www.binachip.com

Company Background

What We Do

BINACHIP develops and markets software products and services that enable embedded systems developers to design and implement high-performance applications such as HD video CODEC, within their price/performance and time-to-market constraints. It also enables seamless migration of software from older generation, general-purpose embedded processors into newer hardware, and mixed hardware/software platforms of the future.

Problem/Solution

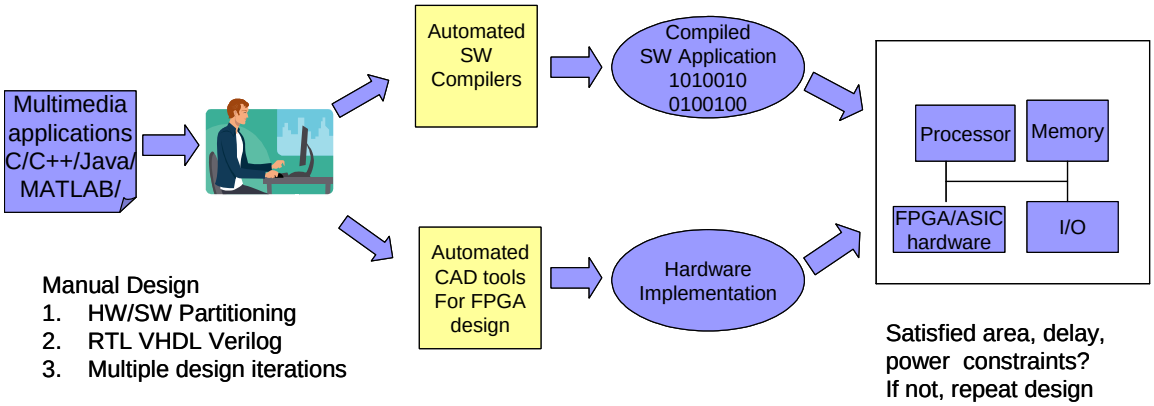
Increasing demands for cell-phones, PDAs, and network devices provide opportunities for the growth of embedded software, operating systems, and development tools. Software and hardware developers are under increasing pressure to develop a large number of more complex embedded software and hardware projects with much tighter price, performance and time-to-market constraints.

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. They enable rapid, cost-effective product development cycles in an environment where the target markets are constantly shifting and standards continuously evolving. BINACHIP enables embedded systems developers 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.

In any complex embedded application, there are parts that can run on a general purpose processor in software, and other portions that need to run in hardware to accelerate the tasks that run too slowly in software. The partitioning of an application into hardware and software is called hardware/software co-design and is illustrated in Figure 1(a). The state of the art in mixed hardware/software platforms is as follows. A designer decides which part of a given application written in C/C++/MATLAB/Java should be mapped to software running on a general purpose processor, and which part should be mapped onto application specific hardware. For the part to be mapped onto general purpose processors, C/C++/MATLAB compilers are used to generate code for a particular

processor. For the part that needs to be mapped to hardware, a hardware designer takes the C/C++/MATLAB language specification of the problem and manually converts it into a Register Transfer Level (RTL) Hardware Description Language either VHDL or Verilog. If the good hardware specification of the problem is available, it is possible for a hardware designer to implement the RTL model in a few weeks. However, in most cases there is a large application implemented in software, part of which needs to be converted to hardware for optimal performance. It takes an average hardware designer several months to manually translate this software implementation to a hardware implementation, and then encode this in RTL VHDL or Verilog.

(a) CURRENT DESIGN FLOW



(b) BINACHIP AUTOMATED FLOW

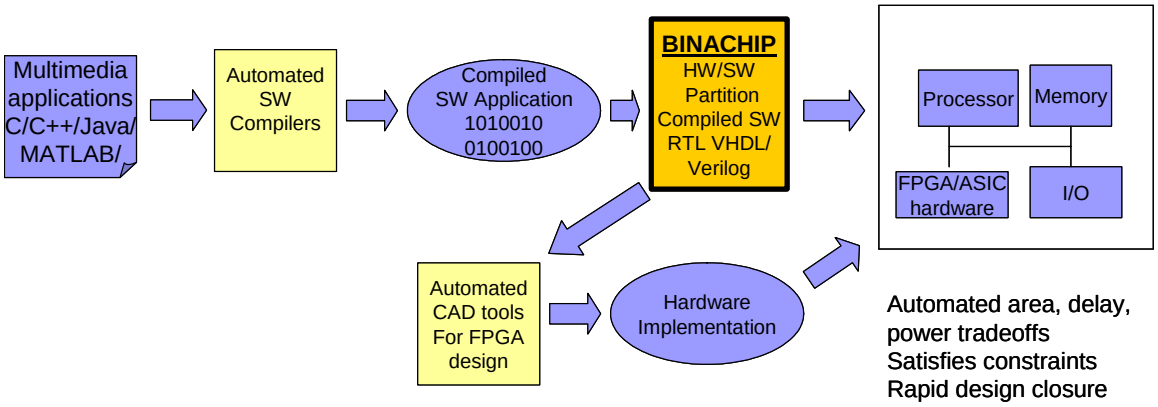


Figure 1. Overview of current and BINACHIP automated design flows

In contrast, BINACHIP automates the translation of software assembly and binaries onto mixed hardware/software platforms, thereby **reducing design times from months to hours**. The given application can be available at a high level such as C/C++ or MATLAB and compiled into a general purpose processor assembly or binary, or the code may be available in binary form from previous legacy designs. BINACHIP takes this software code, and performs automated hardware/software co-design at the assembly

language level, and generates software code for the target processor, and RTL VHDL and Verilog code for the FPGA as shown in Figure 1(b).

Figure 2 shows an illustration of the hardware/software co-design flow using BINACHIP. 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. 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.

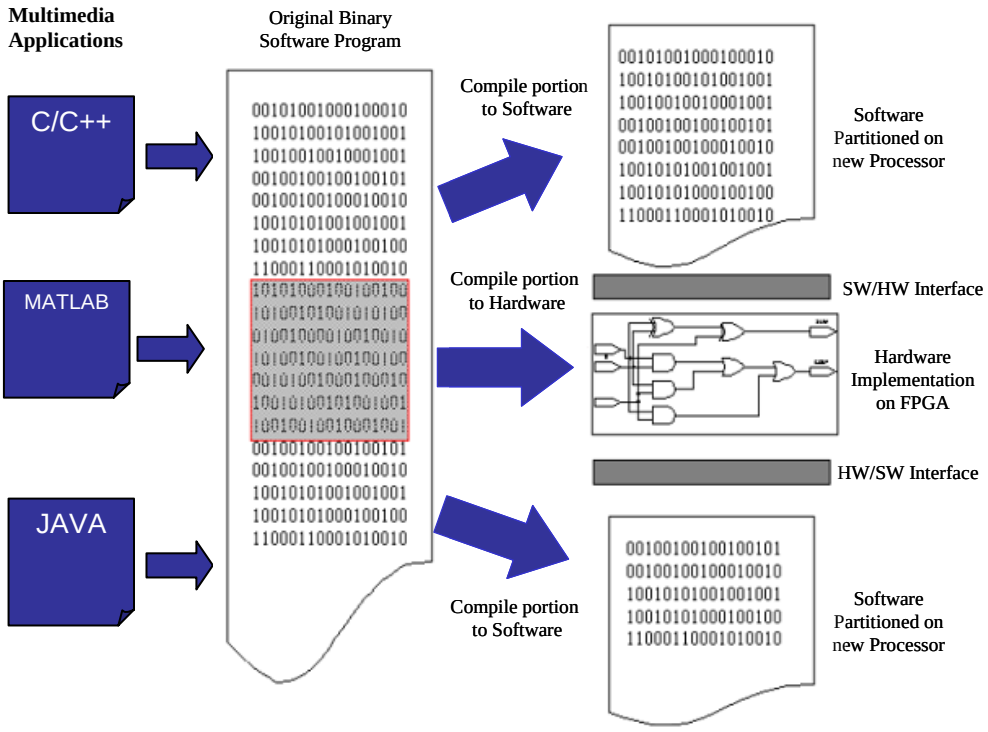


Figure 2. BINACHIP hardware/software co-design

Value Proposition

Using BINACHIP products, designers of high-performance embedded applications can *reduce their development time by up to 6 months*. Currently, the software migration task from older generation processors to newer processors is done manually. It takes a team of 4-5 developers about 6-12 months to port a large piece of software from one processor family to a new hardware/software implementation. In case of a hardware implementation, designers have to translate a software implementation into RTL VHDL or Verilog and then use automated logic synthesis and physical synthesis tools to design an FPGA or ASIC. There are no automated tools available in the market for the proposed BINACHIP products.

Technology Background

BINACHIP'S core technology is the result of work done at Northwestern University by Professor Prith Banerjee (BINACHIP co-founder, Chairman, and Chief Scientist) and his graduate students. This team developed the FREEDOM compiler that takes digital signal processing (DSP) applications written in the assembly language of a Texas Instruments C6000 DSP processor, and automatically generates RTL VHDL and Verilog code that is mapped onto commercial FPGAs from Xilinx and Altera. Patented in 2004, BINACHIP now holds an exclusive license for the technology.