

ADVANCED TOOLS IN RECONFIGURABLE COMPUTING

HW/SW Co-Design Languages/Tools can be broadly classified into **high level languages** and **extending HDL languages**.

High level languages- C-FPGA environment

Direct mapping of C code to configuration level is possible. The software supports emulation and simulation of compiled code for debugging. The software is also capable of handling multiprocessor and multi-FPGA computational definitions. Generally this environment allows explicit data flow control within memory hierarchy. It can produce “Unified Executables” for HW or SW processor execution. The runtime libraries handle required interfacing and management. Some of the important C based mappers used for SoC designs are described below. [5]

- **SpecC:** This is an extension of ANSI-C. Both behavioral and structural hierarchical embedded systems designs are supported. The language is developed for synthesis and verification in mind. But language is targeted for system-level design language intended for specification and architectural modeling.
- **HardwareC:** This modified C language uniformly incorporates both functionality and design constraints.
- **SystemC from Open SystemC Initiative (OSCI):** This C language very much popular in the system level verification industry. It uses C++ class libraries and simulation kernel for creating behavioral and RTL designs. This is Open-source extension of C++ for HW/SW modelling. It includes modules and ports for defining structure, and interfaces and channels. The language supports functional modelling of the design. The language supports hierarchical decomposition of a system into modules. It provides structural connectivity between modules using ports. Events help in scheduling and synchronization of concurrent processes.
- **Catapult C from Mentor Graphics:** this is an algorithmic synthesis tool for RTL generation. It can generate RTL from pure C++. It does not require any extensions, pragmas, etc. The compiler uses “wrappers” around algorithmic code. This gives capability to manage external I/O interface. Internally it can constrain synthesis to optimize for chosen interface. Architectural constraints and optimization are explicitly mentioned. The tool can generate output RTL netlists in VHDL, Verilog, and SystemC.
- **DIME-C from Nallatech:** This is a FPGA prototyping tool. The designs are not cycle-accurate. But allows application synthesis for a higher clock speed. The compiler

includes IEEE-754 FP cores. It has dedicated integer multipliers. The compiler supports pipeline or parallel optimization. Output produced is synthesizable VHDL and DIMETalk components.

- **Handel C from Celoxica:** The tool provides good environment for cycle-accurate application development. All operations occur in one deterministic clock cycle. This feature makes it cycle-accurate. On the flip side of it clock frequency is reduced to slowest operation. Explicitly defined parallelism is achieved in language by the help of pragmas. The compiler can efficiently analyze, optimize, and rewrite code. The output generated from the compiler is VHDL or Verilog, SystemC, or targeted EDIFs.
- **Impulse C from Impulse Accelerated Technologies:** Sequential applications can be modelled using this language. It can process independent, potentially concurrent, computing blocks. Utilizing Streams-C methodology it can communicate and synchronize processes. It also focuses on compatibility with C development environments. Compilation is carried out by considering each process as separate state machine. Output can be generated in two ways-either generic or FPGA specific VHDL.
- **Mitrion C from Mitrion:** The language uses concept of “Soft-core” processor. The soft-core processor creates abstraction layer between C code and FPGA. Compilation of C code is achieved by mapping it to a generic “API” of possible functions. Specific application oriented processor is instantiated on FPGA. The soft-core processor supports custom instruction bit-widths, specific cache and buffer sizes. The language can produce outputs as VHDL IP core for the selected target FPGA architectures.
- **Napa C from National Semiconductor:** This Language/compiler is intended for RISC or FPGA hybrid processors. It capitalize on single-cycle interconnect instead of I/O bus. It uses datapath synthesis technique. From C loops the compiler generates hardware pipelines. The language basically targets National Semiconductor NAPA1000 hybrid processor. It has Fixed-Instruction Processor (FIP) and Adaptive Logic Processor (ALP) which allow programmer to specify whether the section of the code is to be executed in software or hardware. [5]. Outputs generated from the compiler are RTL VHDL, structural VHDL and structural Verilog.
- **SA-C from Colorado State University:** This is a High-level, expression-oriented, machine-independent, single-assignment language. This is designed to implicitly express data-parallel operations such as image and signal processing. The compiler can perform loop optimizations, structural transforms and execution block placement.
- **Streams C from Los Alamos National Laboratory:** This is stream-oriented sequential process modelling language. Data elements are moved through discrete functional blocks. The compiler can generate multi-threaded processor executables and multiple FPGA bitstreams. The language allows parallel C program translation into a parallel design. Attractive feature of this language is this includes functional-level simulation environment. The compiler produces synthesizable RTL output.
- **Java Based Approaches-JHDL:** Java HDL language is capable of converting Java into synthesizable HDL code. The JBits Application Programming Interface (API) is implemented in the Java programming language and permits programmatic access to all of the configurable elements in Xilinx Virtex-II FPGAs. JBits 3.0 complements Xilinx's industry leading ISE software tools and enables the design and generation of partial bitstreams for reconfigurable applications.

- **UML: Unified Modeling Language:** This approach is used for System level modeling. UML is extended so that it can be applied as high level models working with other languages. UML profile for SystemC- this is another extension of the available languages that enables specify, analyze, design, construct, view software and hardware artifacts in a SoC design flow. Extending HDL Languages.
- **System Verilog:** this language blends both Verilog and C. This language is an extension to IEEE 1364-2001 Verilog. The language supports interfaces that allow module connections at a high level of abstraction.
- **SuperLog:** This language is Verilog superset that includes constructs from C. Verilog 2001 and SuperLog at two ends of the spectrum. For a very productive design process the language utilizes power of C with the simplicity of Verilog. MATLAB to FPGA Design IP Core Instantiation. Xilinx has DSP system generation tool.
- **Khorus:** This is a special data flow tool for image processing.
- **Ptolomey:** This is a graphical entry tool for system level design.

Spectrum of C-based application mappers are shown in Figure (1) and Figure (2).

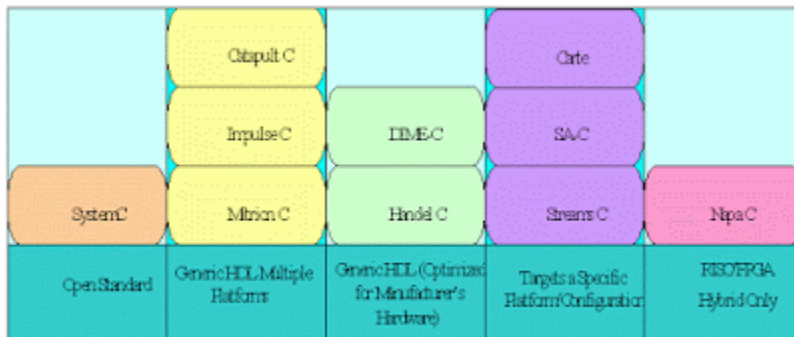


Figure (1) C-based Application Mappers 1 [2]

Code Available				VHDL
Definable			Hardware C	
Language Primitive		DIME-C Impulse C		
ANSI-C	ANSI-C			
	Coffee	Code HDL Program	Code HDL Program	HDL

Figure (2) C-based Application Mappers 2 [2]

Advantages of C-based application mappers

High level languages are popular for many years and skilled engineers are available in this segment. This enables potential RC users with high-level languages. Due to this required HDL knowledge is significantly reduced or eliminated. Time to preliminary results is much less than manual HDL designs methodology. Software-to-hardware porting is considerably easier due to the partitioning of the design. Visualization of C hardware is far easier compared to the HDLs for engineering community. Understanding of any C based mappers is straight forward.

Disadvantages of C-based application mappers

Mapper instructions are several times more powerful than CPU instructions. On the other hand FPGA clocks are many times slower. Due to the hardware nature of mappers they parallelize and pipeline C code, however they generally cannot automatically instantiate multiple functional units. Manual parallelization of existing code using techniques pertinent to algorithm's structure is necessary to optimize C-mapper code. Performance may degrade at the benefit of reduced development time. Available tools are not capable of automatic translation of codes. Programs still require manual assistance for hardware compilation. Optimized software C is not efficient as optimized Hardware C.

At least two major challenges remain to be addressed. Input/output interfaces become a limiting factor as once a generic I/O wrapper is generated, it should be reusable. True hardware debugging remains a challenge. Since there is no idea internal HDL signals higher level

abstraction increases complexity. C and C++ family of languages mainly deal with 32 bit precision which is overhead for most embedded applications. [5]

Advanced Hardwares

There are several hardware platforms available for reconfigurable computing applications. They provide advanced features which are outperforming the standard processors. Some of them are mentioned below.

Xilinx

The Xilinx XC6200 FPGA-based Reconfigurable Co-processor provides open architecture FPGA (XC6200). They provide advanced dynamic reconfiguration capability such as high-speed reconfiguration via parallel CPU interface, full or partial reconfiguration or context switching and unlimited re-programmability.

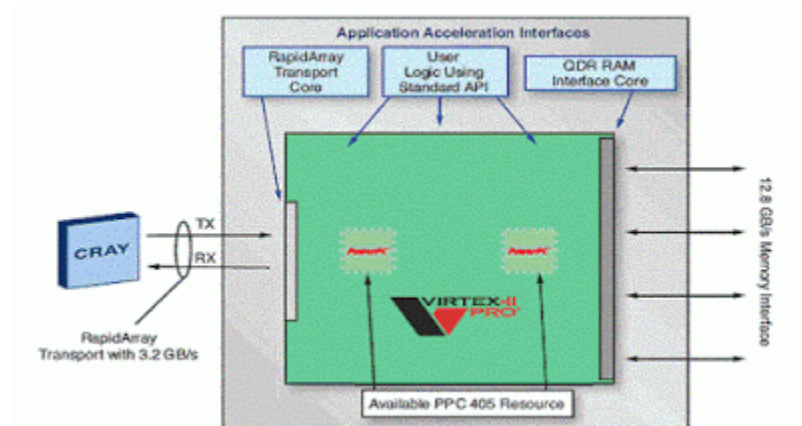


Figure (3) Xilinx-based application acceleration module [9]

Xilinx Vertex series of processors have 4 Power PC processor embedded within. They support Microblaze soft core processor. Reconfigurable computing has got a major step with the Cray XD1 high-performance computer (see Figure (4)) by breaking down performance barriers at substantially lowered cost by using off-the-shelf components from Xilinx to solve difficult computational problems. [9]

Altera

Altera have coprocessor based approach and example FPGAs is ARC-PCI, Compaq Pamette. They have system on chip solutions such as Excalibur device, ARC-PCI. Excalibur device provide embedded processors like ARM, MIPS or NIOS.

Future of Reconfigurable Computing

Several advancements in the field of reconfigurable computing hardware and software are promising greater design flexibility and reduced cost and time to market in embedded system development environments. But the lack of efficient software tools, simpler design methodologies and standards are hindering the complete development of the SoC designs. Engineering community is still not much familiar with the developments of reconfigurable computing domain. High level language based C mappers offer easiness in software development but ignorance of HDL and hardware design methodologies may result in inefficient design which can de motivate the basic power of SoC designs. Advanced FPGAs like Vertex 4 (from Xilinx) and Excalibur (from Altera) are in forefront to provide reconfigurable solutions with third party support.

Related Articles

- Reconfigurable Computing
- Advantages and Disadvantages of Reconfigurable Computing

Source : <http://asic-soc.blogspot.in/2007/11/advanced-tools-in-reconfigurable.html>