

Automatic IP Block Selection with IP-Explorer Technology

You can use the AccelDSP synthesis tool to select from various macro-architectures based on your system requirements.

by Tom Hill
System Generator Product Manager
Xilinx, Inc.
tom.hill@xilinx.com

Developing DSP algorithms for silicon requires careful selection of IP blocks and their macro-architecture specifications in the context of the target application. For example, three methods are commonly used to implement basic trigonometric functions in hardware:

1. Bipartite tables – a good choice when input values require relatively small word lengths.
2. Linearly interpolated look-up tables (LUTs) – best for slightly larger input word lengths.
3. CORDIC – ideal for applications that require large word lengths when performance is not an issue. If performance is important, you can implement the CORDIC algorithm using parallelism and pipeline stages.

The graph in Figure 1 compares the area of these methods over a range of input word lengths.

Knowing how to build and when to use each of these cores is a complex task, but necessary to achieve an optimal hardware solution.

Introducing IP-Explorer

The Xilinx® AccelDSP™ synthesis tool with IP-Explorer technology eliminates the trial-and-error process when using IP blocks by allowing the tool to select from various

macro-architectures. The AccelWare™ DSP IP tool kit provides a set of IP generators that produce architecture-specific, synthesizable MATLAB for common built-in functions such as sine, cosine, log, and divide. These hardware architectures, developed by a team of DSP hardware design experts, are characterized by area and performance for each supported FPGA technology using more than 6,000 designs.

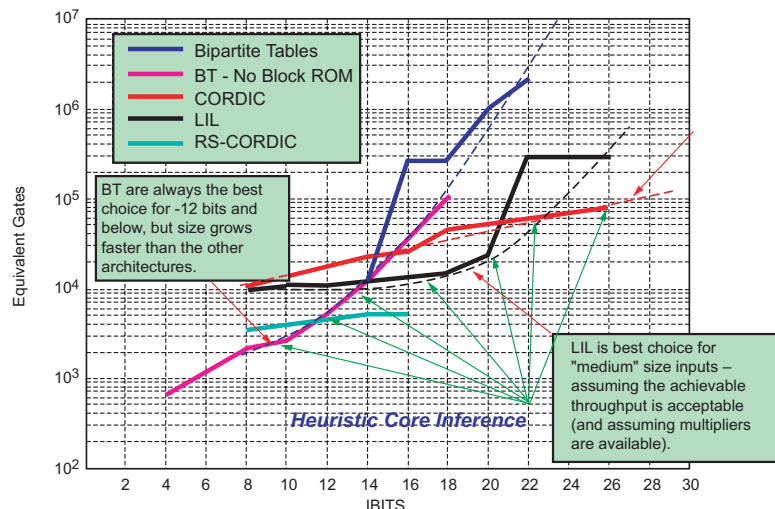


Figure 1 – Area versus input word length

During the DSP synthesis process, IP-Explorer analyzes each use of these functions against the overall system area and performance requirements to determine the most optimal hardware solution. Once determined, the tool automatically inserts the appropriate IP core into the design (Figure 2).

Designing at a Higher Level

IP-Explorer raises the abstraction level of DSP hardware design significantly closer to that of DSP algorithm development. Basic

MATLAB language primitives form the foundation of this abstraction pyramid (Figure 3) by providing the ability to model any function or algorithm at a low level. IP-Explorer elevates this abstraction level significantly by targeting MATLAB's standard library of pre-configured DSP building blocks and functions directly into hardware.

Design Example

To illustrate the effect that IP-Explorer has on a design, consider the example shown

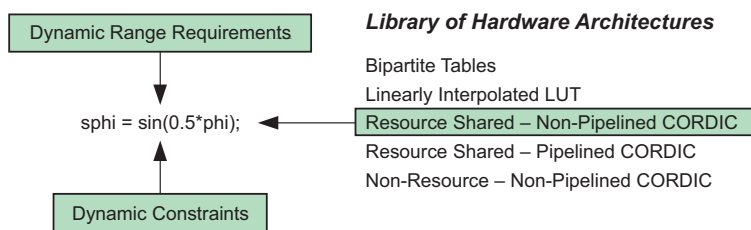


Figure 2 – Design context-based automatic IP insertion

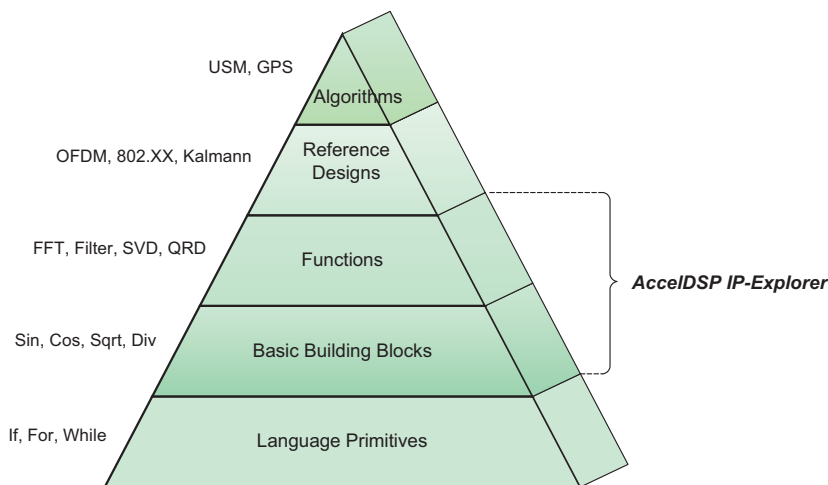


Figure 3 – DSP design abstraction pyramid

```
function [q] = euler2quat(phi,theta,psi)
```

```
sphi = sin(0.5*phi);
cphi = cos(0.5*phi);
stheta = sin(0.5*theta);
ctheta = cos(0.5*theta);
spsi = sin(0.5*psi);
cpsi = cos(0.5*psi);
```

```
q(1) = sphi*stheta*spsi+cphi+ctheta+cpsi;
q(2) = sphi*ctheta*cpsi-cphi*stheta*spsi;
q(3) = cphi*stheta*cpsi+sphi*ctheta*spsi;
q(4) = cphi*ctheta*spsi-sphi*stheta*cpsi;
```

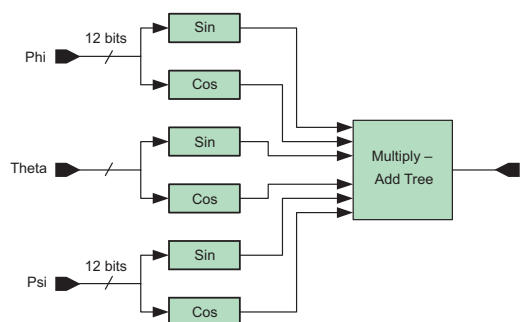


Figure 4 – Euler-to-quaternion angle converter block diagram

Sine/Cosine Architecture	Area (LUTs)	Performance
Six CORDIC Architectures	2,670	8.4 MSPS
Selected by IP-Explorer	1,287	100 MSPS

Table 1 – Area and performance results for CORDIC and IP-Explorer-selected approaches

in Figure 4: a Euler-to-quaternion angle conversion algorithm that performs a sine and cosine operation on three input angles. In this example, each input has a word length of 12 bits.

Let's compare a typical approach – designing a single implementation of the trigonometric function based on the CORDIC method and instantiated multiple times – versus IP-Explorer. Table 1 shows the area and performance results for both approaches.

IP-Explorer selects a bipartite table approach because the input bit widths are reasonably small. A secondary benefit is that this approach leverages the built-in DSP blocks of the targeted FPGA device. This would not have been the optimal choice for an ASIC. Not only has the task of designing hardware for the sine and cosine functions been eliminated, but this approach offers a 33% area savings and a 12x performance advantage.

Conclusion

IP-Explorer represents a truly unique advancement in the development of DSP hardware. By providing a seamless path to hardware for key DSP building blocks, hardware design has moved significantly closer to the abstraction level enjoyed by algorithm developers using MATLAB. 🌈

TAKE THE NEXT STEP
 (Digital Edition: www.xcellpublications.com/subscribe/)

- Learn more about the AccelDSP synthesis tool.
- Listen to the free recorded lecture, "AccelDSP Jump Start Modules."