Ptolemy Code Generation for Texas Instrument's TMS320C6x

Sresth Kumar

Vikram Sardesai

Hamid Rahim Sheikh

- **Motivation**
  - Current Electronic Design Automation (EDA) tools are geared towards *implementation independent* design of heterogeneous systems.
  - *Ptolemy* is one such framework that supports simulation and code generation domains.
- **Objective**
  - To add Code Generation (CG) functionality for Texas Instrument's TMS320C6x.
  - To evaluate performance issues of our approach versus C code generation domain (CGC).

Approach

- Derive the new domain from existing CGC domain.
- Use a library of optimized C callable assembly routines for actor functionality.
- Leverage on *optimized* benchmark kernels provided by Texas Instruments(TI) for common Digital Signal Processing (DSP) blocks.

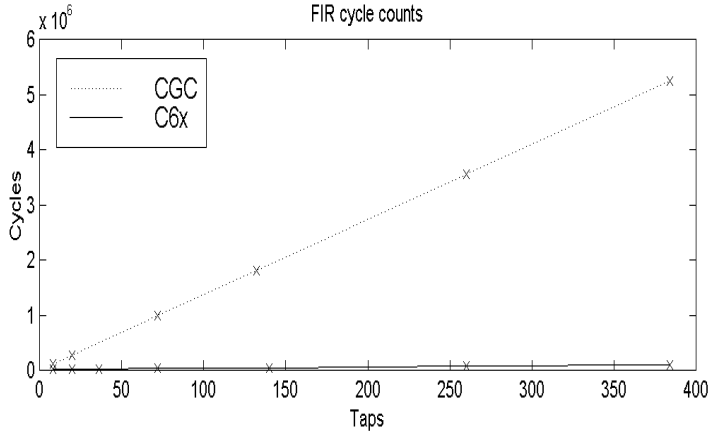
Implementation

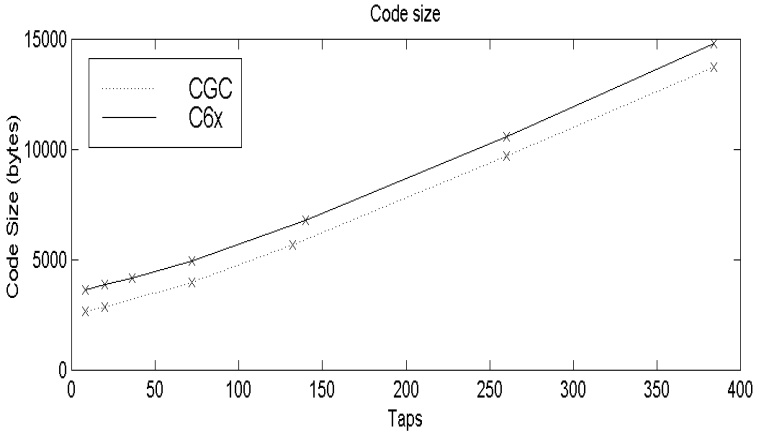
- C6xTarget for CGC.
- DSP blocks FIR, IIR, decimators & interpolators.
- Fixed-point (C62x) and Floating point (C67x) stars.
- Stars have the same constraints as TI's optimized assembly routines.
  - Data alignment issues
  - Data length issues

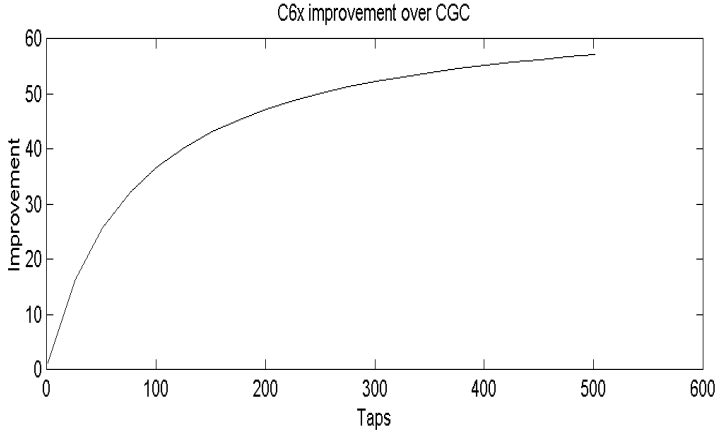
Performance

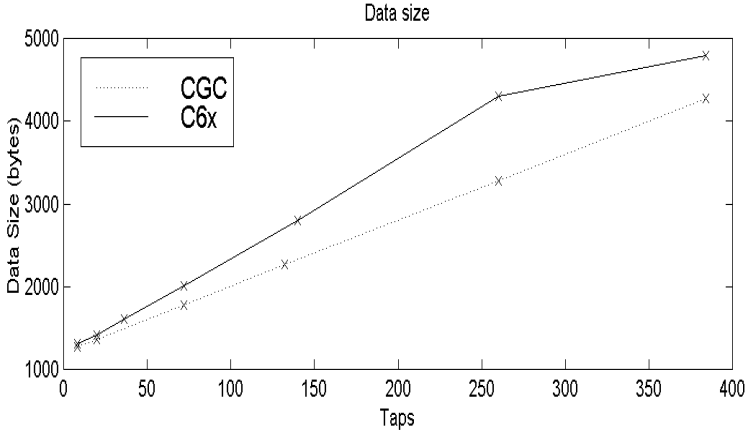
- We compare performance against code generated using CGC stars.
- Cycle counts, Code Size and Data Size comparisons.
- Comparison made for Floating Point stars only for consistency.
- Different implementation structures.
- Goal: Test CD to DAT converter.

Finite Impulse Response Filters

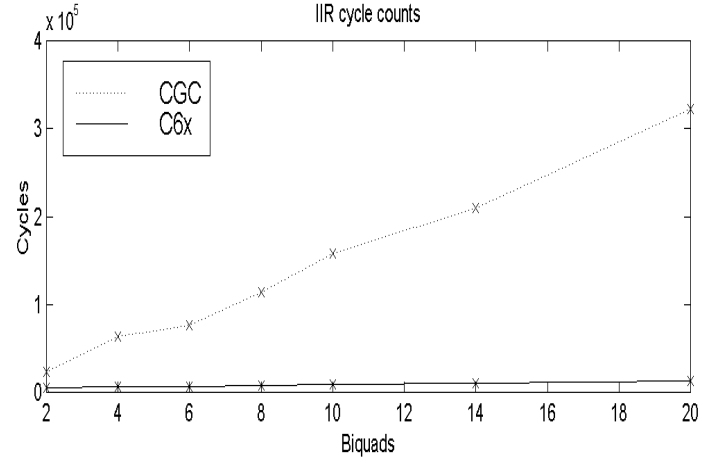


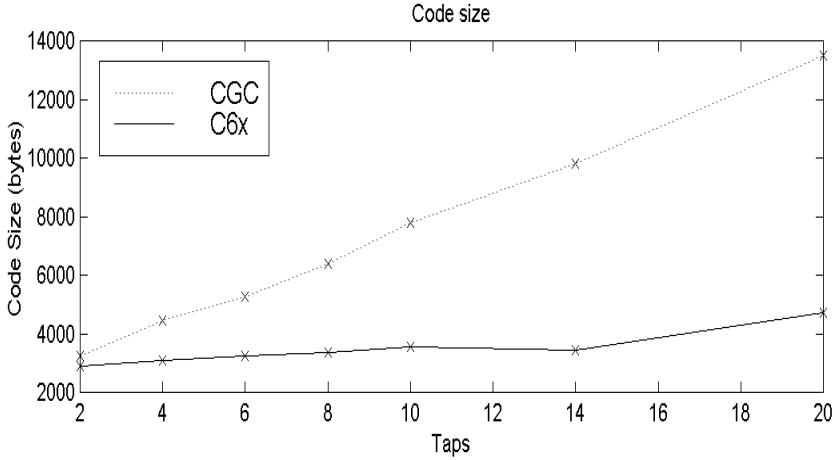


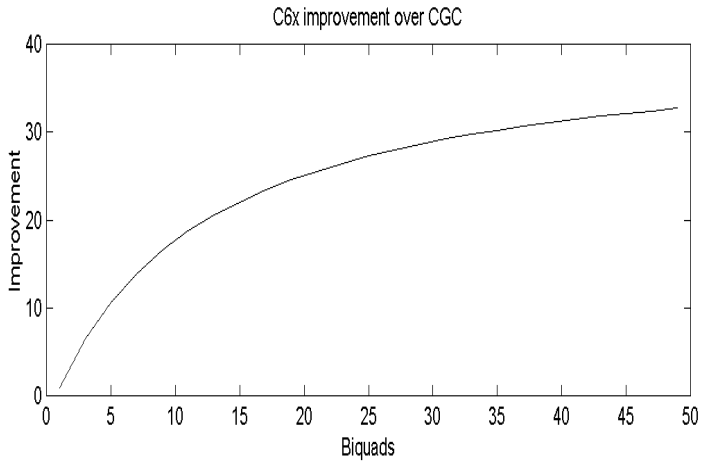


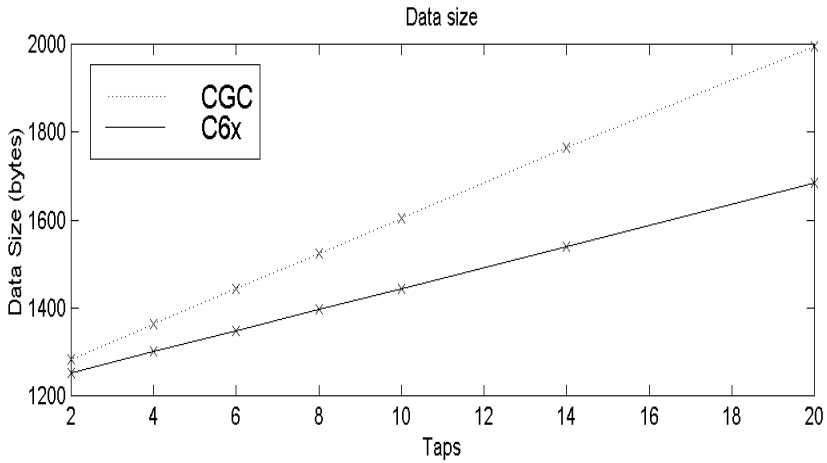


Infinite Impulse Response Filters









Results

Category	FIR	IIR
Cycle Count	10x to 40x up to 66x	4x to 18x up to 40x
Code Size	Fixed Overhead for C6x	Linear overhead for CGC
Data Size	Linear Overhead For C6x	Linear overhead for CGC

Conclusion

- Processor complexity increasing much faster than code generation & compilation techniques.
- Compilers perform poorly when it comes to understanding the global structure of a system.
- Hand optimized library based approach is better than general C-code generation.