation

• Single create_supply_port command
• Single create_supply_net command
• Supply is “routed” to all design elements in PD

Flexibility

• PD can consist of non-contiguous design elements
• PD always has scope
  – Ports, nets, etc created in that scope
• Supply is routed to all design elements
• Auto re-naming avoids conflicts
Simulation Semantics Overview

• Key Concept:

All design elements in a power domain share the same primary power and ground supplies

• ON:
  – Both primary power and ground are ON
  – Supply port drives a voltage value

• OFF:
  – Primary power and/or ground are OFF
  – Voltage value is irrelevant

• PARTIAL_ON
  – For power-aware models may more accurately reflect switching capacitive transition
OFF Means

• All registers are corrupted
  – Retention (shadow) registers have separate supply(ies)
  – Logic types = X
  – Other types = default initial value

• Any signal/net driven by logic that is OFF is corrupted
  – Isolation cells have separate supply(ies)

• No evaluation of logic occurs while it is OFF
ON Means

• When logic is powered on (event)
  – Combinatorial processes are evaluated
  – Including continuous assignments
  – Edge triggered processes are not evaluated until the next active edge
  – Logic (processes) are re-enabled for evaluation
UPF / HDL Interoperability

• Accellera 1.0 UPF Standard
  – Packages
one_hot resolution:
- Predefined
- Specified in create_supply_net command
- At most one supply (PARTIAL_)ON at any time
- Voltage value is the value of the ON port
**Supply Net Resolution**

**Parallel Switch**

parallel resolution:
- Predefined
- Specified in create_supply_net command
- All must be OFF
- Or all must be ON and at same voltage
- If any PARTIAL_ON, then PARTIAL_ON
Mentor’s UPF Low Power Design Verification Solutions
Mentor’s UPF Low Power Design Verification Solutions

- Questa Simulation
  - RTL & gate verification of low power design intent
    - Power gating
    - Retention
    - Isolation

- FormalPro
  - Is the implementation equivalent to what was verified?
Accellera’s UPF: The Power of One

HDL Logic Design

Vlog / VHDL Compile

Library

Vopt Elab & Optimizer

UPF Low Power Intent

Vsim Simulation

Questa Coverage Report

Questa Power Aware Simulation Flow
Low Power Equivalency Checks

- Power Management Status in the main transcript
- Power Management Details Report
  - Failed due to wrong register type
Unifying Low Power Design with UPF

*The Power of One*

Arvind Narayanan
Agenda

• Introduction (00:15)
  – Kevin Kranen, Synopsys
• UPF Low Power Design Basics (00:30)
  – Stephen Bailey, Mentor Graphics
• Low Power Design Implementation (00:30)
  – Arvind Narayanan, Magma Design
• Comprehensive MV Verification (00:15)
  – Anand Iyer, ArchPro
• Q & A
Talus™ Platform
Rapid concurrent closure of timing, power and yield

25% less power
Power signoff

Low Power & Power Synthesis

Extraction, Timing/SI Closure

DFM & Variability

RTL to GDSII

Automation

Signoff In the Loop

10% better QoR
5X faster TAT

10% Margin Reduction
DFM signoff

Accellera’s UPF: The Power of One
Total Power Optimization
RTL to GDSII

Integration, Innovation, Automation!

- **Sign-off power and IR drop**
  - Static and Transient
  - Intelligent de-cap methodology
  - Built-in spice engine

- **MTCMOS**
  - Power On/Off behavior
  - Rush current analysis

- **Thermal**
  - Impact on delay/leakage

- **Leakage Power**
  - Automated MTMOS/VTCMOS
  - Concurrent Multi-vt flow

- **Dynamic Power Reduction**
  - Virtually flat MVDD flow
  - Unique Gas station methodology

- **Up to 25% reduction in CTS**
  - Advanced cloning and sink clustering

- **Automatic Power Grid Synthesis**
  - Incremental power grid design

---

Accellera's UPF: The Power of One

Interoperability Forum 26 Apr 07
UPF Support in Magma

- Ability to allow the specification of implementation-relevant power information early in the design process

- UPF provides a consistent format to specify power-aware design information

- UPF also defines consistent semantics across verification and implementation
  - `create_power_domain`
  - `create_supply_port`
  - `create_supply_net`

Sample UPF Commands for MVDD flow
Complete Low Power Design Specification
= HDL + UPF

- Power Domains
- Power Distribution Network
  - Switches and Supply Nets
- Power State Table
- Level Shifting
- Isolation
- Retention
- Switching Activity
Automated Voltage Island Methodology

- Complete automated flow for defining and connecting domains
- Maintains virtual hierarchy in the flow
- Handles level shifters and isolation cell insertion
- CTS honors domain boundaries
- Routing and cells contained within domains
- Concurrent analysis and optimization

Types of MVDD Designs

1. **DVFS**
   - 1.2v 200Mhz Constant VDD
   - 1.4v 200Mhz Constant VDD

2. **Switched**
   - 1.2v 200Mhz Constant VDD
   - 1.4v 200Mhz Constant VDD

3. **MTCMOS**
   - 1.2v 200Mhz Switched VDD
   - 1.4v 200Mhz Constant VDD

   - 1.6v 200Mhz Switched VDD - Constant
   - 1.2 to 1.6v 100 to 200Mhz Variable VDD - "ON"
UPF / Logic Design Relationship

• Key Concept:

Everything defined in UPF exists in the Logic Hierarchy

```
create_power_domain pdA
   -include_scope A
```
UPF / Logic Design Relationship

- Key Concept: *Everything defined in UPF exists in the Logic Hierarchy*

```plaintext
set_domain_supply_net pdA
-primary_power_net PR
-primary_ground_net VSS
```

Module A

Logic Design
Domain Creation and Mapping

- Domain Creation
- Attaching cells
- Domain Parameters
- Libraries for domain
- Floorplans for domains

Logical Mapping

Electrical Mapping

Physical Mapping
Level Shifters in MVDD flows

- Level shifters translate from one voltage swing to another

- Level shifter considerations:
  - Pick a power domain or a set of elements
  - Select input ports, output ports, or both
  - Tolerate a voltage difference threshold
  - UPshift or downSHIFT rule
  - Location (self, parent, sibling, fanout, auto)
  - Do or don’t do it
Isolation Cells in MVDD flows

- Floating outputs of power-gated circuits
- Isolation control signals force known value
- Isolation and level shifting can be merged
set_level_shifter Command Example

```plaintext
set_level_shifter my_ls  
-domain PDgreen  
-rule low_to_high  
-location self  
-applies_to outputs
```

```plaintext
map_level_shifter_cell  
ls_L2H  
-domain PDgreen  
-lib_cells { /lib/ls_123 }
```

**Diagram:**
- **Before:** Module G6 in logic design with points P5, P6, and P7.
- **After:** Module G6 in domain PDgreen with points P5, P6, Vlo, Vhi, and P7, indicating a level shifter.
**set_isolation Command Example**

```
set_isolation iso3
-domain PDgreen
-isolation_power_net Vbu
-clamp_value 0
-applies_to_outputs
```

```
set_isolation_control
iso3
-domain PDgreen
-isolation_signal CPU_iso
-location self
```

---

**Diagram Description**

- **Module G6** in domain PDgreen
- **Logic Design**
- **P5**, **P6**, **P7**
- **Vbu** node
- **CPU_iso** node
- **P7_iso** node
Level Shifter/Isolation Cell Insertion - Magma

- Uses supply type definition for LS/ISO insertion
- Length and IR drop based insertion

**run gate levelshifter –UPF**
**run gate isolation –UPF**

**Supply Types**

- **Constant** - voltage supply is constant over time
- **Variable** - voltage supply varies over time
- **Switched constant** - constant voltage supply that can be switched off
- **Switched variable** - variable voltage supply that can be switched off
Gas Station Methodology for MVDD Flow

- Mini islands to help buffering long top level nets without level shifters
- Helps handle doughnut shaped domains with congestions
- Buffering on long nets through switched domains can be gracefully handled
- Automatic power tapping from the top level supply
- Optimal number of repeaters and supply taps inserted
MTCMOS Switches

- Coarse Grain Distributed Switches
  - Switches placed in rows to control groups of logic

- Standard cell specific Switches
  - A domain replaced with cells that have header and/or footer switches
  - Easier to implement at the cost of area

- Fine Grain Distributed Switches
  - Cones of logic replaced with cells with header/footer switches

- Global Header/Footer Switches
  - MTCMOS switches at the periphery of the domains

MVDD based MTCMOS Methodology
UPF / Logic Design Relationship

- Key Concept:

Everything defined in UPF exists in the Logic Hierarchy

```
create_power_switch SW1 -domain pdA
    -input_supply_port {inp RET}
    -output_supply_port {outp PR}
```
set_retention Command Example

```plaintext
set_retention ret3
   -domain PDgreen
   -retention_power_net Vbu
   -elements { u37 }  
```

```
set_retention_control
   ret3
   -domain PDgreen
   -save_signal s
   -restore_signal r
```

---

module G6
logic design

module G6
in domain PDgreen
Coarse Grain Insertion Methodology

• Number of switches inserted is based on:
  
  – Explicit – Switches inserted on every grid point
    • Minimum # of switches per row – user defined
  
  – Power – Total power being consumed by a block
    • Evenly distributed over the placeable area
  
  – Voltage Drop – Rail analysis used to determine the location of the current sinks

• Voltage aware incremental switch insertion/removal

  run gate switch –UPF
Quartz Rail - Power Integrity Sign-Off

- Standalone Power sign-off accuracy with early predictability
- Concurrently addresses power, voltage drop, electromigration, Thermal and Timing issues
- Analyze impact of temperature on leakage and performance
- On-the-fly characterization for accurate dynamic IR drop analysis
- Leakage optimization through intelligent de-cap insertion
- MTCMOS power-on and rush current analysis
UPF Roadmap

• UPF version 1.0 released – Feb 22nd 2007
• Magma support for UPF
  – UPF Implementation support – May 07
  – Talus Power for Implementation
  – Quartz Rail for Analysis
Comprehensive Power Management Solution

• Advanced power management techniques from RTL-to-GDSII

• Unique architecture - Single executable, Unique data model

• Embedded analysis - enables concurrent power, timing, area tradeoffs

• Reference Methodology provides well-defined design guidelines
Agenda

• Introduction (00:15)
  – Kevin Kranen, Synopsys
• UPF Low Power Design Basics (00:30)
  – Stephen Bailey, Mentor Graphics
• Low Power Design Implementation (00:30)
  – Arvind Narayanan, Magma Design
• Comprehensive MV Verification (00:15)
  – Anand Iyer, ArchPro
• Q & A
UPF Workshop

Anand Iyer
Senior Director of Marketing
ArchPro Design Automation
Do Designers Need UPF?

- **Manage complexity**
  - # of islands in a single chip is growing
  - Adhoc methods/scripts are useless after 3 islands
  - Formal definition of power intent is a must

- **Consistent flow across the design stages**
  - Support for UPF is available across the flow
  - Descriptive and prescriptive

**UPF provides a structured design flow**
ArchPro Support of UPF

- Comprehensive MV verification based on UPF
- Read-only support of UPF by DAC 07 (June 2007)

- Read/write support – Q4 07
ArchPro: Proven solution at 65nm

Accelerating Multi-Voltage Low Power Designs

Architecture Verification (incl. Coverage, assertions)

Full Verification before Silicon! ➔ First Pass Silicon Success

Early S/W validation on RTL ➔ Months gained for TTM

System & Software validation

Speedy Silicon Debug

Silicon Debug

Explore and Tradeoff Aggressive Architectures

Seamless connection to Mainstream design flows

ArchPro: Proven solution at 65nm

Architecture Verification (incl. Coverage, assertions)

Full Verification before Silicon! ➔ First Pass Silicon Success

Early S/W validation on RTL ➔ Months gained for TTM

System & Software validation

Speedy Silicon Debug

Silicon Debug

Explore and Tradeoff Aggressive Architectures

Seamless connection to Mainstream design flows

ArchPro: Proven solution at 65nm

Architecture Verification (incl. Coverage, assertions)

Full Verification before Silicon! ➔ First Pass Silicon Success

Early S/W validation on RTL ➔ Months gained for TTM

System & Software validation

Speedy Silicon Debug

Silicon Debug

Explore and Tradeoff Aggressive Architectures

Seamless connection to Mainstream design flows
ArchPro Solutions with UPF Support

- **MVSIM**
  - Multi-voltage co-simulator works with ModelSim and VCS

- **MVRC**
  - Vectorless verification of multi-voltage conditions
  - Power sequence prediction

- **MVSYN**
  - Scriptless insertion of protection devices into RTL

Functional Simulation OK

MVSIM identifies the error
How Does ArchPro’s UPF-based Flow Help You?

- **Manage verification complexity**
  - Complete coverage of the states and transitions
  - User generated assertions cover all legal conditions

- **Implementation is guaranteed to be clean**
  - Directives for implementation tools
  - Sign-off checking for any errors
  - Help in silicon de-bug

- **Compare multiple power architectures for feasibility**
What’s Next …

• The Panel
• Get involved with UPF for your customer’s sake
  – Download current standard at accellera.org
  – Join the IEEE P1801 study/working group
  – UPF Workshop at DAC
THANK YOU
Atrenta Support of Unified Power Format (UPF)

Piyush Sancheti
The Need for a Power Standard

- Power-aware design requires specification of power intent
- No existing standard for power intent specification
  - Power intent described in Atrenta SGDC format for SpyGlass
  - Other EDA tools have proprietary formats
  - Customers have internal formats
- Power standard is important
  - Consistent power intent throughout the design flow
  - Interoperability between EDA tools
  - Ease of adoption and consistent results from power-aware design tools
- Atrenta power solution
  - Voltage and power domain verification
  - Power domain sequencing verification
  - Domain-aware power estimation
  - Power reduction and planning
Atrenta Support for UPF

- Atrenta is an active participant in UPF
  - Strongly support efforts for unification of power formats and IEEE standardization
  - Donated SpyGlass SGDC format to Accellera in 2006
  - Participate in technical sub group (TSG) & IEEE p1801 working group
  - Dave Allen (Power Architect) represents Atrenta in UPF

- Atrenta plans to support UPF in SpyGlass Power by July 2007

- Atrenta will work closely with customers for UPF support in SpyGlass Power
  - Provide a transition path from/to SpyGlass SGDC format to/from UPF
  - Ensure UPF support in SpyGlass is adequate and robust for use in design projects
  - Work with UPF members to resolve any tool interoperability issues
UPF Support in SpyGlass Power
SpyGlass Power Requirements from UPF

SpyGlass Power

**Power Reduction and Planning**
*Intelligent power reduction and domain planning at RTL*

**Power Estimation**
*Timing-aware power estimation at RTL, gates, layout*

**Power Domain Sequencing**
*Formally prove power up/down sequencing*

**Power & Voltage Domain Verification**
*Verify and fix level shifter, isolation logic, SRPG, MTCMOS RTL, gates, layout*

UPF

- Library data
- Supplies
- Scope
- Domains
- signals
### UPF examined

- UPF contains all the major categories required by Atrenta SpyGlass
- Atrenta is planning a power format translator for our customers

<table>
<thead>
<tr>
<th>CPF</th>
<th>Atrenta</th>
<th>UPF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Library</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>define_always_on_cell</td>
<td>always_on_cell</td>
<td>set_pin_related_supply</td>
</tr>
<tr>
<td>define_isolation_cell</td>
<td>always_on_pin</td>
<td>set_power_switch</td>
</tr>
<tr>
<td>define_level_shifter_cell</td>
<td>aonbuffer</td>
<td></td>
</tr>
<tr>
<td>define_open_source_input_pin</td>
<td>apcell</td>
<td></td>
</tr>
<tr>
<td>define_power_clamp_cell</td>
<td>isocell</td>
<td></td>
</tr>
<tr>
<td>define_power_switch_cell</td>
<td>isocell</td>
<td></td>
</tr>
<tr>
<td>identify_power_logic</td>
<td>ppgains_naming</td>
<td>powerswitch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>retencell</td>
</tr>
<tr>
<td><strong>Design - supplies</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_bias_net</td>
<td>supply</td>
<td>create_supply_net</td>
</tr>
<tr>
<td>create_power_nets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_ground_nets</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Design - domains</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_power_domain</td>
<td>voltagedomain</td>
<td>add_domain_elements</td>
</tr>
<tr>
<td>update_power_domain</td>
<td>pinvoltage</td>
<td>merge_power_domains</td>
</tr>
<tr>
<td></td>
<td></td>
<td>set_domain_supply_net</td>
</tr>
<tr>
<td><strong>Design - levelshifters</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_level_shifter_rule</td>
<td>leveshifter</td>
<td>map_level_shifter_cell</td>
</tr>
<tr>
<td>update_level_shifter_rules</td>
<td>set_level_shifter</td>
<td></td>
</tr>
<tr>
<td><strong>Design - retention cells</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_state_retention_rule</td>
<td>map_retention_cell</td>
<td></td>
</tr>
<tr>
<td>update_state_retention_rules</td>
<td>set_retention</td>
<td>set_retention_control</td>
</tr>
<tr>
<td><strong>Design - isolation logic</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_isolation_rule</td>
<td>ignoredcrossing</td>
<td>map_isolation_cell</td>
</tr>
<tr>
<td>update_isolation_rules</td>
<td>set_isolation</td>
<td>set_isolation_control</td>
</tr>
<tr>
<td><strong>Design - powerswitches</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_power_switch_rule</td>
<td></td>
<td></td>
</tr>
<tr>
<td>update_power_switch_rule</td>
<td>map_power_switch</td>
<td></td>
</tr>
<tr>
<td><strong>Design - always on</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identify_always_on_driver</td>
<td>aonbufferedsignals</td>
<td></td>
</tr>
<tr>
<td><strong>Design - scoping</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>end_design</td>
<td>current_design</td>
<td>load_upf</td>
</tr>
<tr>
<td>set_design</td>
<td>set_instance</td>
<td>set_scope</td>
</tr>
<tr>
<td><strong>Design - modes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_mode_transition</td>
<td>add_pst_state</td>
<td></td>
</tr>
<tr>
<td>create_power_mode</td>
<td>create_pst</td>
<td></td>
</tr>
<tr>
<td>update_power_mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multimode analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_analysis_view</td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_nominal_condition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_operating_corner</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set_switching_activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>update_nominal_condition</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Simulation semantics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bind_checker</td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_hdl2upf_vct</td>
<td>create_upf2hdl_vct</td>
<td></td>
</tr>
<tr>
<td><strong>Miscellaneous</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_global_connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>define_library_set</td>
<td>connect_supply_net</td>
<td></td>
</tr>
<tr>
<td>set_array_naming_style</td>
<td>create_supply_port</td>
<td></td>
</tr>
<tr>
<td>set_cpf_version</td>
<td>get_supply_net</td>
<td></td>
</tr>
<tr>
<td>set_hierarchy_separator</td>
<td>name_format</td>
<td></td>
</tr>
<tr>
<td>set_power_target</td>
<td>save_upf</td>
<td></td>
</tr>
<tr>
<td>set_power_unit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set_register_naming_style</td>
<td>upf_version</td>
<td></td>
</tr>
<tr>
<td><strong>Design - scoping</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>end_design</td>
<td>current_design</td>
<td>load_upf</td>
</tr>
<tr>
<td>set_design</td>
<td>set_instance</td>
<td>set_scope</td>
</tr>
<tr>
<td><strong>Design - modes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_mode_transition</td>
<td>add_pst_state</td>
<td></td>
</tr>
<tr>
<td>create_power_mode</td>
<td>create_pst</td>
<td></td>
</tr>
<tr>
<td>update_power_mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Multimode analysis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_analysis_view</td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_nominal_condition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_operating_corner</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set_switching_activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>update_nominal_condition</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Simulation semantics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bind_checker</td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_hdl2upf_vct</td>
<td>create_upf2hdl_vct</td>
<td></td>
</tr>
<tr>
<td><strong>Miscellaneous</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>create_global_connection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>define_library_set</td>
<td>connect_supply_net</td>
<td></td>
</tr>
<tr>
<td>set_array_naming_style</td>
<td>create_supply_port</td>
<td></td>
</tr>
<tr>
<td>set_cpf_version</td>
<td>get_supply_net</td>
<td></td>
</tr>
<tr>
<td>set_hierarchy_separator</td>
<td>name_format</td>
<td></td>
</tr>
<tr>
<td>set_power_target</td>
<td>save_upf</td>
<td></td>
</tr>
<tr>
<td>set_power_unit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set_register_naming_style</td>
<td>upf_version</td>
<td></td>
</tr>
</tbody>
</table>
UPF Support in SpyGlass Power

SpyGlass Desktop

Translation command line

Accellera's UPF: The Power of One
Agenda

• Introduction (00:15)
  – Kevin Kranen, Synopsys
• UPF Low Power Design Basics (00:35)
  – Stephen Bailey, Mentor Graphics
• Low Power Design Implementation (00:35)
  – Arvind Narayanan, Magma Design
• Power Intent Checking (00:15)
  – Piyush Sancheti, Atrenta
• Q & A
Interleaver UPF Demo

Logic Hierarchy View

Interleaver Tester (TB)

Interleaver1 (PD_main)

in2wire  pkt_counter  out2wire  fifo

PD0  PD1

Floorplan View

PD_main

PD0  pkt_counter

in2wire

PD1  fifo  ram_block

out2wire

VDD