the Brainware company

THE DSPACE SDE

DSPACE

Vincenzo Pii

NOORDWIJK, 28 AUG 2012

SEVENTH FRAMEWORK PROGRAMME

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

OUTLINE

- ✓ Requirements
- ✓ SDE modules and architecture
- ✓ Usage and Examples
- ✓ Conclusions

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

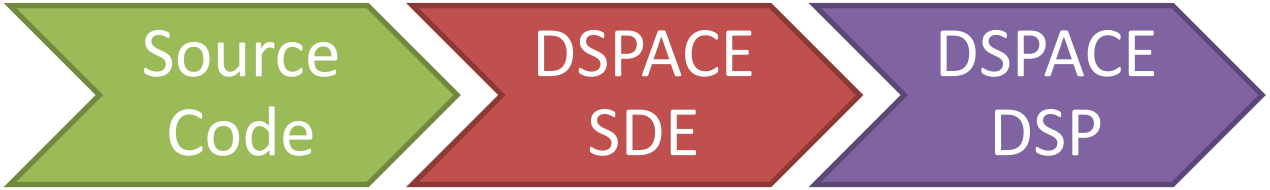
DSPACE SDE

MAIN REQUIREMENTS

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

WHY A DSPACE SDE?

- The DSPACE is a new VLIW Digital Signal Processor
  - New hardware architecture
  - New instructions Set
- Existing software products cannot be used to operate on this device
- A new set of dedicated tools must be provided as a necessary supplement to the hardware
  - Compiler
  - Assembler
  - Linker
  - Debugger
  - Simulator
  - Device drivers
- The DSPACE SDE (Software Development Environment) is the collection of software components that programmers can use to develop DSPACE software



Source Code

DSPACE SDE

DSPACE DSP

INTERNAL USE ONLY- REPRODUCTION FORBIDDEN

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

MAIN REQUIREMENTS

- To provide a consistent **Software Development Environment** (SDE) for the DSPACE is a task of critical importance
- A comprehensive, easy to use, reliable software environment with standard functionality is required
  - Programmers want to be able to write code for the new processor the way they are accustomed to, using well-known interfaces
- The use of existing software, where applicable, is strongly encouraged

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

AVAILABLE TOOLS

- The DSPACE processor architecture has been described using the LISA language
- From the LISA description, the following tools were automatically generated
  - Assembler
  - Linker
  - Other tools (e.g., debugger, simulator)

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

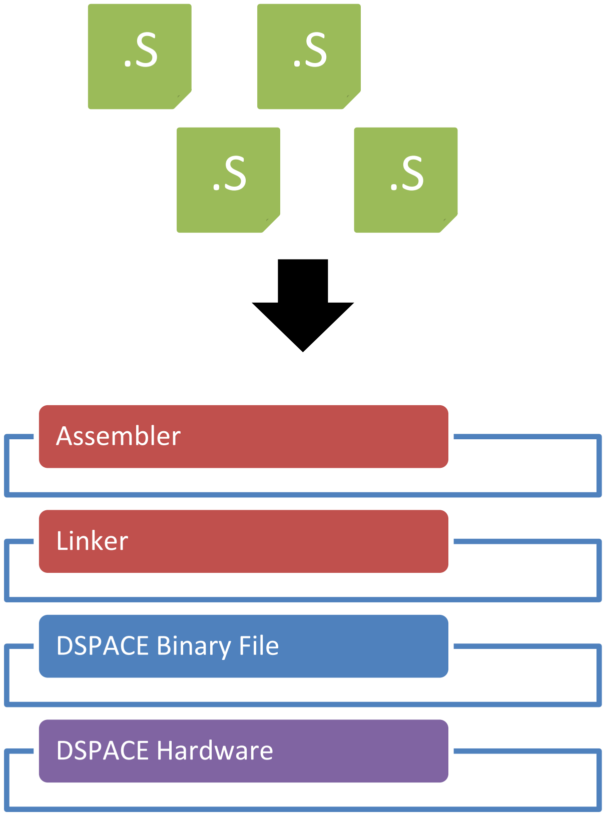
DSPACE SDE

MODULES AND ARCHITECTURE

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

A POSSIBLE WORKFLOW

- With the Assembler and the Linker a first development workflow is made possible
  - Source code is written in the DSPACE **assembly** language
  - Object code is generated by the **Assembler**
  - The **Linker** creates the final executable



INTERNAL USE ONLY- REPRODUCTION FORBIDDEN

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

BEYOND ASSEMBLY SUPPORT

- To manually write assembly code is generally a poor solution for developers
  - Error prone
  - Hard to maintain
  - Time-consuming
  - Difficult to implement (VLIW architecture)
    - ❑ Resource allocation
    - ❑ Code scheduling
    - ❑ Register spilling
    - ❑ Parallel instructions
- Common practice to optimize small critical code portions
- This is not enough to ensure the success of the new hardware
- Developers must be provided with a C source code compiler

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

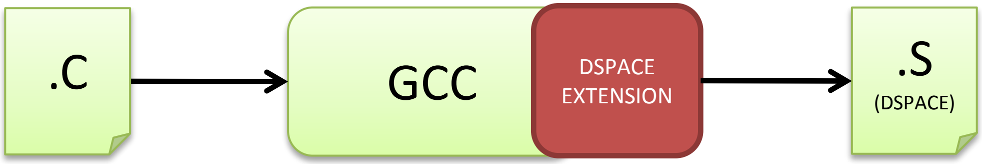
BUILDING A DSPACE C COMPILER

- New module to let programmers develop in C
- Requirements for a DSPACE C Compiler module
  - Assembler and linker already available
  - **Compile C source code into DSPACE Assembly**
  - This completes the tool-chain
- Implementation strategies
  - Create a new dedicated compiler from scratch
  - Build on top of existing software components (e.g. reuse C front-end capabilities)
- Survey of existing software tool-chains led to **gcc**-based implementation
  - Usability
  - Maturity
  - Reliability
  - Quality
  - Licensing
  - Source code availability

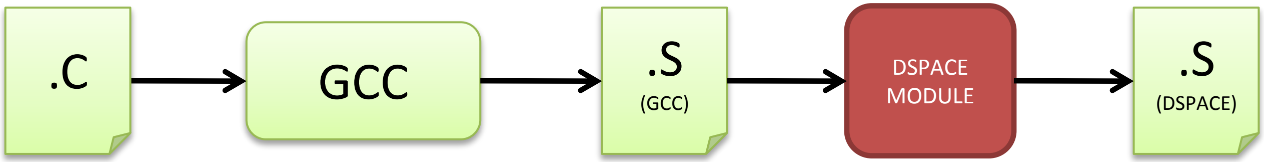
The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

GCC INTEGRATION SCHEMES

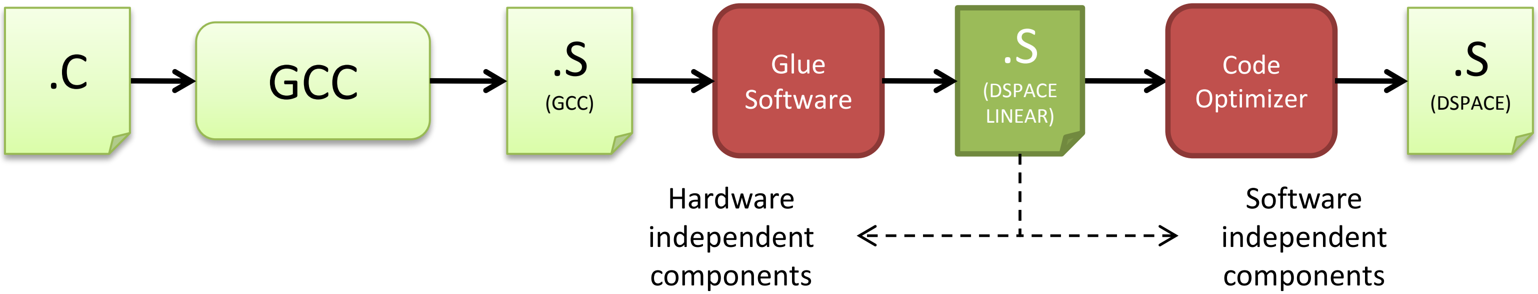
- Different integration schemes were envisaged in order to build the **DSPACE compiler** from the **GCC**
  - Extend the GCC compiler back-end by accessing its source code



  - Single component adapting GCC output to DSPACE

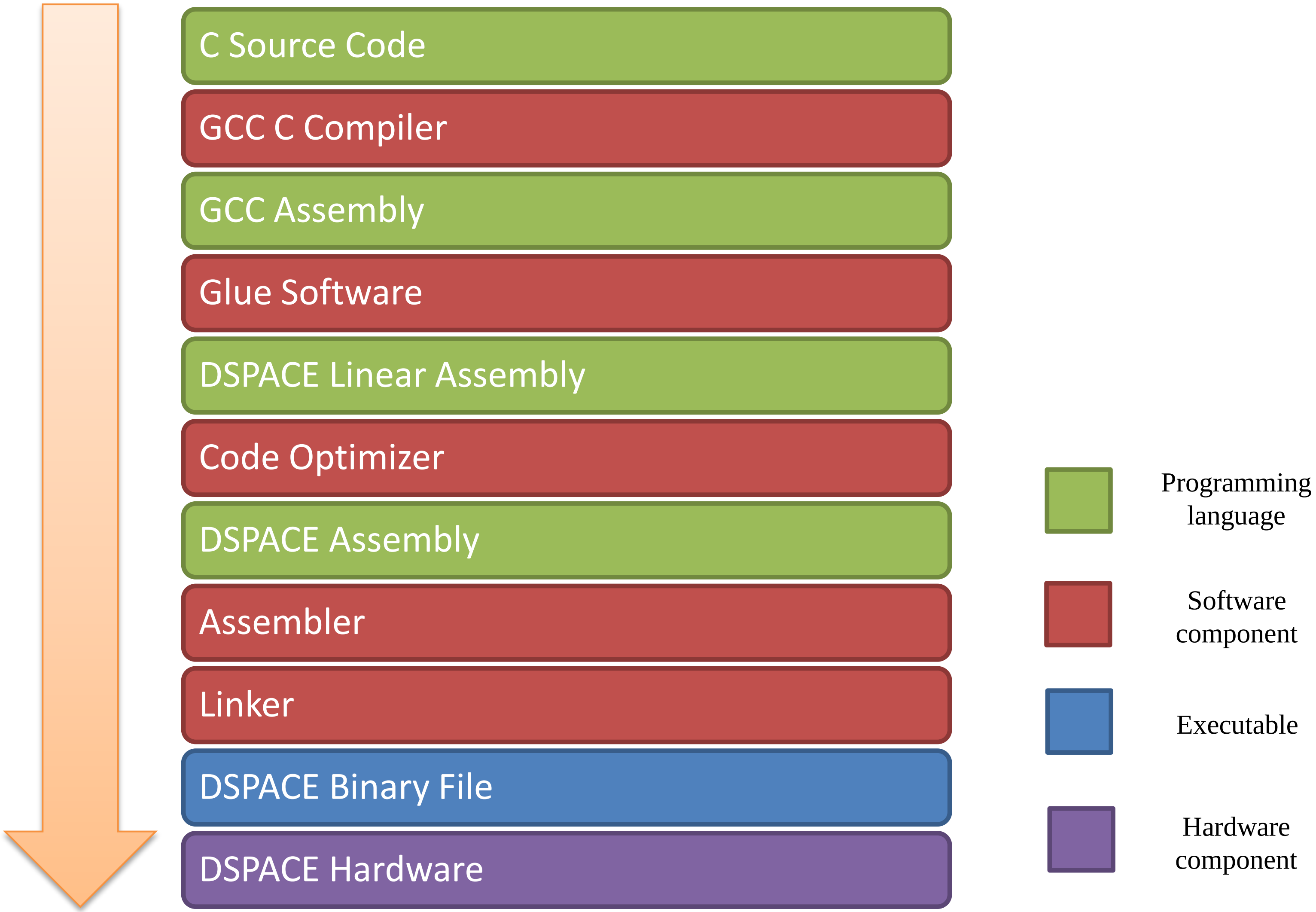


  - Two components creating an abstraction layer between software and hardware



The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

COMPLETE DSPACE TOOL-CHAIN



C Source Code

GCC C Compiler

GCC Assembly

Glue Software

DSPACE Linear Assembly

Code Optimizer

DSPACE Assembly

Assembler

Linker

DSPACE Binary File

DSPACE Hardware

🟩 Programming language

🟥 Software component

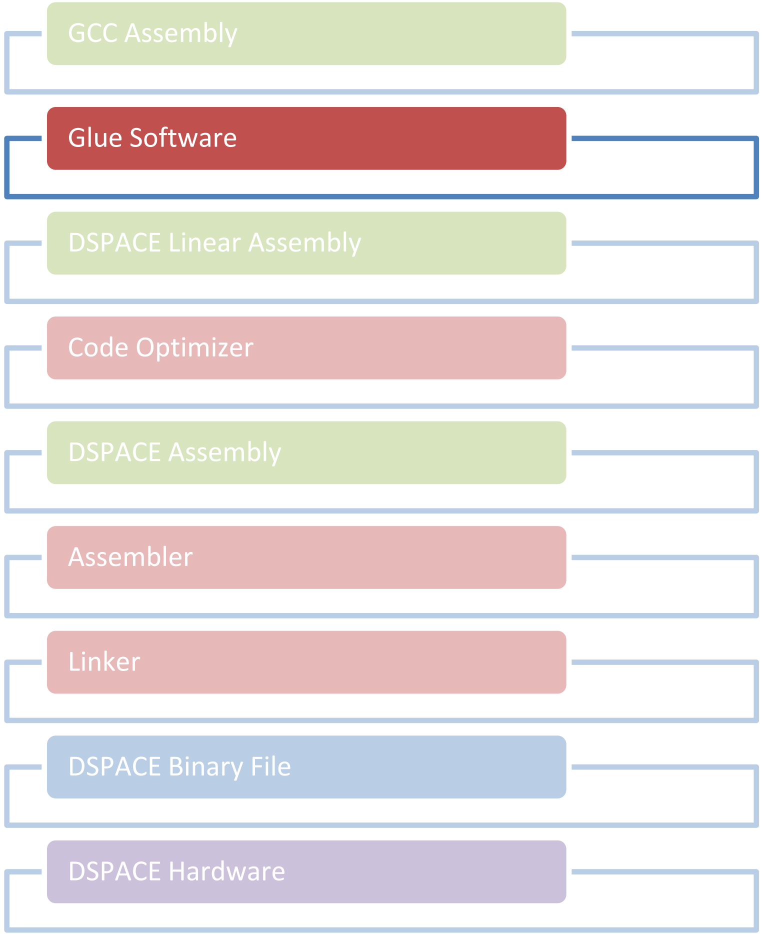
🟦 Executable

🟪 Hardware component

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

THE GLUE SOFTWARE

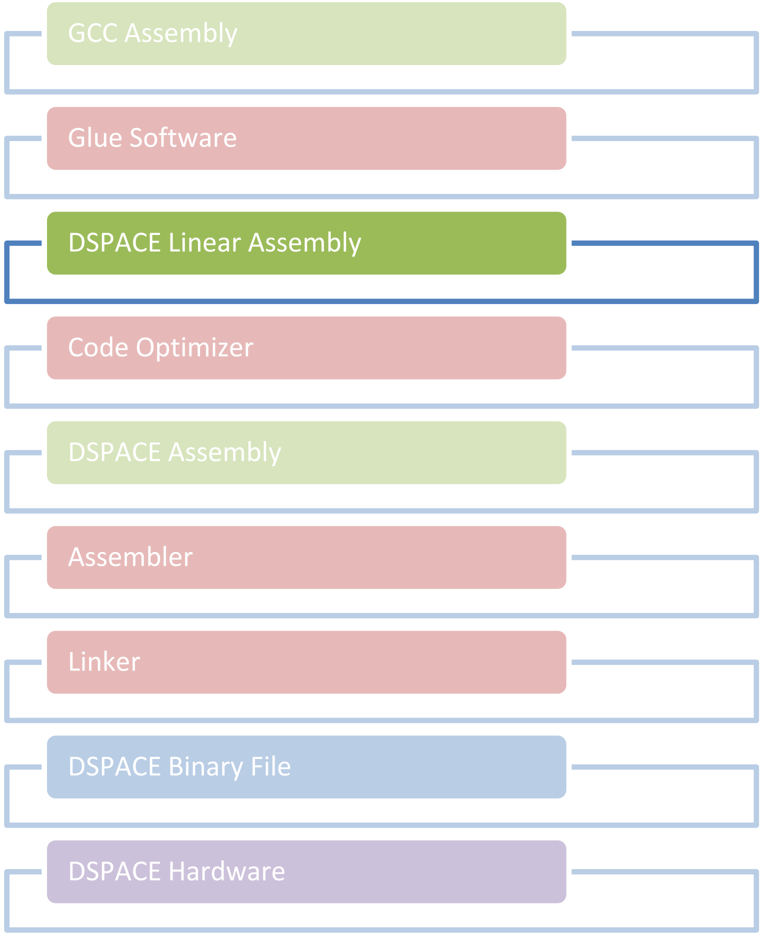
- INPUT
  - Assembly code generated by the **GCC**
- OPERATIONS
  - Source code parsing
  - Code transformations for hardware abstraction (e.g., SSA)
    - ❑ Enables for more code optimizations
    - ❑ Removes hardware constraints
    - ❑ (Hardware independent optimizations are kept from the GCC)
  - Instruction set translation
- OUTPUT
  - DSPACE **Linear** Assembly



The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

DSPACE LINEAR ASSEMBLY

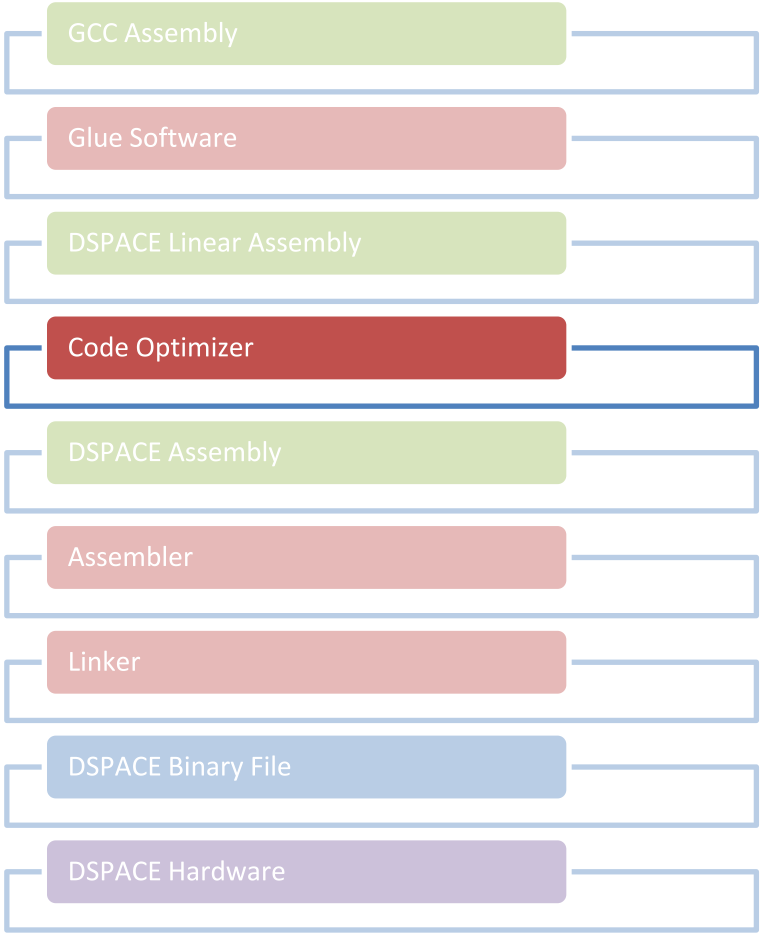
- Language specification
- Hardware independent form of DSPACE Assembly (w/DSPACE instruction set)
  - No functional units specification
  - No code scheduling
    - ❑ Order of statements corresponds to order of execution (no delay slots)
  - Virtual registers (infinite number of registers)
  - No parallel statements
- Abstract interface between Glue Software and Code Optimizer
  - Decouples hardware from software (and vice-versa)
  - Makes the development of the tool-chain easier
- Can be easily developed manually



The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

CODE OPTIMIZER

- INPUT
  - DSPACE Linear Assembly
- OPERATIONS
  - Code scheduling
  - Registers allocation
  - Functional units assignment
  - Introduction of VLIW parallelism
  - Code Optimizations
- OUTPUT
  - DSPACE Assembly



GCC Assembly

Glue Software

DSPACE Linear Assembly

Code Optimizer

DSPACE Assembly

Assembler

Linker

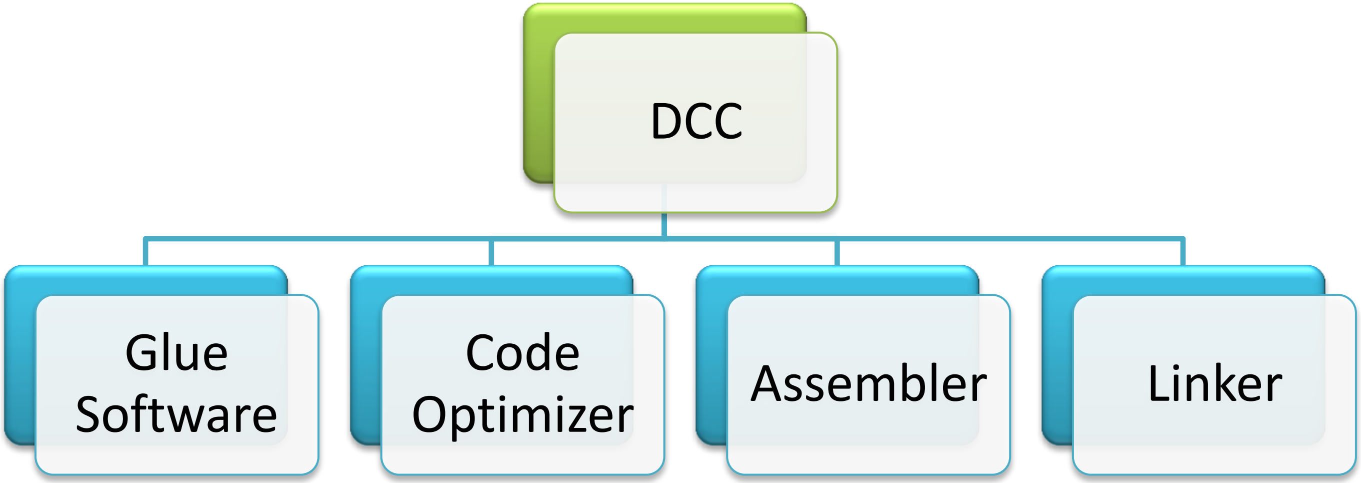
DSPACE Binary File

DSPACE Hardware

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

DCC (DSPACE C COMPILER) AS A WHOLE

- The DCC is the single DSPACE software component that calls the modules of the tool-chain
- It globally operates as a static binary translation compiler
- It has the same Command Line Interface as the **gcc** (plus DSPACE specific options)



The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

DSPACE SDE

USAGE AND EXAMPLES

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

DCC CLI USAGE

- Usage:

```
        dcc [options] file...
```

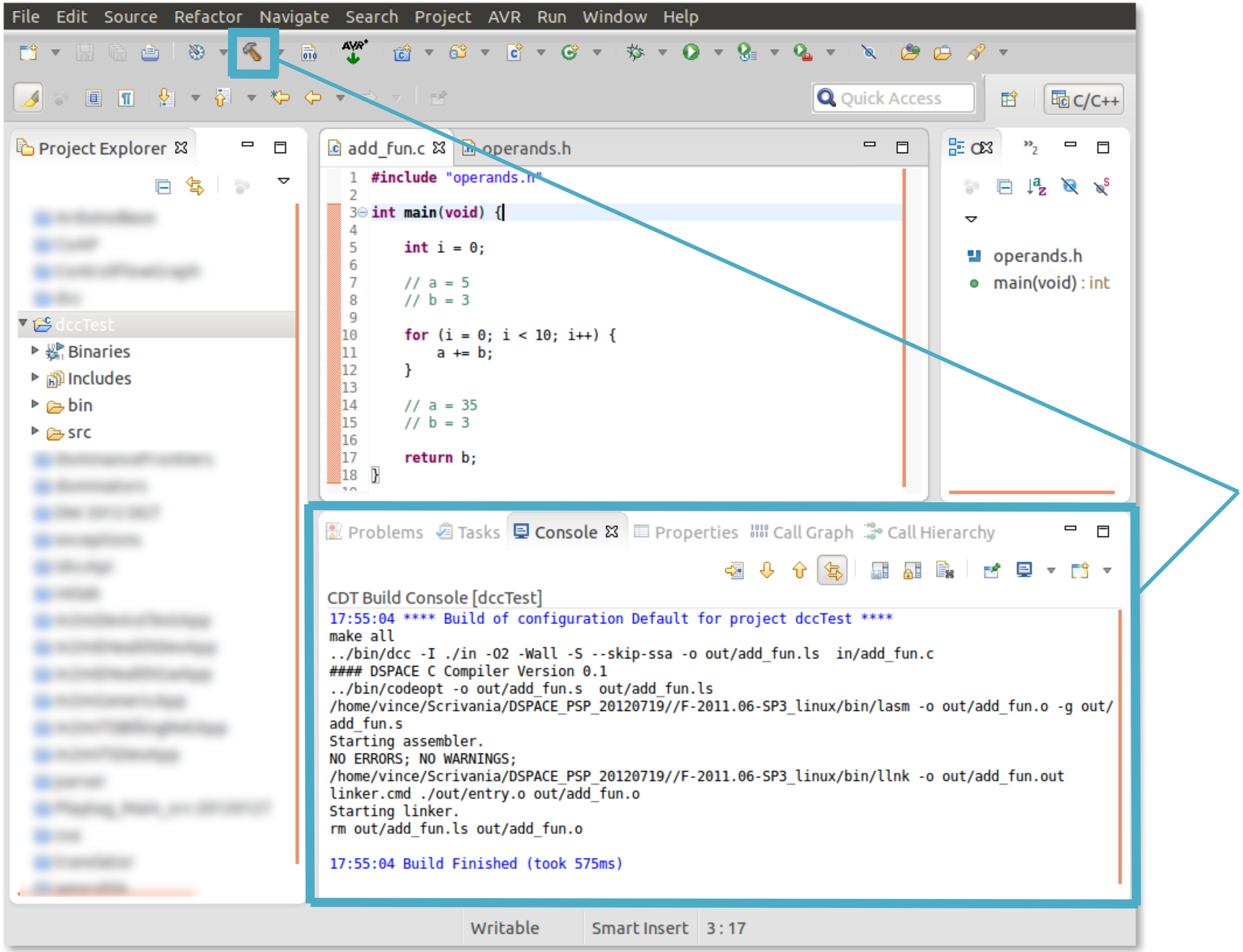
- Options:
  - Same as GCC, plus DSPACE specific

- Example:

```
        dcc -O2 -Wall main.c --keep-temps -I../include/
```

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

ECLIPSE INTEGRATION - BUILD



```
#include "operands.h"

int main(void) {

    int i = 0;

    // a = 5
    // b = 3

    for (i = 0; i < 10; i++) {
        a += b;
    }

    // a = 35
    // b = 3

    return b;
}
```

CDT Build Console [dccTest]

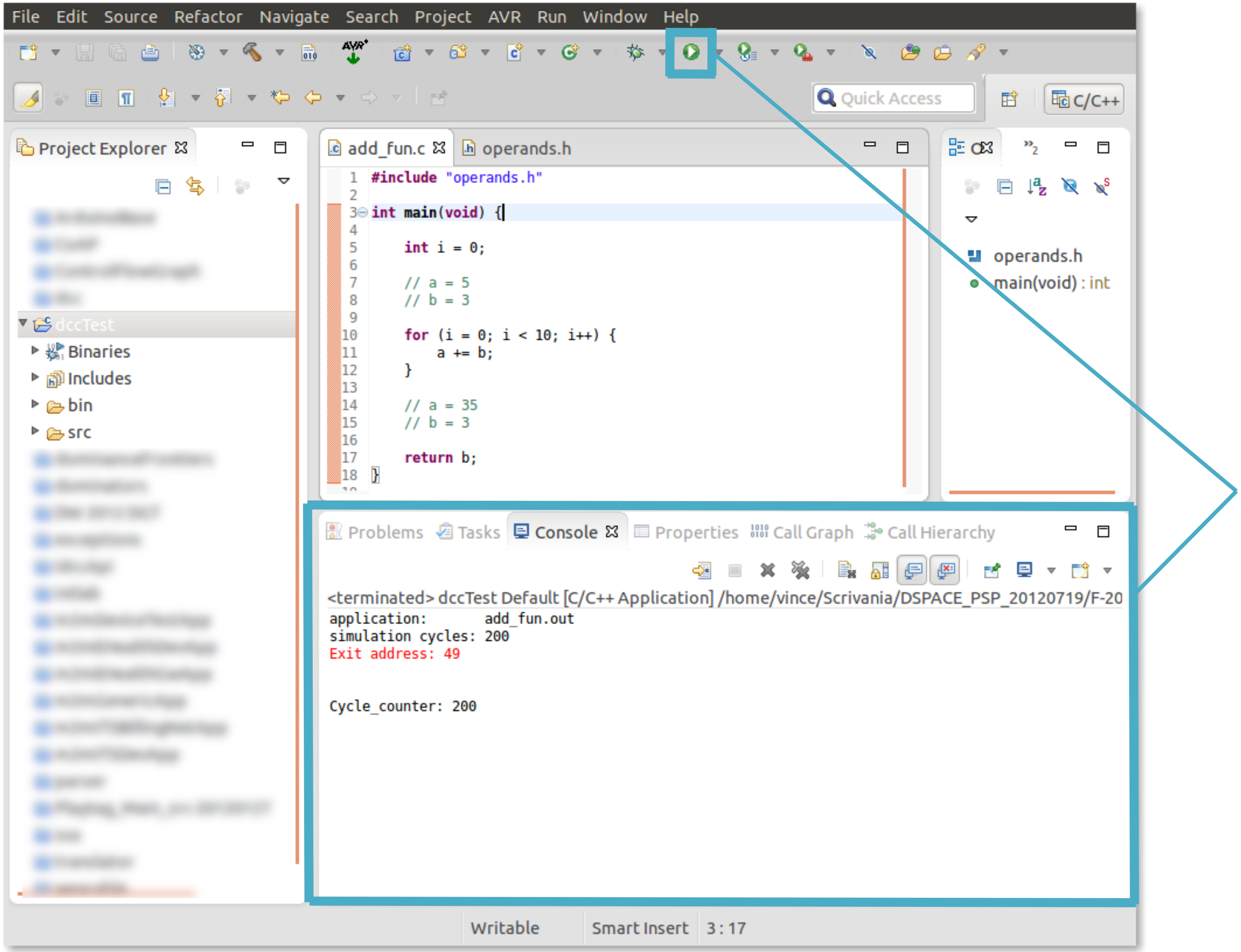
```
17:55:04 **** Build of configuration Default for project dccTest ****
make all
../bin/dcc -I ./in -O2 -Wall -S --skip-ssa -o out/add_fun.ls  in/add_fun.c
#### DSPACE C Compiler Version 0.1
../bin/codeopt -o out/add_fun.s  out/add_fun.ls
/home/vince/Scrivania/DSPACE_PSP_20120719//F-2011.06-SP3_linux/bin/lasm -o out/add_fun.o -g out/
add_fun.s
Starting assembler.
NO ERRORS; NO WARNINGS;
/home/vince/Scrivania/DSPACE_PSP_20120719//F-2011.06-SP3_linux/bin/llnk -o out/add_fun.out
linker.cmd ./out/entry.o out/add_fun.o
Starting linker.
rm out/add_fun.ls out/add_fun.o

17:55:04 Build Finished (took 575ms)
```

DSPACE
Tool-Chain
Eclipse
Integration

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

ECLIPSE INTEGRATION - RUN



```
#include "operands.h"

int main(void) {

    int i = 0;

    // a = 5
    // b = 3

    for (i = 0; i < 10; i++) {
        a += b;
    }

    // a = 35
    // b = 3

    return b;
}
```

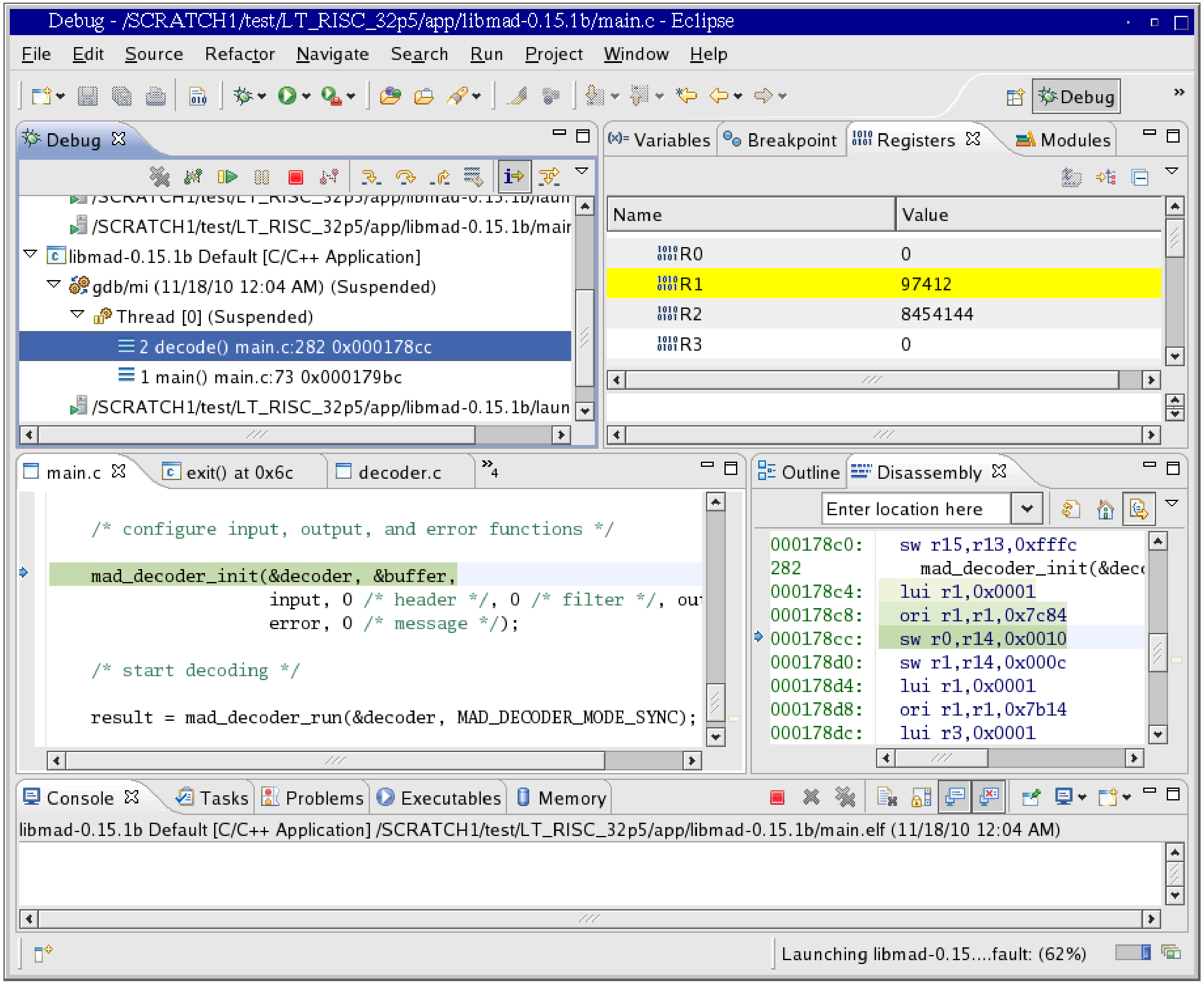
```
<terminated> dccTest Default [C/C++ Application] /home/vince/Scrivania/DSPACE_PSP_20120719/F-20
application:       add_fun.out
simulation cycles: 200
Exit address: 49


Cycle_counter: 200
```

DSPACE
Simulator
Eclipse
Integration

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

ECLIPSE INTEGRATION - DEBUGGER



Debug - /SCRATCH1/test/LT_RISC_32p5/app/libmad-0.15.1b/main.c - Eclipse

```
/SCRATCH1/test/LT_RISC_32p5/app/libmad-0.15.1b/laun
/SCRATCH1/test/LT_RISC_32p5/app/libmad-0.15.1b/mair
libmad-0.15.1b Default [C/C++ Application]
  gdb/mi (11/18/10 12:04 AM) (Suspended)
    Thread [0] (Suspended)
      2 decode() main.c:282 0x000178cc
      1 main() main.c:73 0x000179bc
/SCRATCH1/test/LT_RISC_32p5/app/libmad-0.15.1b/laun
```

Name	Value
R0	0
R1	97412
R2	8454144
R3	0

```
  /* configure input, output, and error functions */

  mad_decoder_init(&decoder, &buffer,
                   input, 0 /* header */, 0 /* filter */, ou
                   error, 0 /* message */);

  /* start decoding */

  result = mad_decoder_run(&decoder, MAD_DECODER_MODE_SYNC);
```

```
000178c0:   sw r15,r13,0xfffc
282           mad_decoder_init(&dec
000178c4:   lui r1,0x0001
000178c8:   ori r1,r1,0x7c84
000178cc:   sw r0,r14,0x0010
000178d0:   sw r1,r14,0x000c
000178d4:   lui r1,0x0001
000178d8:   ori r1,r1,0x7b14
000178dc:   lui r3,0x0001
```

```
libmad-0.15.1b Default [C/C++ Application] /SCRATCH1/test/LT_RISC_32p5/app/libmad-0.15.1b/main.elf (11/18/10 12:04 AM)
```

Launching libmad-0.15....fault: (62%)

The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

DSPACE SDE

CONCLUSIONS

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

CONCLUSIONS

- A new DSP processor has been designed within the DSPACE FP7 project
- For the success of the processor among software developers a complete tool-chain was strongly required
- We have developed the components for a DSPACE dedicated tool-chain that include hardware specific optimizations
- The DCC (DSPACE C Compiler) can be used from the command line with a well-known and standard interface or integrated into famous IDEs such as Eclipse

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

THANK YOU!

The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798

CONTACTS

Intecs S.p.A.
Rome, Pisa, Milan, Naples, Bacoli, Turin,
Cagliari, Genova, L'Aquila, Paris, Tolouse
www.intecs.it

PISA Premises
Via Umberto Forti, 5
Loc. Montacchiello
I–56121 Ospedaletto – Pisa
T. +39 050 96 57 411
F. +39 050 96 57 400

the Brainware company

DSPACE Members

Elena Cordiviola
Project Manager
elena.cordiviola@intecs.it

Vincenzo Pii
Software Engineer
vincenzo.pii@intecs.it

Antonio Spada
Software Engineer
antonio.spada@intecs.it

INTERNAL USE ONLY- REPRODUCTION FORBIDDEN

*The research leading to these results has received funding from
the European Community's Seventh Framework Programme ([FP7/2007-2013]) under Grant Agreement n°262798*