In this paper, we present the UPF Successive Refinement methodology in detail. We explain how power management constraints can be specified for IP blocks to ensure correct usage in a power-managed system. We explain how a system’s power management architecture can be specified in a technology-independent manner and verified abstractly, before implementation. We also explain how implementation information can be added later. Finally, we explain the benefits of Successive Refinement.

Successive Refinement

Constraint UPF:
- Describes the power intent inherent in the IP - power domains/states/isolation/retention etc. - Constraints are part of source IP and travel with the RTL

Configuration UPF:
- Describes application-specific configuration of instances - supply sets, power states, logic expressions, etc. - Required for simulation - created by end user

Implementation UPF:
- Describes technology-specific implementation of system - supply nets/ports, switches, etc. - Required for implementation - created by end user

Requirements for Successive Refinement:
- Clear Communication between IP Provider and IP Consumer
- Decreases risks and facilitates successful usage
- Separation of Logical Design from Implementation
- Verification can start earlier, before technology is known
- Easier retargeting to different technologies
- Easier debugging at each stage
- Preservation of Verification Equity
- No need to re-verify logical configuration for new technology

Arm Cortex®-MPCore Processor IP

A System with an Instance of the IP Block

Configuration UPF for an IP Instance

1. Load Constraint UPF Files onto Instances
2. Define Control Logic
3. Define Retention Strategy
4. Define Isolation Strategies
5. Update Supply Sets with Supply Nets
6. Update Power States with Voltages
7. Define Other Technology Info As Required

Arm Cortex®-MPCore Processor IP

A System with an Instance of the IP Block

Configuration UPF for an IP Instance

1. Load Constraint UPF Files onto Instances
2. Define Control Logic
3. Define Retention Strategy
4. Define Isolation Strategies
5. Update Supply Sets with Supply Nets
6. Update Power States with Voltages
7. Define Other Technology Info As Required

Arm Cortex®-MPCore Processor IP

A System with an Instance of the IP Block

Configuration UPF for an IP Instance

1. Load Constraint UPF Files onto Instances
2. Define Control Logic
3. Define Retention Strategy
4. Define Isolation Strategies
5. Update Supply Sets with Supply Nets
6. Update Power States with Voltages
7. Define Other Technology Info As Required

Successive Refinement: A Methodology for Incremental Specification of Power Intent

Adnan Khan, Eamonn Quigley, John Biggs, Erich Marschner
[adnan.khan/eamonn.quigley/john.biggs]@arm.com, erich_marschner@mentor.com

IEEE 1001 UPF enables early specification of “power intent”, or the power management architecture of a design, so that power management can be taken into account during design verification and verified along with design functionality. The verified power intent then serves as a golden reference for the implementation flow.

To fully realize the advantages of this capability, a methodology called Successive Refinement was conceived during development of IEEE 1801-2009 UPF. However, this methodology is still not well understood in the industry.

In this paper, we present the UPF Successive Refinement methodology in detail. We explain how power management constraints can be specified for IP blocks to ensure correct usage in a power-managed system. We explain how a system’s power management architecture can be specified in a technology-independent manner and verified abstractly, before implementation. We also explain how implementation information can be added later. Finally, we explain the benefits of Successive Refinement.

Successive Refinement

Constraint UPF:
- Describes the power intent inherent in the IP - power domains/states/isolation/retention etc. - Constraints are part of source IP and travel with the RTL

Configuration UPF:
- Describes application-specific configuration of instances - supply sets, power states, logic expressions, etc. - Required for simulation - created by end user

Implementation UPF:
- Describes technology-specific implementation of system - supply nets/ports, switches, etc. - Required for implementation - created by end user

Benefits of Successive Refinement:
- Clear Communication between IP Provider and IP Consumer
- Decreases risks and facilitates successful usage
- Separation of Logical Design from Implementation
- Verification can start earlier, before technology is known
- Easier retargeting to different technologies
- Easier debugging at each stage
- Preservation of Verification Equity
- No need to re-verify logical configuration for new technology