

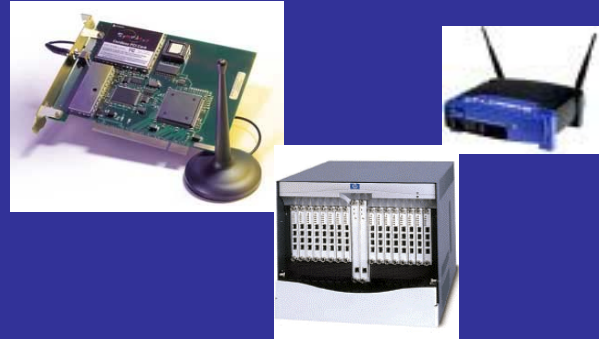
EMBEDDED APPLICATIONS DRIVE DEMAND FOR SOC PLATFORMS

Most computationally intensive real-time applications such as voice-over-IP, video-over-IP, 3G and 4G wireless communications, MP3 players, JPEG and MPEG encoding/decoding, require an integrated hardware/software platform for optimal performance. Parts of the application run in software on a general purpose processor and other portions need to run on application-specific hardware to meet performance requirements. This has caused an increasing number of software applications to be migrated to System-on-a-Chip (SOC) platforms.

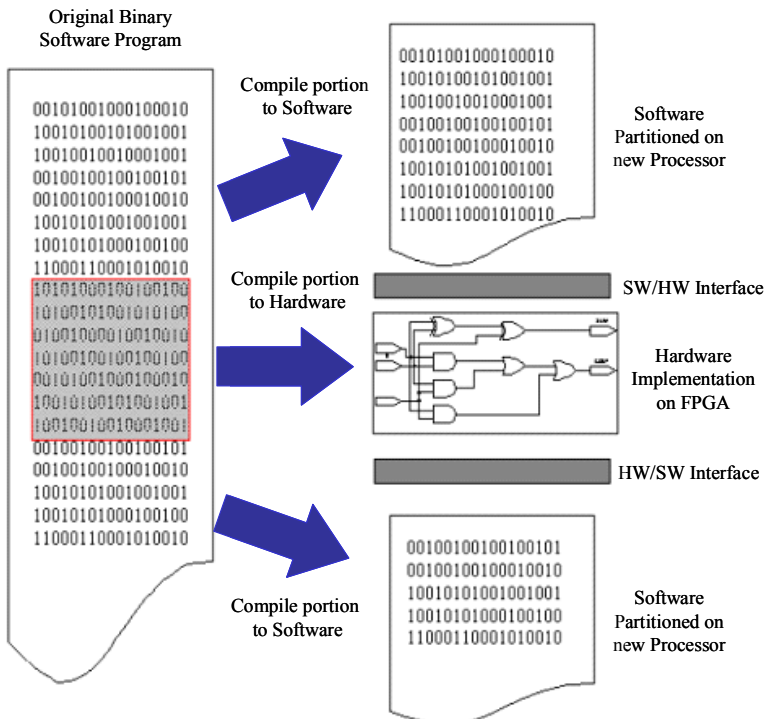
FPGAs have emerged as the SOC platform of choice, particularly in the fast-paced world of embedded computing. They enable rapid, cost-effective product development cycles in an environment where the target markets are constantly shifting and standards continuously evolving.

BINACHIP-FPGA enables embedded systems developers using Platform FPGAs to make hardware/software trade-offs for optimal performance. It also allows seamless migration of software from older general-purpose embedded processors onto hardware and mixed hardware/software SOC platforms of the future.

Networking

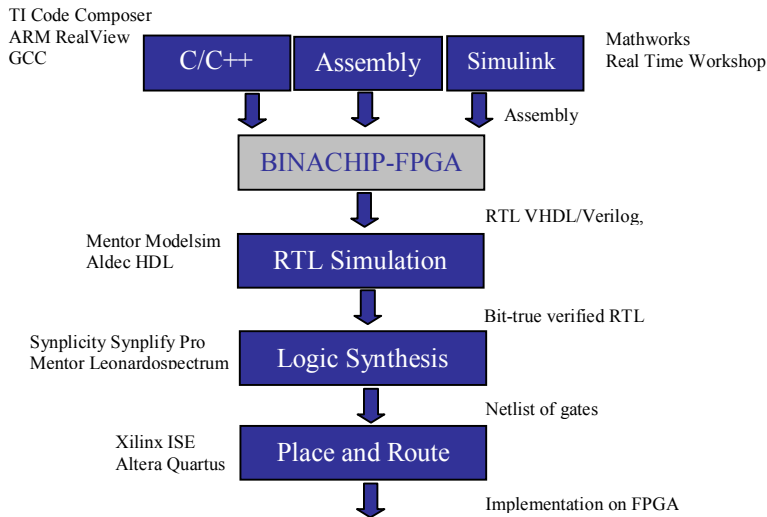


Embedded Computing



HARDWARE/SOFTWARE CO-DESIGN USING BINACHIP-FPGA

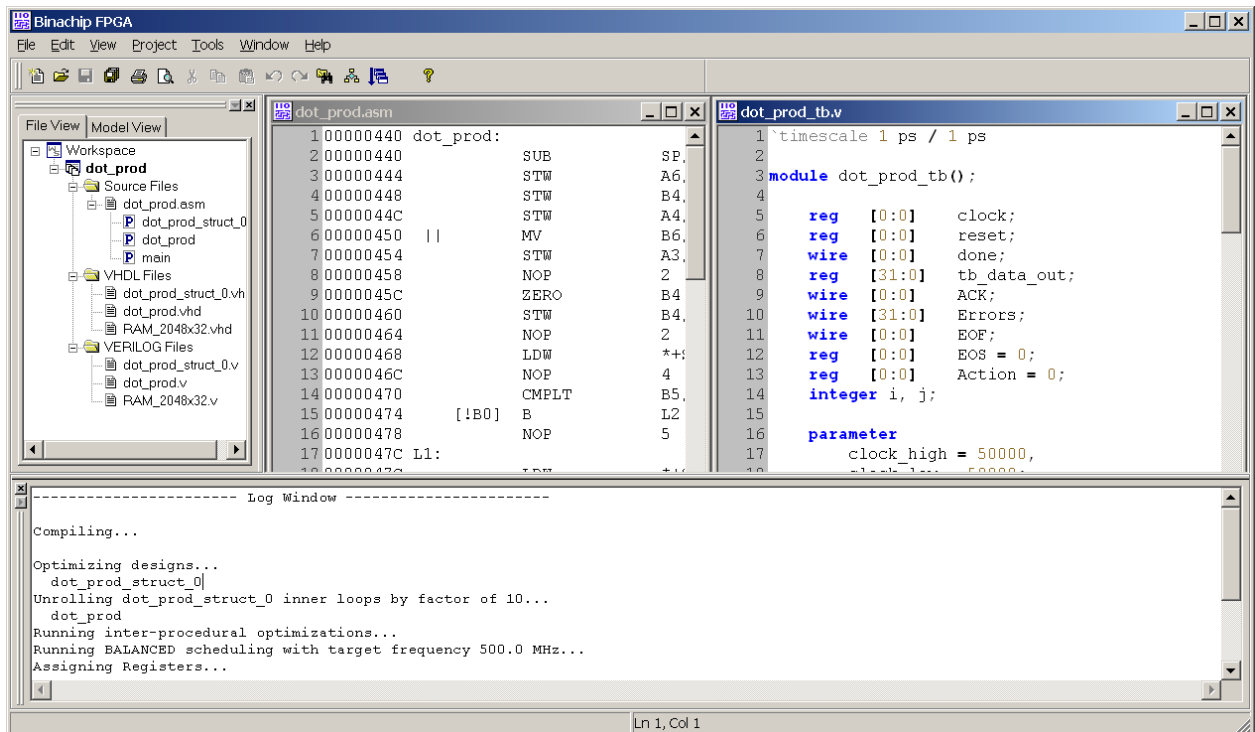
The adjacent figure shows an illustration of the hardware/software co-design flow using BINACHIP-FPGA. On the far left is a binary program that needs to be optimized for performance. The code is first profiled using standard profiling tools. If it is determined that a portion of the code (shown in grey) will benefit from a hardware implementation, it is automatically compiled into hardware using BINACHIP-FPGA. In addition, the appropriate hardware/software interfaces are generated. The remaining code segments are translated into binary for the target processor. Depending on the application, the resulting implementation can provide a 10X to 50X speedup over a pure software implementation.



BINACHIP-FPGA TOOL FLOW

PRODUCT HIGHLIGHTS

- ✓ Binaries from multiple sources
 - C/C++, MATLAB/Simulink, Manual Assembly
- ✓ Multiple source processor instruction sets
- ✓ Support for multiple target FPGA's
- ✓ Outputs RTL in either VHDL or Verilog
- ✓ Supports hardware/software partitioning
- ✓ Allows designer to perform area/delay trade-offs
- ✓ Generates test-benches for bit-true verification
- ✓ Advanced Optimizations
 - Procedure extraction
 - Loop unrolling
 - Memory and data partitioning
 - Advanced scheduling techniques
 - Register allocation
 - Control and data flow optimizations



BINACHIP-FPGA USER INTERFACE