Approaches to increase IP reuse

Who is motivated to increase IP reuse?

Why do they do it?

How do they do it?
Where Does IP Come From?

Semiconductor Industry

3rd Party Designers

Customer Designs

module func (A,B,C,D,Z,mult,adder);
......Z=A*B + C*D;endmodule
Utility of Reuse - Semiconductor

*Easy integration is desirable for all segments*

Semiconductor industry:

- Profits = \(\text{margin\_per\_design} \times \text{design\_wins} \times \text{volume\_per\_design}\)
- High-value IP implies high margin per design
- Currently, the cost of integrating a significant IP block (e.g., 32-bit RISC microprocessor such as MIPS R4000) is quite high.
- Non-recurring engineering (NRE) cost limits the volume size that can be accommodated within a particular program (>1M parts in many programs)
- Lower NRE costs lower the barriers to utilizing semiconductor IP. More design\_wins means more volume. More volume means more profit.
- Semiconductor companies are also trying to glue customers into their fab
Utility of Reuse - 3rd party IP

Easy integration is desirable for all segments

3rd party IP:

- Profits = fixed_charge_per_design X design_wins + (royalty)
- royalty = royalty_per_design X design_volume
- Since principle source of revenue comes from fixed charges per design, 3rd party IP companies are more highly dependent on design wins than are semiconductor companies, thus they are even more motivated to make integration easy.
- Lower NRE costs lower the barriers to utilizing 3rd party IP. More design_wins means more fixed charges. More charges, more profit.
Utility of Reuse - Customer designs

*Easy integration is desirable for all segments*

Customer designs:

- Focus is on reducing NRE costs for design an integrated circuit
- Lower NRE costs lower the barriers to utilizing 3rd party and semiconductor IP and reduces overall design cost
- In addition, while semiconductor companies wish to use high-value IP (e.g. TMS320CXX) to tie end customers to a fab. End customers, on the other hand, desire NOT to be tied to a particular semiconductor fabrication facility. Variety of IP designs as well as variety of access is desirable.
Summary of Motivation

Across the board, the market forces motivate:

- Providing a wide variety of intellectual property blocks
- In a manner that makes it easy to integrate the IP blocks

OK, so how do we do it?
Technical Challenges to Integration

Verification
  ● Functionality
  ● Electrical correctness
  ● Testability

Performance parameters
  ● Timing
  ● Power
  ● Area
How the integration challenge is faced

Semiconductor company
- Motorola CSIC program - MC68332
- LSI Logic Coreware program
- (TI - cDSP - custom DSP Program)

3rd party IP provider
- (Inventra)
- Tensilica Xtensa Processor
- Synopsys 8051
- Sonics (auxillary slides)

End Customer
- Reuse methodology consulting
Third Party IP Suppliers
The tensilica solution:

- Ultra small and efficient, new architectures
- Fast, safe tailoring of cores
- Extensibility with synchronization to the hardware
- DSP and peripheral blocks
- S/W development environment
- Pre-verified function library

uP Cores
uP Generator
Imagine the possibilities......

Describe the processor attributes from a browser-like interface

Using the tensilica processor generator, create...

Tailored, RTL uP core

Customized Compiler, Assembler, Linker, Debugger, Simulator

Use a standard cell library to target to the silicon process
Prototype and Emulation Support

* Uniform interface to a robust co-verification suite

Cycle Accurate Instruction Set Simulator

Verilog VHDL

FPGA-based HW emulator
Standard EDA tools are supported

<table>
<thead>
<tr>
<th>Function</th>
<th>EDA Tool Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Synthesis</td>
<td>Synopsys, Ambit</td>
</tr>
<tr>
<td>Logic Simulation</td>
<td>Verilog XL/NC, VCS, MTI, Vantage</td>
</tr>
<tr>
<td>Timing Analysis</td>
<td>MOTIVE, Primetime, PEARL</td>
</tr>
<tr>
<td>Floorplanner</td>
<td>HLD</td>
</tr>
<tr>
<td>Place/Route</td>
<td>Si Ensemble, Apollo, Aquarius</td>
</tr>
<tr>
<td>LVS/DRC</td>
<td>DRACULA</td>
</tr>
</tbody>
</table>
DW8051 : The Complete Solution

“Embedded System On A Chip” Design Flow

Evaluation → Development → Simulation → Synthesis → Fabrication → System Debug

**HardWare Development**
- DW8051 Source Code
  - VHDL Source
  - Verilog Source
  - 8031/8032
  - 8051/8052
- Multi-Simulators Tested/Supported
  - VSS
  - Vantage
  - MTI
  - Leapfrog
  - Verilog-XL
  - Chronologic
- Synthesis Support
  - Syn. Scripts
  - Test Compiler Scan Insert
  - 98% Fault Cov.
- Technologies Tested/Ported
  - 0.6Micron CBA Silicon
  - 0.6Micron TSMC CBA
  - 0.6Micron Charter CBA
  - 0.65Micron LSI300K
  - 0.5Micron LSI500K
  - 0.5Micron Motorola H5C
  - 0.5Micron Oki

**Software Development Tools**
- DW8051 CBA Silicon
- DW8051 Daughter Card
- ChipView-51 Software Debugger

**Software Development**
- DW8051 CBA Silicon
- DW8051 Daughter Card
- ChipView-51 Software Debugger

**Complete System Debug & Development Environment Tested**
- Hitex In Circuit-Emulator
- ChipView Software Debugger
- Multiple C-Compilers Tested
- 8051 Assemblers Tested

From Start to Finish We Provide Designers The Complete Solution!!!
Design for Reuse

General Solution
  ● parameterized for various applications
Meets quality goals in multiple technologies
Designed to simulate with common commercial simulators (Verilog and VHDL)

Fully Verified
  ● reuse requires higher confidence levels - 99%
  ● complete test bench for system integration
Why Reuse is Hard

Requires best design practices
Requires generalized solutions
Verification of parameterized designs is hard
Optimal, generalized synthesis is hard
Requires
  ● discipline
  ● processes
  ● tools
  ● additional effort and time
  ● management buy-in that value >> cost
High Quality Documentation

1st Generation

Block Diagrams
Functional Specification
Description of parameters
Interface signal description
Timing diagrams/req’ments
Verification Strategy
Synthesis Constraints

1st Thru n\textsuperscript{th} Generation

- Project it was used on
- Personnel on the project
- Verification Report
- Technology used
- Tools (w/version) used
- Actual Timing/Area Results
- Revision history for any modifications
Example of Poor Partitioning

Separate Application-Specific Logic from Reusable Sections

Reusable Sub-blocks

Reusable Block

App-Specific Glue Logic
Example of Good Partitioning

Clean Partitioning of Glue Logic Away from Main Reusable Areas
Consistent Coding Guidelines are Key

Signal Naming Conventions
Blocking vs Non-blocking ops
Signals vs variables (VHDL)
Sync vs Async resets
Copyright banners
Change history
Indentation
Begin/End closure
Commenting do’s/don’ts
Use of constants/defines
Design for test constructs
State machine coding
Technology portability

4 of many ways to name active low signals. Pick one!

In _n  L_out
Not_clk
Reset_bar
Verification is Key to Reuse

Test Plans
Compatibility
Corner Tests
Random Tests
Synthesis results
Code Coverage
begin
  if(reset_n = '0') then
    -- AD bus
    int_ifc_ad <= (others => '0');

  end if;

  -- C_BE bus
  int_ifc_c_be <= (others => '0');

  -- PAR bus
  int_ifc_par <= '0';
  int_ifc_par64 <= '0';
  elsif (clk = '1' and clk'EVENT) then
    -- AD bus
    if(i2p_ad_ld = '1') then
      int_ifc_ad <= i2p_ad;
    else
      int_ifc_ad <= int_ifc_ad;
    end if;

    -- C_BE bus
    if(ism_cbe_ld = '1') then
      int_ifc_c_be <= ini_c_be;
    else
      int_ifc_c_be <= int_ifc_c_be;
    end if;
DesignWare™ Development Lifecycle

Requirements Phase
- Marketing Requirements Document
- Strict pass criteria at each checkpoint
- Signoff required for each phase
- ISO 9000-like

Planning Phase
- Planning Checkpoint

Development Phase
- Development Checkpoint
- Product and Engineering Specifications
- Technical Specifications
- Tested Si prototype with reports

Verification Phase
- Verification Checkpoint
- Successful customer synthesis and simulation

Rollout Phase
- Rollout Checkpoint
Verification Phase Activities

Design Phase

System Test Plan

- Compliance Suites
- Directed Corner Tests
- Random Tests
- Prototype Silicon

Rollout Phase
Semiconductor Companies
Architectures for Higher Computation Requirements

Example: Motorola MC 683xx - family of controllers

Processor: CPU 32
- 68000 - processor enhanced by most of the 68030 features
- CISC processor: code density
- pipelining
- standard register sets (not in RAM) ⇒ context switch is more expensive
- virtual memory
- supervisor and user modes \{ \text{aims at use of operating systems} \}
- table lookup instructions for compressed tables with built-in linear interpolation ⇒ \text{data density is concern}


control dominated systems

Source: Prof. Rolf Ernst
MC68332

Control dominated systems

Designed for automotive applications with mixture of computation intensive tasks and complex I/O-functions

Idea: off-load CPU from frequent I/O interactions to make use of computation performance: \[\rightarrow\] TPU

Source: Prof. Rolf Ernst
control dominated systems

- independent programmable timer channels: single-shot "capture & compare"
- channel coupling and sequence control with control processor

**Source:** Prof. Rolf Ernst
http://www.lsil.com/products/unit5_5.html

<table>
<thead>
<tr>
<th>Product Family</th>
<th>1K</th>
<th>3.3V</th>
<th>5.0V</th>
<th>G16™</th>
<th>G11™</th>
</tr>
</thead>
<tbody>
<tr>
<td>TinyRISC 16/12-bit Embedded TR111 CPU</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>P</td>
</tr>
<tr>
<td>TinyRISC 16/12-bit Embedded TR110 CPU</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>MiniRISC 12-bit SuperSparc Embedded CW403 CPU</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>P</td>
</tr>
<tr>
<td>MiniRISC 12-bit SuperSparc Embedded CW402 CPU</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td>P</td>
</tr>
<tr>
<td>MiniRISC 16-bit Embedded CW40XX CPU</td>
<td></td>
<td></td>
<td></td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Ouroboros Core CPU 16/32-bit Stand-Alone CWDSF1600</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ouroboros Core CPU 16-bit Stand-Alone CWDSF1600</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gigabit Transceiver</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Meriva Fiber Channel Protocol Controller</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Virtex Decoder</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Real Solution Decoder</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet-3 Controller (1000Base TP-PMD), 10 Mbps</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIFRED-16 Ethernet Transceiver, Encoder-Decoder, 10 Mbps</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet-10 Mbps, 10/100 Mbps</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SONET/SDH Interface 155.52 Mbps</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARM® Thumb Processor</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TI Power</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSDL</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet-10 Serdes, 10/100 Mbps</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethernet-100BASE-T 10/100 Mbps</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCI-Express Transfer</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For a list of additional cores, contact your local sales representative. A = Available C = Consult Marketing P = Planned Technology

---

<table>
<thead>
<tr>
<th>Product Family</th>
<th>LCB120K</th>
<th>LCB160XX</th>
<th>G16 Products</th>
<th>G11 Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-bit Sample &amp; Hold DAC 10 MSPS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4-bit Low Power Flash DAC 10 MSPS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6-bit Flash ADC 65 MSPS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6-bit Flash ADC 90 MSPS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8-bit Flash ADC 40 MSPS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10-bit Successive Approximation ADC 350 KSPS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Triple 10-bit RGB Video DAC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10-bit Low Power DAC 10 MSPS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>10-bit Low Power Multiple Output DAC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sample and Hold Output Stage for 10-bit Low Power Multiple Output DAC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Programmable Frequency Synthesizer 300 MHz</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SONET/ATM 155 MSPS PMD Transceiver</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>155 and 267 MBPS High Speed Backplane Transceiver</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ethernet 10BASE-T 10BASE-T 100Base-TX 100Base-FX, 5-V</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ethernet 100BASE-X Clock Generation/Data Recovery Function, 2-V</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Note: Most of the mixed signal cores listed above will be ported to the G11 product family. Contact your local sales representative for schedule information.
OakDSPCore CWDSP1650
ARM Core7 Thumb Embedded
ARM/LSI’s Features to Ease Integration

• *synthesizable* version of the widely popular ARM7TDMI processor core

• “LSI Logic’s RTL-level approach, implemented with the CoreWare design methodology, results in the fastest time-to-market for system-level ASICs in the industry. “

• “LSI Logic’s ARM cores are also supported with a rich set of peripherals easing the implementation of a complete CPU subsystem in a large system-level ASIC. This peripheral library, including complex peripherals like SDRAM controllers, is constructed around the open AMBA standard facilitating IP reuse.”

• “These building blocks have been designed to facilitate customization to specific requirements.”
ARM7 core
ARM’s Amba open standard

Advanced System Bus, (ASB) - high performance, CPU, DMA, external
Advanced Peripheral Bus, (APB) - low speed, low power, parallel I/O, UART’s
External interface

http://www.arm.com/Documentation/Overviews/AMBA_Intro/#intro
More Features to Ease Integration

- LSI Logic’s ARM cores are supported with the ARM software development toolkit.
- This integrated development environment, available on multiple hosts, comes with the full ARM compiler suite and complete debugging environment including
  - the ARMulator, cycle accurate instruction set simulator, which supports simulation in an off-line mode or can be connected via a JTAG port to the target application for real time debugging even in large, complex ASICs.
Even more Features to Ease Integration

The development software is complemented with the standard PID7T H/W evaluation board provided by ARM or LSI Logic’s AMCU development board which features a highly integrated MCU showing the peripherals available in the library.
End Customer Integration
Who will “manage” your Internal IP?

An effective strategy requires some degree of internal coordination

Some possibilities:

- R&D group
- CAD group
- Document Control group
- Quality group
Organizing for Reuse-Based Design

Need to do this

Else

ASIC Group

IP Repository

support team

reuse team

block design chip design

block design chip design

ASIC Group

IP Repository

block design + reuse chip design

IP support

 ASIC Group

IP Repository (empty)

block design chip design
**IP Repository - Heart of the EcoSystem Infrastructure**

![Diagram of IP Repository and Management System](image)

- **IP Repository**
  - User Interface
  - Web I/O Infrastructure
  - Configuration Management
  - IP Protection, Encryption, Licensing

- **Database**
  - M-Core:
    - RTL
    - SWIFT model
    - GDSII
    - ISA model
    - Timing model
    - Floorplan model
    - SPW model
    - AppBuilder
  - PCI:
    - RTL
    - Synth Scripts
    - Test Scripts
    - BFM
    - SPW model
    - AppBuilder
  - PLL:
    - GDSII
    - Behavior model
    - Timing model
    - Floorplan model
    - SPW model
    - AppBuilder

- **IP Management System**
  - Qualification
  - Distribution
Summary

All segments are motivated to increase design reuse

- 3rd party IP supplier
- Semiconductor
- End system customer

Approaches to increasing reuse

- Increasing breadth of capability
- Increasing ease of integration
  - easy to verify function
  - easy to verify timing
  - easy to verify electrically
- generally, easy to integrate in tool flow