

# **Part 6**

## **Design and Development Tools for Digital Signal Processors**

- Development environments
- Debugging facilities
- Assembly language tools
- C development tools

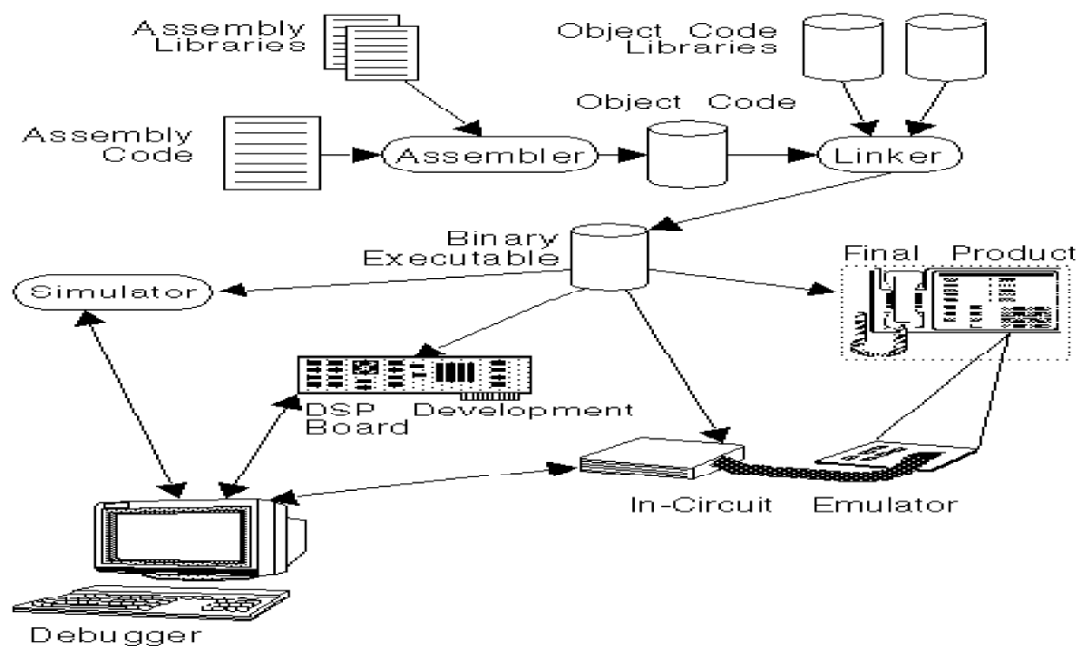
## **Development Environments**

# Development Environments

- DSP development tools
    - C compiler
    - assembler
    - assembly language libraries
    - linker
    - instruction set simulator
    - debuggers
    - in-circuits emulators
    - development boards
- Software
- Hardware

227

## Development Tools



228

## Development Tools

---

- No matter how powerful a processor may be, a system designer requires good development tools to efficiently tap the processor's capabilities.
- Because the quality and sophistication of development tools for DSP processor varies significantly among processors, development tool considerations are an important part of choosing a DSP processor.

229

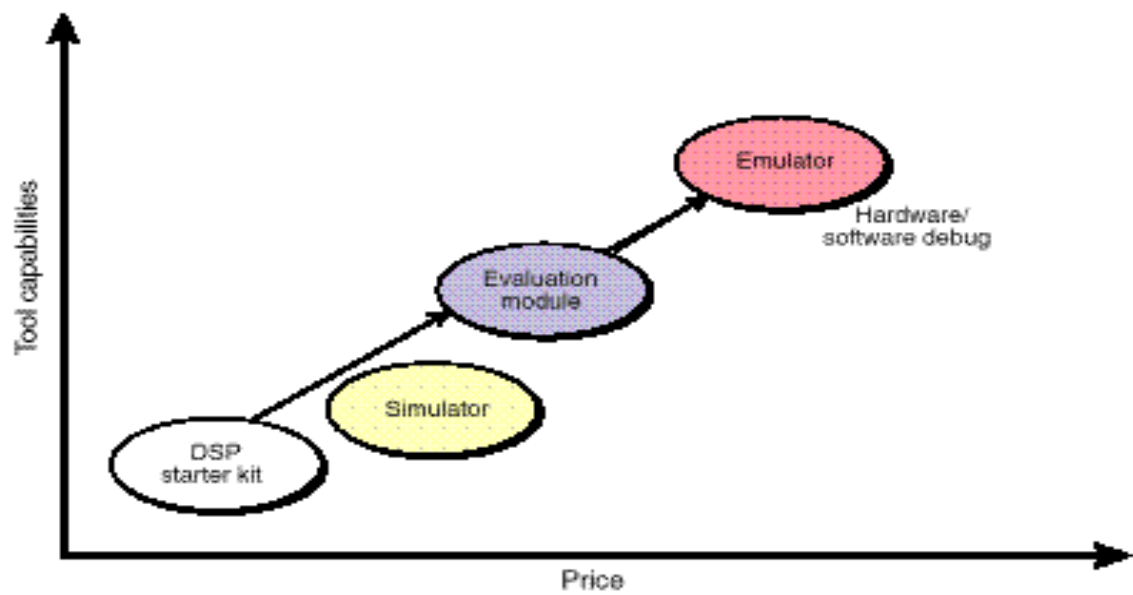
## Development Flow

---

- Write code in a high-level language (such as C) or assembly language.
- Code can be debugged on one of several debug platforms (simulator, emulator), all of which feature the same debugger interface.
- SW/HW can be developed in parallel by using the compiler, assembler/linker, simulator, evaluation module, and by using the processor behavioural models and emulators for hardware development.

230

# Tool Performance Comparison



231

## Debugging Facilities

# Debuggers

---

- A term use to describe the front-end program that provides the user interface and much of the functionality of an emulator or instruction set simulator.
- A critical tool for software and hardware development.
- Emulators share the same debugger as the vendor's instruction set simulator (users do not need to learn different interfaces for the two tools).

232

## TMS320 Debugger's Interface

---

- Brings new levels of power and flexibility to embedded systems development.
- C/assembly source debugger is the standard interface for the simulators, evaluation modules, and emulators.
- Supports 'C2xx, 'C3x, 'C4x, 'C5x, 'C54x, 'C6x, and 'C8x tools.
- Provides complete control over programs written in C or assembly language.

233

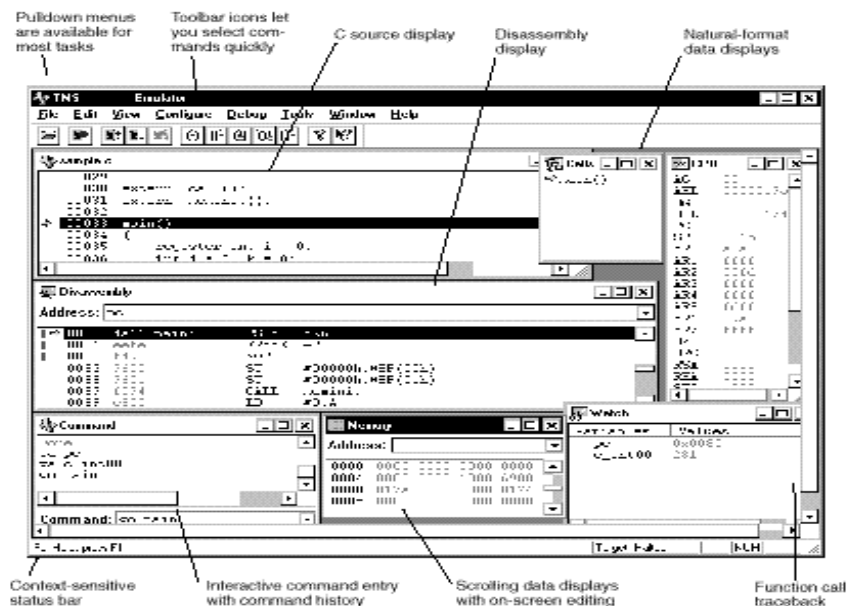
# Debugger Features

- Helps to improve productivity by enabling user to debug a program in the language in which it is written.
- Programs can be debugged in C, assembly language, or both.
- Has profiling capabilities that show where to focus development time by quickly identifying the hot or time-consuming sections of a program.
- Menu-driven window and mouse-oriented interface reduces learning time and eliminates the need to memorise complex commands.
- A shortened learning curve and increased productivity reduce the software development cycle, speeding products to market.

234

## Debugger Features ...

- Custom-made displays and flexible command entry let user develop a debugging environment that suits the systems needs.



235

## Other Features

---

- Multi-operation support
- Multilevel debugging
- Fully configurable, window-oriented interface
- Flexible command entry
- On-screen editing
- Continuous update
- Comprehensive data display
- Powerful command set
- Compatibility
- Simplicity
- Calls stack window
- Memory window

236

## Software Simulator

---

- A simulator is a software program that simulates the microprocessor and microcomputer modes for cost-effective software development and program verification in other than real time.
- Allows user to debug the algorithms/routines without target hardware.
- Time-critical code, as well as individual portions of the program, can be tested.
- The clock's counter allows loop timing during code optimisation.

247



# Simulator Features

---

- Input and output files can be associated with the port addresses of the I/O instructions to simulate I/O devices connected to the processor. In this way, specific I/O values can be used during test and debug.
- Breakpoints can be established according to read/write executions or instruction acquisitions.
- Interrupt flag can be set periodically at a user-defined interval for simulating an interrupt signal.

248



# Program Execution Features

---

- Breakpoints can be defined and trace mode set.
- Internal registers and both program and data memories can be inspected and/or modified when execution suspended.
- Trace memory can also be displayed.
- A record of the simulation session can be maintained in a journal file so it can be re-executed to regain the same machine state during another simulation session.

249

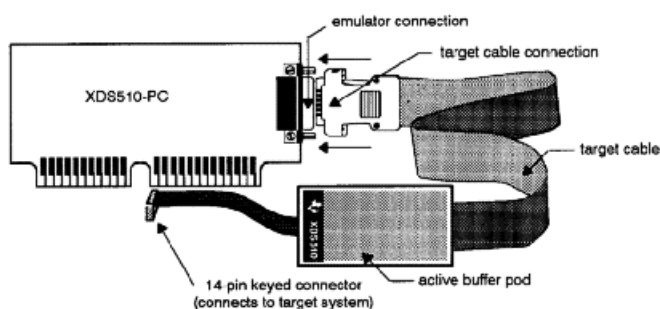


# In-Circuit Emulation

- User can use a General purpose computer to emulate the DSP processor and later replace the emulator with the DSP processor.
- Assists the user in debugging and optimising applications running on the DSP in the target system.
- External hardware and software on a personal computer or workstation provide the user with the ability to monitor and control the processor in the target system as it executes application programs.
- Provide the user with the ability to single-step through a program and to view and edit the contents of registers and memory.
- Ability to trace program flow and pin activity in real-time, the sophistication of real-time and non-real-time breakpoint conditions supported, and profiling capabilities.

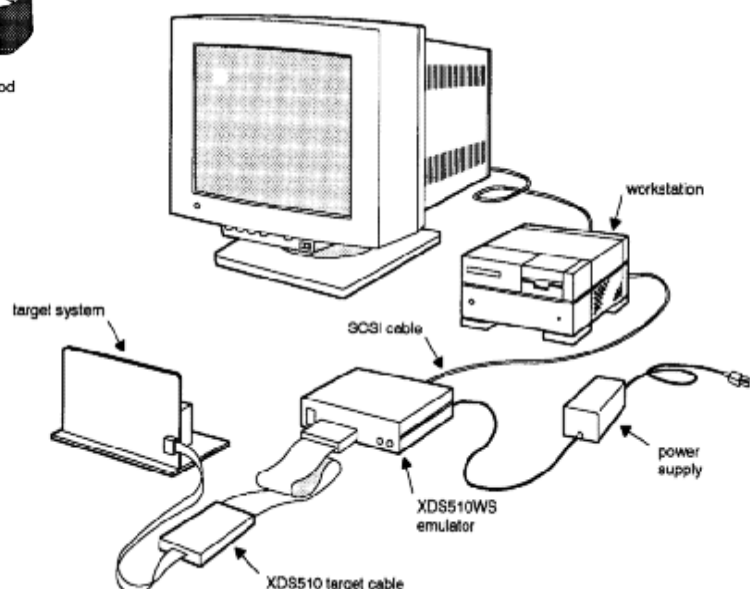
252

## Connecting to Target System



PC connection

WS connection



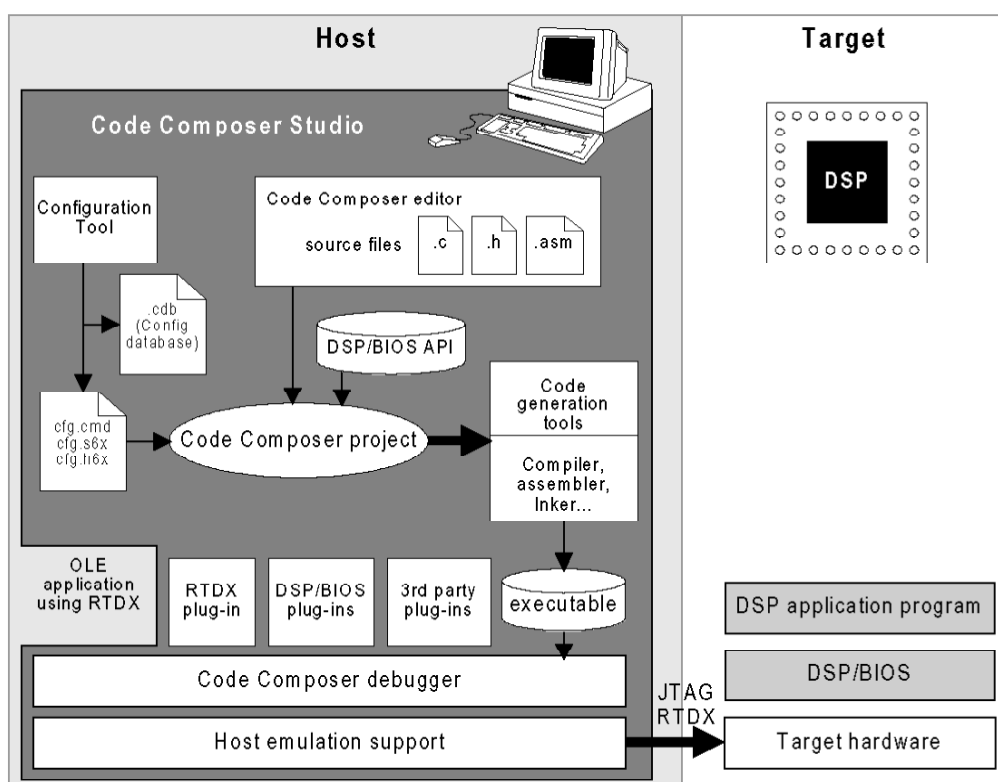
253

# TMS320 Code Composer Studio

- Code Composer Studio is an integrated development environment (IDE) provided by the TI. *(A trial version can be downloaded from the TI website).*
- It provides software simulation with a real-time debugger.
- Also supports hardware emulation with real-time data exchange between the host PC and the DSP processors.

254

## Code Composer Studio



255

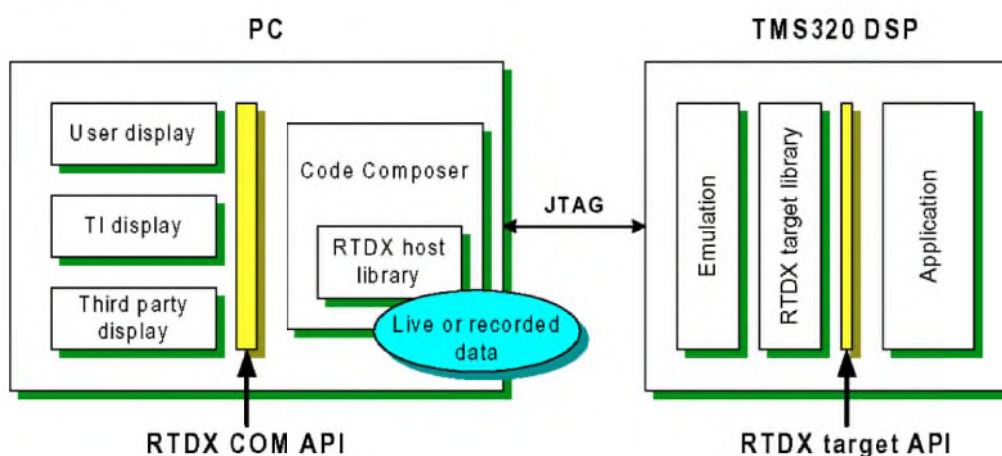
# CCS – IDE, Features

- Application Building
  - Create software application programs by adding files to an object: C source files, assembly source files, object files, Libraries, link command files, include files.
- Debugging:
  - Create software Breakpoints, watching variables, view & edit memory & registers, view call stack, graphing signals on the target, profiling execution statistics.
- DSP/BIOS
  - During the analysis phase of the software development cycle, traditional debugging features are ineffective for diagnosing subtle problems that arise from time-dependent interactions.
  - With DSP/BIOS support, one can perform real-time analysis, such as probing, tracing and monitoring, with minimal impact on real-time performance

256

## ... Features

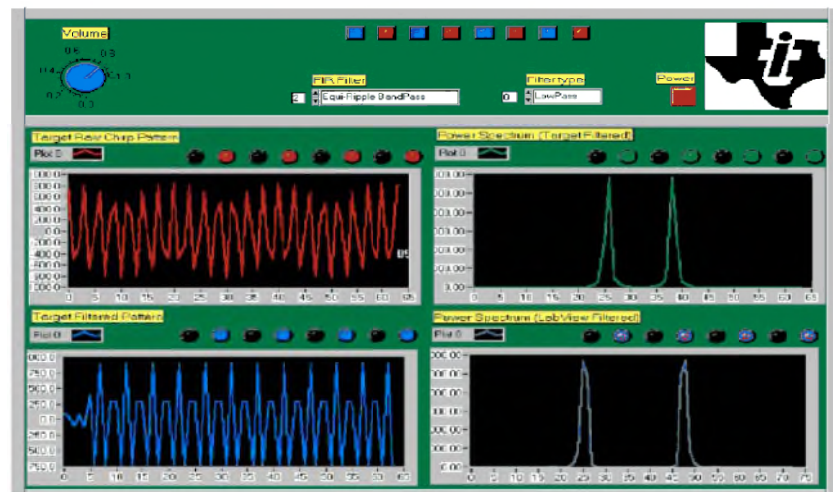
- Hardware Emulation and RT Data Exchange (RTDX)
  - TI DSPs has on-chip emulation support (JTAG link)
  - Capabilities: Starting, stopping or resetting the DSP, Loading code or data into DSP, Examining the register or memory of DSP, Hardware instruction or data-dependent breakpoints, RT data exchange between host and the DSP.



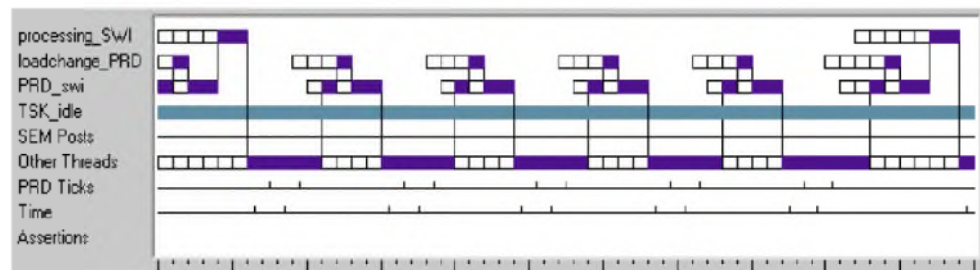
257

# RT Data Exchange and Analysis

Real - Time  
Data  
Exchange



Real - Time  
Performance  
Analysis

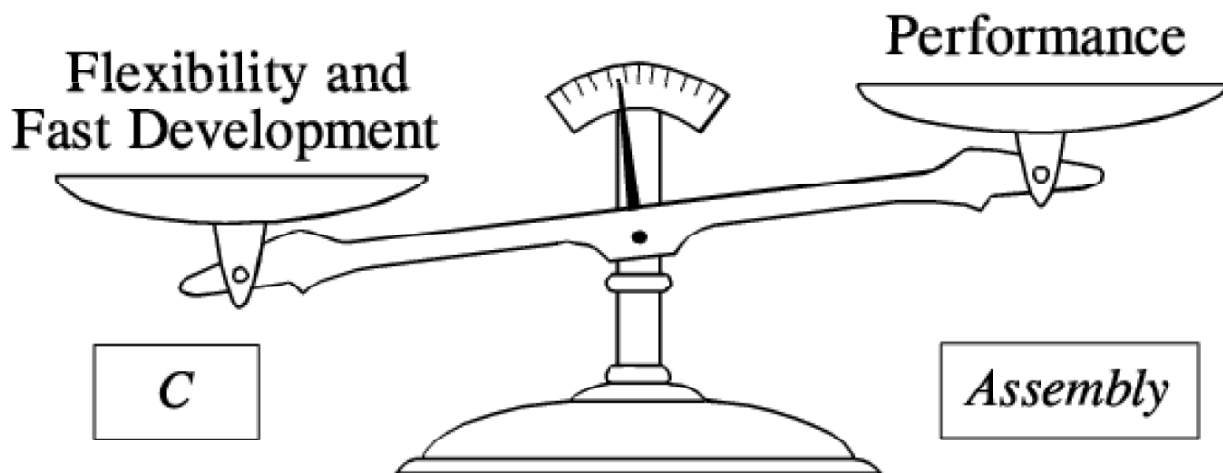


258

## Assembly Language Tools

# C or Assembly?

---



239

## Assembly Language

---

- DSP systems are often subject to extremely demanding production cost and performance requirements.
- Software for their programmable components often must be extremely efficient.
- Software written in assembly language places heavy emphasis on optimisation and is usually developed from the ground up for each new system.

242

# Assembly Language Approach

---

- Incurs high development costs.
- It is a time-consuming and difficult process.
- Irregular instruction sets and highly specialised features of DSP complicates programming (especially true for fixed-point DSPs).

Note: As in so many aspects of the design process, good tools can make a dramatic difference in the productivity of DSP software developers.

243

## Assembler

---

- Assembler translates processor-specific assembly language source code file into machine language object files. The machine language is based on COFF(common object file format).
- Object code requires the additional steps of linking and relocation to transform it into a binary executable file that can be run on the target processor.
- The most basic DSP software development tools.
- They are also among the most important, since a vast amount of software for DSP processors is developed in assembly language to optimised code speed and size.

244



# Assembly Code Libraries

---

- An extremely valuable tool for shortening software development time and improving software quality.
- Software developers are able to reuse optimised blocks from a library without modification.
- Well tested and documented libraries can result in a huge time saving.
- Existing library blocks are useful as examples of efficient coding style.

250



## Two Categories of Libraries

---

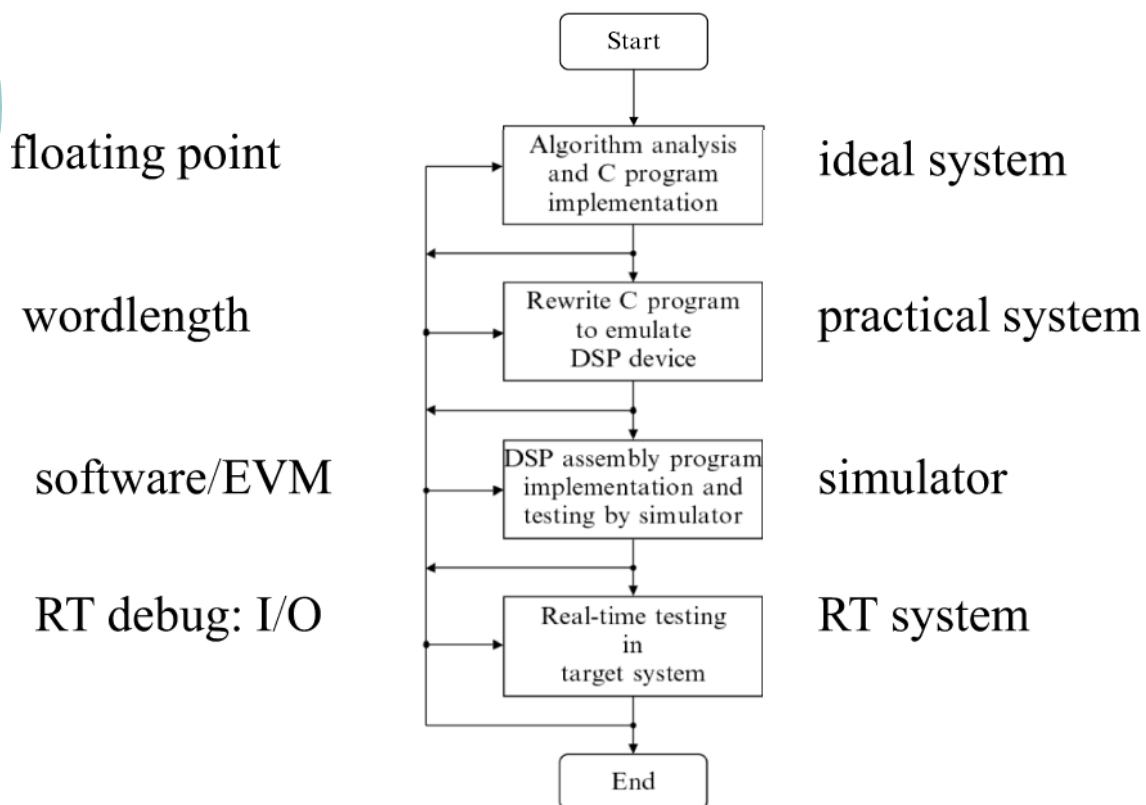
- *Function libraries* consist of relatively small building block functions, such as matrix multiplication, filtering, and I/O drives.
- *Application libraries* contain much larger blocks of code that implement complete applications or parts of applications.

*Note that libraries written in assembly code are often usable with users' C code as well as assembly code.*

251

# C Development tools

## Algorithm Implementation Procedure for Real-Time Applications





# Algorithm Development

---

- Algorithm description using difference equations, or signal-flow diagrams, data flow diagram. Sequence of operations to be performed in flowcharts or structured descriptions.
- Developing DSP S/W on a GP computer:
  - Using HL languages such as MATLAB, C/C++, or other DSP software package can significantly save development time
  - Easy to debug and modify programs
  - I/O operations based on disk files are simple to implement and the behaviors of the system are easy to analyze
  - Using floating-point data can achieve higher precision
  - With fixed-point simulation, verification of an algorithm in fixed-point DSP can be easily conducted

238

## C Compiler

---

- C language is the most popular HL tool for evaluating DSP algorithms and developing RT software for practical applications.
- C compiler translates the C source code into assembly source code first. The assembly code is then given to the assembler for generating machine code.
- Optimiser to improve run-time and code density efficiency
- Supporting ANSI C and its run-time-support library

240



## C Compiler

---

- C language lacks specific DSP features, especially those of fixed-point data operations that are necessary for many DSP algorithms.
- To improve compiler efficiency for RT DSP applications, it's necessary to have the capability of adding in-line assembly language routines directly into the C program.
- This allows to write highly efficient assembly code for the time-critical sections of a program.

241



## Software Development Methodologies

---

- Mixing assembly code and high-level language code can be used to increase the range of applications in which high-level language can be used.
- Optimising ANSI C compilers translate ANSI standard C language files into highly efficient processor assembly language source files, which then are input to a processor assembler/linker.
- All processor compilers must be validated for their conformance to the ANSI C specification, using the industry-standard Plum-Hall test suite.

245

# Software Development Tools

