SoC Design

Lecture 6: IP Cores

Shaahin Hessabi
Department of Computer Engineering
Sharif University of Technology
IP Core

- Intellectual Property (IP) core: predesigned and pre-verified modules used in ASIC/SOC designs.
  - A Virtual Components (VC) with well-defined functions, usage method, and tools to support its usage.

Benefits:
- Reduce time-to-market by means of design reuse
- Reduce the possibility of failure

Drawback:
- Designing an IP block generally requires greater effort and higher cost

Resources versus Number of Uses

---

Sharif University of Technology

IP Cores
Design Reuse

Design with reuse can have a significant impact:

- IP blocks should have well-defined interfaces.

![Diagram showing Design Reuse](image)
What Is an IP Core?

- At least 5K gates,
- Pre-designed,
- Pre-verified,
- Re-usable,
- H/W S/W Functional Blocks.
- Examples:
  - Processor: ARM7, ARM9, and ARM10, ARC, Leon.
  - Mixed signal: ADCs, DACs, Audio Codecs, PLLs, OpAmps.
  - Encryption: PKuP, DES.
  - I/Os: PCI, USB, 1394, 1284, E-IDE, IrDA.
  - Miscellaneous: UARTs, DRAM Controller, Timers, Interrupt Controller, DMA Controller.
Environment

- **Core Providers:**
  - What do I need to deliver a “product” while protecting technology and avoiding downstream liabilities?

- **Core Users:**
  - How do I effectively use external components?
  - How do I guarantee that the end product works?

- **Foundry Services.**

- **Standards Folks:**
  - Define interfaces/standards/methods to maximize mix-and-match.

- **The Legals:**
  - Clarify responsibilities and liabilities.
Advantages of Core-Based Design

- Allows to functionally verify each complex function block with post-layout timing as soon as RTL design and simulation are completed.
- Makes the back-end physical data available at the front-end of the design cycle, where it can influence algorithm and/or architecture trade-offs (Coupling physical and system design).
- Bridging the gap between front end and back end virtually eliminates timing uncertainty
  - It is possible to use post-layout parasitics.
- Results:
  - More time for block design (block-level timing convergence), but net savings 25% to 30% on the overall timing convergence (chip-level) and gate-level debugging.
Marketplace Experiences

■ Small IP:
  - Blocks requiring 1-2 staff years to design are priced at 1/3 of the development cost.
  - Buyers are skeptical about the value and often prefer to do these in-house.

■ Medium IP:
  - Blocks requiring 5-10 staff years are profitable for both seller and buyer.

■ STAR IP:
  - Blocks requiring 100+ staff years to design (like ARM, MIPS) have become bestsellers and come with lots of support.
Core Types- Soft Cores

- Soft cores ("code"):
  - RTL (or higher level) description, to provide functional descriptions of IPs.
  - Maximum flexibility and reconfigurability, i.e., can be changed to suit an application.
  - Must be synthesized, optimized, and verified by user before integration into designs.
    - Quality of a soft IP depends on the effort needed in the IP integration stage.
  - Technology independent: may be re-synthesized across processes.
  - Significant IP protection risks.
  - May include macroblocks or megacells, typically designed at the physical level.
    - Soft core provider supplies behavioral or functional models for the megacells used in the core (for simulating the core at high levels).
    - The core user has to implement the technology-dependent megacells—a potentially challenging task.
Core Types- Firm Cores

- Firm cores ("code+structure"): role of firm IPs in ASIC design flow
  - Targeted gate-level netlists to specific physical libraries after going through synthesis without performing the physical layout.
    - To be placed and routed.
  - Structurally and topologically optimized for performance and area through floor-planning and placement.
    - using a range of process technologies
  - Exist as synthesized code or as a netlist of generic library elements.
Core Types- Hard Cores

- Hard cores (“physical”):
  - Consist of hard layouts using particular physical design libraries.
  - Delivered in masked-level designed blocks (GDSII format).
  - Ready for “drop in.”
    - Integration of hard IP cores is quite simple.
  - Include layout and timing (technology dependent).
  - Optimized implementation, highest performance for their chosen physical library.
  - Minimum flexibility and portability in reconfiguration and integration across multiple designs and technologies.
  - IP is easily protected.
  - Mostly processors and memory.
  - Functional test vectors or ATPG vectors available.
Advantages/disadvantages of hard core

- **Ease of use**
  - Developer already designed and tested core
    - Can use right away
    - Can expect to work correctly

- **Predictability**
  - Size, power, performance predicted accurately

- **Not easily mapped (retargeted) to different process**
  - E.g., core available for vendor X’s 0.25 micrometer CMOS process
    - Can’t use with vendor X’s 0.18 micrometer process
    - Can’t use with vendor Y
Advantages/disadvantages of soft/firm cores

- **Soft cores**
  - Can be synthesized to nearly any technology
  - Can optimize for particular use
    - E.g., delete unused portion of core
      - Lower power, smaller designs
  - Requires more design effort
  - May not work in technology not tested for
  - Not as optimized as hard core for same processor

- **Firm cores**
  - Compromise between hard and soft cores
    - Some retargetability
    - Limited optimization
    - Better predictability/ease of use
Core Types (cont’d)

<table>
<thead>
<tr>
<th>Behavioral HDL</th>
<th>System Specification</th>
<th>Bus Functional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTL HDL</td>
<td>System Design</td>
<td>RTL Functional</td>
</tr>
<tr>
<td>Firm</td>
<td>Gate Netlist</td>
<td>Gate Functional</td>
</tr>
<tr>
<td>Hard</td>
<td>Mask Data</td>
<td>Physical Design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fault Coverage</td>
</tr>
</tbody>
</table>
Comparison of Different IP Formats

<table>
<thead>
<tr>
<th>IP Format</th>
<th>Representation</th>
<th>Optimization</th>
<th>Technology</th>
<th>Reusability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard</td>
<td>GDSII</td>
<td>Very High</td>
<td>Technology Dependent</td>
<td>Low</td>
</tr>
<tr>
<td>Soft</td>
<td>RTL</td>
<td>Low</td>
<td>Technology Independent</td>
<td>Very High</td>
</tr>
<tr>
<td>Firm</td>
<td>Targeted Netlist</td>
<td>High</td>
<td>Technology Generic</td>
<td>High</td>
</tr>
</tbody>
</table>
Benefits of Licensing over Designing IPs

1. Lack of expertise in designing application-specific reusable building blocks.
2. Savings in time and cost to produce more complex designs when using third-party IP cores.
3. Ease of integration for available IP cores into more complicated systems.
4. Commercially available IP cores are pre-verified and reduce the design risk.
5. Significant improvement to the product design cycle.
Guidelines for Outsourcing IP Cores

- Outsource IPs from a well-known IP provider with large customer base and great track record.
- Evaluate the IP functionality using demos and executable models before purchasing.
  - Executable models allow you to change parameters and make sure the IP provides expected functional results for your design.
- Ask for a full verification test environment.
  - A set of models for different stimuli to verify the IP functionality.
- IPs should be accompanied by detailed documentation.
  - Data sheet, user manuals, simulation and reuse models, test benches and technology migration guidelines
- Become familiar with the interfaces and functionality of the outsourced IP.
- Agreement with IP provider for technical support during the integration process.
Business Model for IP Core Providers

- Pricing models:
  - Past
    - Vendors sold product as IC to designers
    - Designers must buy any additional copies
      - Could not (economically) copy from original
  - Today
    - Vendors can sell as IP
    - Designers can make as many copies as needed
  - Vendor can use different pricing models
    - Royalty-based model
      - Similar to old IC model
      - Designer pays for each additional model
    - Fixed price model
      - One price for IP and as many copies as needed
    - Many other models used
IP Protection

■ Past
  ❖ Illegally copying IC very difficult
    - Reverse engineering required tremendous, deliberate effort
    - “Accidental” copying not possible

■ Today
  ❖ Cores sold in electronic format
    - Deliberate/accidental unauthorized copying easier
    - Safeguards greatly increased
    - Contracts to ensure no copying/distributing
    - Encryption techniques
      ■ limit actual exposure to IP
    - Watermarking
      ■ determines if particular instance of processor was copied
      ■ whether copy authorized
New Challenges to Users

- Licensing arrangements
  - Not as easy as purchasing IC
  - More contracts enforcing pricing model and IP protection
    - Possibly requiring legal assistance

- Extra design effort
  - Especially for soft cores
    - Must still be synthesized and tested
    - Minor differences in synthesis tools can cause problems

- Verification requirements more difficult
  - Extensive testing for synthesized soft cores and soft/firm cores mapped to particular technology
    - Ensure correct synthesis
    - Timing and power vary between implementations
  - Early verification critical
    - Cores buried within IC
    - Cannot simply replace bad core
Issues in Synthesizing Soft Cores

- Synthesis performed by the core user.
  - However, synthesis process depends on the core provider and the technology provider.

- Challenges for core users:
  1. Handling tool-dependent constructs in the core description.
  2. Handling unsupported megacells.
  3. Handling technology-dependent constraints.
Actual Implementation Process for a Soft Core

![Diagram of the actual implementation process for a soft core.](image)
IP Core Classes

- 3 classes:
  1. Digital IP
  2. Analog IP,
  3. Programmable IP

- Digital IP design process:
  1. Specification and documentation of the reusable IP;
  2. Implementation using standardized coding practices;
  3. Full verification including code coverage and behavioral (or functional) coverage.
Analog IP

- AMS design: ad-hoc custom design process
  ⇒ More time-consuming to develop
- Productivity of AMS design can be improved using a mixed-signal SoC design flow, employing AMS IP.
- AMS IP must provide more flexibility in design parameters and performance characteristics w.r.t. digital IP.
  ✉ Because design specs varies widely between applications,
  ✉ performance of AMS IPs is significantly influenced by parasitics and interactions with the surrounding environment.
  ⇒ Mostly delivered in the form of hard IP and targeted to one application in a specific fabrication technology.
    - not easily migrated to other applications/technologies by end user.
Analog IP (cont’d)

- Hard IP reduces design cycle significantly when the specifications and fabrication processes are identical
  - But not if it has to be modified, or migrated to a new process.
  - Firm IP: the most appropriate format to deliver the AMS IP library components.
    - Firm IP captures suitable schematics of the analog blocks with parameters that are adjustable to optimize the design for specific applications.
    - Allows ease of migration of IP from foundry to foundry, customer to customer, and application to application.

- Verification of mixed-signal SoCs requires cosimulation of analog and digital behavioral models to reduce simulation costs.
Analog IP (cont’d)

- Requirements for a successful IP block:
  - should be parameterized,
  - easily verified through reusable test benches,
  - well documented,
  - contain associated views to ease the design process.
    - a behavioral/analytical view (in AMS-HDL),
    - a parameterized schematic view (transistor level),
    - a layout view (floor plan).
  - test benches are needed to validate the performance under different operating conditions and at various process corners.
    - Used as the basis for verification of specifications and for exploration of the design space for the system.
Programmable IP

- The key to programmable SoC designs: provide some form of flexible
  - hardware (using programmable logic cores) and/or
  - software (using an embedded processor)

- Components of programmable logic cores (hardware):
  - Programmable logic elements,
  - Programmable interconnect,
  - Flexible memory arrays,
  - Dedicated arithmetic blocks,
  - High-speed communication blocks.

- Hardware programmable core can be soft or hard IP.
Programmable IP (cont’d)

- Software Programmability:
  - use of libraries of code and data structures, along with off-the-shelf kernel and real-time operating systems (RTOSs) to improve productivity.
Differences in Design Between IC and IP

- Limitation of IC design
  - Number of I/O pins
  - Design and implement all the functionality in the silicon

- Soft IP
  - No limitation on number of I/O pins
  - Design all the functionality in HDL code, implement desired parts in silicon
  - IP compiler/Generator: select what you want !!
  - More high level auxiliary tools to verify design
  - More difficult in chip-level verification

- Hard IP
  - No limitation on number of I/O pins
  - Provide multiple level abstract model
  - Design and Implement all the functionality in the layout