Design Process Overview

- System Level Design
- Board/Module Design
- IC Design
- Package Design
System Level Design

- Functionality
  - Transformation performed on input
- Constraints
  - Cost
  - Power
  - Performance
  - Size
System Level Design Partitioning

- Software / Hardware
- IC’s & Communication
IC Design

• Many of today’s ICs look like “systems”
• Considerations for functionality in:
  – Hardware
  – Firmware
• Memory and Cache Sizing
• Functional partitioning into blocks
  \[ f_{1,1}(\text{in}_{1,1}) \quad f_{1,2}(\text{in}_{1,2}) \]
• Power Management Decisions
**“Typical” SoC’s**

- Full-Featured CPUs
  - Multiple Cores
  - Cache
- Graphics Processing
- Special Purpose
  - Video Enc/Dec
    - H.264/H.265
  - Encrypt/Decrypt
  - Differentiating IP ...
- Interconnect / Communication
- Radios
- Sensors
- ...

![Diagram showing components of SoC: Multi-Core CPU w/Cache, Graphics Processing (GPU), Special Purpose (0...n), Radios, Sensors.](image-url)
IC Design Power Partitioning

• Aggressive designs need more than simple frequency scaling
  – Quadratic power reduction from voltage scaling on top of frequency scaling is a *must* for power sensitive applications

• IC, Blocks and Sub-Blocks are partitioned into power domains

\[ f_{1,1,PD1} (in_{1,1,PD1}) \quad f_{1,1,PD2} (in_{1,1,PD2}) \]

– *Important for models to handle this effect!*
IC Design Power Partitioning

• Important Considerations:
  – Number of rails
  – Control complexity
  – Interfaces
  – Noise
  – Design cost
IC Design Power Management

P / I Measurement
- On-Chip / Off-Chip
- Package I/O
- Board Impact

V / I Regulation
- On Chip / Off Chip
- Package I/O
- Board Impact

Power Design Points
- Thermal Timescales: 100’s milliseconds to seconds
- Component Protection Timescales: microseconds
Package Design

- Thermal / Power Considerations
  - Organic vs. Ceramic
  - Active vs. Passive Cooling
  - Surrounding Thermal / Noise Environment
- Size
- Number of I/O pins
- Number of Power/GND pins
- Cost
- More than Moore
Board/Module Design

• Analog Chips
  – Power, Voltage, Current Sensors & Regulators
  – Radios
  – Motion and other sensors ... 

• Power Integrity Issues
  – Rail noise
  – $\Delta$ Voltage performance

• Size

• Cost / Yield
Design Process Overview

System Level Design

Partition into:
• HW/SW
• IC’s
• Interconnect

Board/Module Design

Partition into:
• HW/SW/FW
• Blocks/Sub-Blocks
• Power Domains

IC Design

Partition into:
• HW/SW/FW
• Blocks/Sub-Blocks
• Power Domains

Package Design

Partition into:
• HW/SW/FW
• Blocks/Sub-Blocks
• Power Domains

Partition into:
• HW/SW
• Interconnect

Solve:
• Thermal/Cooling
• Integrity
• Size

Solve:
• Thermal/Cooling
• Integrity
• Size
Design Exploration

• Enable division of work while:
  – Maintaining functionality (verifiably)
  – Annotating refinement (partitioning, interfaces)
  – Tracking new constraints (domains, regions, boundaries)
  – Implementation based power and timing
Example 1

- 4-Core CPU
- GPU
- PMU

- PMU receives activity-based (on-chip) power usage from CPU cores and GPU
Example 1 (continued)

• Based on power data, we would like to implement an algorithm in the PMU to maximize performance based on a fixed max power budget (power/performance steering)

• Scenarios
  – CPU
    1. Each core is independently DVFS
    2. All cores share DVS (i.e. all run at same voltage)
  – PMU
    A. Algorithm in HW
    B. Algorithm in SW
  – GPU
    • DVFS
Example 1 (continued)

• Use transaction-based models to evaluate 4 options:
  1. CPU each core is independently DVFS, PMU HW
  2. CPU each core is independently DVFS, PMU SW
  3. CPU all cores share DVS, PMU HW
  4. CPU all cores share DVS, PMU SW

• Functionality is the same, but
  – Different power
  – Different performance
  – Different area
  – Different cost

6/9/2015  B. Pangrle, 52nd DAC: System Level Power Workshop
Example 2

- 4-Core CPU
- GPU
- Video
  Encoder/Decoder

- CPU/GPU based SoC with video encoding/decoding functionality
Example 2 (continued)

• Use transaction-based models to evaluate 3 options:
  1. Video en/decode in SW on CPU
  2. Video en/decode in SW on GPU
  3. Video en/decode in special purpose HW unit

• Again video en/decode functionality is the same but:
  – Different power, performance, area and cost

• Annotate functional model with implementation based information

• Opens possibility for direct SW->HW synthesis
High-Level Synthesis (HLS) Opportunity

• Functional Module Generation
  – Enable designers to “quickly” generate multiple implementations of the same functionality

• Automate Adding Implementation Specific Information to Transaction-Based Models
  – Timing
  – Power
Relevant IEEE Standards and WGs

- UPF IEEE Standard 1801™-2013, 1801a™-2014
- Unified Hardware Abstraction and Layer WG IEEE P2415™
- Power Modeling Meta-standard WG IEEE P2416™
- SystemC®/TLM IEEE Standard 1666™-2011
- IP-XACT IEEE Standard 1685™-2014
Summary

1. Cleanly support functional models w/ multiple possible implementations
   - Timing
   - Power
2. Automatically generate constraints
   - Create UPF for chosen partitioning (IC’s, blocks, power domains)
   - Enable activity analysis
3. Enable verification (functionality, timing, power)
   - Multiple levels
   - Allow run-time vs. accuracy tradeoffs
4. Enable back annotation / model updates
   - As design progresses improve and refine models
Thank You

I'm just beginning to see
Now I'm on my way

--Justin Hayward