

DTTF and SANKHYA Tools Collection

1. Introduction

Compilers and debuggers have been around for several decades now. The nineties have seen several advancements in the tools in areas like optimization. Today compiler tools handle a variety of languages, machine architectures – real and virtual.

The continuous advancements in semiconductors have finally led system designers to design and manufacture whole systems on a single chip. Often such chips are assembled from off the shelf CPU cores and IP blocks. SoC's have presented several challenges to the tools developers.

Often, a CPU core can be configured or reconfigured for a particular SoC design. A new hardware acceleration module may be added reducing a whole FFT operation or an FIR filter into a single customized CPU instruction! The number of functional units may be configured at system development time.

Sankhya has been aware of these challenges and has been conducting research and development for more than 4 years to create a technology that will help tools vendors to create highly configurable tools, tools that system developers can configure for a new processor variant -- instead of depending on their tool vendor. This will open up a whole new system design paradigm, where a designer can go beyond co-design and co-simulation to unified-design and simulation in the future!

2. Dynamically Targetable Tools Framework (DTTF)

The Dynamically Targetable Tools Framework is a result of the research

and development efforts at Sankhya Technologies. DTTF[3] includes a processor modeling language SMDL[2] (SANKHYA Machine Description Language) and a set of C++ class libraries that can be used to build tools like code generators, assemblers and simulators.

DTTF allows the following key capabilities to be built into a new toolchain developed using DTTF:

- a) Use of a unified processor model with no redundant information
- b) Dynamic targeting of tools to a processor variant at execution time.

2.1 Unified SMDL Processor Model

Tools like GNU GCC use a processor model. Another tool will process such a processor model and generate a set of C programs that can be compiled and linked with the architecture independent parts of the compiler to create a compiler "retargeted" for particular processor architecture.

However, the GNU GCC machine description (processor model) is only used by the code generator -- not by the simulator or assembler, which have their own processor models.

However, a toolchain built using DTTF will have a single processor model shared by all components of the toolchain including the a) Code generator b) Assembler c) Simulator and d) Disassembler. Please refer to fig (i).

2.2 Dynamically Targetable Tools

Tools built using DTTF need not be "retargeted", because they are not

really "targeted" for processor architecture in the first place. The tools (in their binary form) exist with nearly no information on any particular processor model. The tool can load a processor model at run-time and dynamically target itself for that processor.

This is a major feature, which allows SoC designers to update the Processor model and simulate the new version of the processor within minutes!

3. SANKHYA Tools Collection™ (STC)

STC[1] is a growing list of tools including a code generator, assembler, linker, librarian, object file dumper, simulator, simulator server and a debugger built using DTF. STC tools are ideal for use in the following situations:

- a) When a new processor is being designed
- b) When a processor variant is being designed
- c) When a reconfigurable processor is used for system design
- d) For exploring different designs for a processor
- e) In general when a new tool chain is being developed.

4. Comparison Chart

NOTE: STC is not yet a complete toolchain. In particular a C Front-End is still under development. An optimizer has not yet been developed.

Compilers Features	SANKHYA Tools Collection	GNU GCC
Languages	C	C, C++, FORTRAN, Java, Objective-C
Processors	MIPS32, ARM4	Several Processors
Unified Model	Yes	No
Dynamically targetable	Yes	No
Optimizer	Planned	Available
Dynamic Transformations	Yes	No
GUI Debugger	Yes	Yes
System Simulation (C++ Model)	Yes	No
Multi-CPU Simulation	Yes	No
Extensible for Hardware Acceleration	Yes	No
End-User Controlled targeting for a processor Variant	Yes	No
Availability	Now*	Now

*: C Front-End under development. Optimizer is planned.

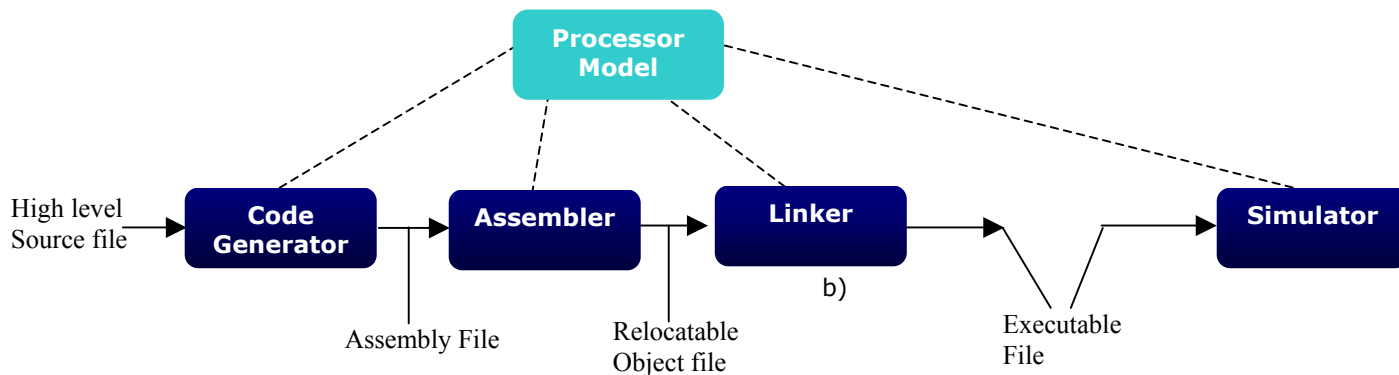


Fig (i) SANKHYA Tools Collection

5. Conclusion

DTTF is a novel framework for creating dynamic model driven tools for embedded systems development. STC demonstrates the power of DTTF. Additional tools like disassemblers, decompilers, binary translators, VHDL generators and opcode space analysis and optimization tools can be developed using DTTF.

6. Technology Partners

Sankhya welcomes proposals from researchers from leading industry and academic research organizations for extending DTTF and developing additional DTTF powered tools.

7. References

- [1] SANKHYA Tools Collection user guide and reference manual

www.sankhya.com/info/products/tools/docs/stc.pdf

- [2] SANKHYA Machine Description Language reference manual

www.sankhya.com/info/products/tools/docs/smdl.pdf

- [3] SANKHYA Translation Framework user guide and reference manual

www.sankhya.com/info/products/data/docs/stf-ugrm.pdf

SANKHYA™

Sankhya Technologies Private Limited

#30-15-58, Third Floor, "Silver Willow",
Dabagardens, Visakhapatnam 530 020
India.

Tel: +91 891 554 2666

Fax: +91 891 554 2665

Toll Free (within India): 1600 12 4477

Email: research@sankhya.com

Sales: sales@sankhya.com

Website:

www.sankhya.com/info/research.html

This document may contain forward-looking statements that are subject to change without notice. SANKHYA, SANKHYA TECHNOLOGIES, SANKHYA Tools Collection, Dynamically Targetable Tools Framework, SANKHYA Debugger and SANKHYA Software are trademarks, service marks or registered trademarks of Sankhya Technologies Private Limited. All other brands and names are the property of their respective owners.