

Release Notes

ACSE 1.2.3

Released on 2021-12-15

- Fixed a bug in ACSE where multiple assignments of the same label did not generate an error.
- Fixed a bug in the assembler where multiple definitions of the same label did not generate an error.
- In the assembler, return a non-zero code from `main()` in case of error.
- Minor corrections to the comments in ACSE.
- Minor cleanup of the code related to labels.

ACSE 1.2.2

Released on 2021-12-13

- Made shifts with amount greater than 31 and less than zero deterministic in both ACSE and MACE.
- Minor fixes to the makefiles.

ACSE 1.2.1

Released on 2021-09-13

New Features

- Now almost every header file in acse has a comment describing its purpose.
- Strings passed to `yyerror()` are now displayed rather than discarded.
- Improved formatting of the acse logs and the assembly printout.
- Changed file extension of the acse logs to `.log` from `.cfg` and `.out`.
- The name of the acse log files is now determined from the name of the input source code file to be compiled.
- Show a warning when the source code contains a shift amount that is too large.

Bug Fixes

- Fixed documentation of the `gen_bhi_instruction()`, `gen_bls_instruction()`, `gen_bcs_instruction()`, and `gen_bgt_instruction()` functions.
- Acse no longer crashes when the source code contains a division by zero in a constant expression.
- Fixed semantics of the `.SPACE` assembler directive to take a number of words rather than a number of bytes.

- Removed `SEGMENTED` mode from MACE, as it did not properly load the data segment of executables.
- Other minor fixes to documentation and comments.

ACSE 1.2.0

Released on 2020-09-14

New Features

Changes to acse:

- The logical `NOT` operator (!) now supports expression arguments.
- Added `clang-format` and `editorconfig` files for automatic configuration of indentation and linting in several editors.
- The debug dump files and the compiled assembly now contains comments that map the instructions to the line in the source code that produced them.
- Labels corresponding to variables are now named like the variable in dump files and the compiled assembly.
- `collections`: added new `addBefore()` and `addAfter()` functions for adding an element before or after a given element in a list in $O(1)$ time.
- `collections`: added new macros (`INTDATA`, `LINTDATA` and `SET_INTDATA`) for simplifying the use of the list data pointer as an integer.
- `axe_engine`: added new functions `newNamedLabel()` and `assignNewNamedLabel()` to create labels with a personalized name.
- It is now possible to use immediates of any size (not restricted to 16 bits anymore) when generating instructions with immediate operands (`ADDI`, `SUBI`, `MULI`...)
- Add a new debug output file named `frontend.out` containing the state of the IR just after parsing, before any transformation takes place.
- `acse` now prints the version number on startup.
- Added support for computing def/use of the PSW (flags) register, as a debugging aid (the PSW register now appears in `dataflow.cfg`)
- Debug files are now flushed more frequently to ensure complete logs in case of crashes.
- Documentation improvements (fixed typos and grammar mistakes, clarified descriptions of some functions)
- Other API additions for internal use.

Changes to mace:

- Implemented `JSR` and `RET` instructions.
- Add a new instruction (`XPSW`) which exchanges the value of the PSW register with the value of a general-purpose register.
- When compiling `mace` with `DEBUG` enabled, `fflush stderr` at every instruction executed to ensure complete logs in case of crashes.
- Added a verification tool to use for regression testing. It can be executed by running `make verify` inside the `mace` directory.
- All comments written in Italian were replaced by comments written in English.

- Minor improvements to the debug features.

Changes to asm:

- Comments are now allowed anywhere in assembly language sources (previously they were allowed only after an instruction)
- Added support for hexadecimal immediates in the form `0x[0-9A-Za-z]+`
- Added support for inserting multiple data words within the same `.WORD` directive (syntax: `.WORD <word_1> <word_2> ... <word_n>`)

Changes to tests:

- Renamed syntax error tests to more descriptive names.

Other changes:

- Code formatting improvements across the entire project.
- Improvements to the makefiles:
 - All makefiles can now use the standard configuration variables `CFLAGS`, `YFLAGS` (bison), `LFLAGS` (flex) and `LDFLAGS`.
 - The test makefile now rebuilds the tests when `Acse` has changed.
 - Changed the test makefile to be whitespace-friendly.
 - Other improvements

Bug Fixes

Changes to acse:

- `handle_bin_numeric_op` now supports all binary operations used in `Acse.y`.
- Fixed `gen_move_immediate` and `gen_load_immediate` to properly handle positive numbers larger than 32767 with bit 15 set.
- Fix a bug where an initialized variable was not properly loaded if it was used and defined exactly once by the same instruction in the first basic block.
- Fix a bug where invalid tokens after a valid statement caused parsing to stop mid-program without an error.
- Fix a bug where line numbers were incorrectly counted on lines with C++ style comments.
- Fix a bug in register allocation materialization where spill store instructions were placed one instruction too early in basic blocks without an explicit terminator instruction.
- Fixed a bug where the register allocator would unexpectedly run out of spill registers because the liveness of registers allocated to RD operands was not tracked correctly.
- Handle `RET` instructions like `HALT` instructions when performing analysis passes.
- Properly generate a `NEG` instruction instead of a `SUB` instruction in `gen_neg_instruction`.
- Properly return `SY_LOCATION_UNSPECIFIED` from `getLocation()` even when `errorcode` is `NULL`.
- Removed unused tokens deriving from incomplete implementations of old exams (`MOD_OP`, `FOR`, `COLON`).
- Fix several minor uninitialized memory accesses in exceptional situations.
- Other improvements

Changes to mace:

- Fix a bug where the carry flag is set incorrectly after SHR/SHRI instructions.
- Fix a bug where the carry flag is set incorrectly after SUB/SUBI instructions.
- Fix a bug where the overflow flag is set incorrectly after MUL/MULI instructions with specific operands (i.e. $-2147483648 * -1$ did not set the overflow flag even though it should have)
- Do not set the carry flag after MUL instructions.
- Do not crash when performing the specific division $\text{INT_MIN} / -1$.
- Fix a bug where the carry flag is set incorrectly after SHL and SHR when the shift amount is zero (it should have been reliably set to zero, instead it was undefined behavior).
- Fix behavior of ROTR when shift amount is zero.
- Ensure that SUB and NEG instructions set flags in the same way.
- Other improvements (code cleanup)

Changes to asm:

- Fixed a bug where `asm` would crash if `findLabel` is called with incorrect arguments.

Other changes:

- Fixed several bugs involving expressions with negative numbers, which were caused by a mismatch between signedness conventions of `acse` and `mace`.
- Makefiles: updated the list of tests (removed a duplicate, added a missing test)
- Fixed build on MinGW (Windows)
- Fixed various compilation warnings

ACSE 1.1.5

Released on 2019-12-31

- Fix wrong operand order in `gen_shl_instruction` and `gen_shr_instruction`. Add a test to verify correct behavior.

ACSE 1.1.4

Released on 2019-11-21

- Replaced deprecated bash option `-a` with the equivalent `-e`.

ACSE 1.1.3

Released on 2018-06-13

- Fix `gen_move_immediate` and `gen_load_immediate` to properly handle negative numbers.

ACSE 1.1.2

Released on 2017-12-14

- Fixed various compilation warnings

Changes to asm:

- Fixed parsing of EORL/EORLI/EORB/EORBI instructions (which were incorrectly spelled as XOR*)

Changes to mace:

- Sign-extend immediate operands in a platform-independent way.
- Code cleanup and logic fixes to ADD and SUB instructions.

ACSE 1.1.1

Released on 2016-02-02

- Removed references to `malloc.h` to fix build on recent systems.

Changes to acse:

- Modified the `gen_load_immediate()` function to correctly handle arbitrarily-sized integers.
- Introduced the new `gen_move_immediate()` function.
- Fixed implementation of `gen_shl_instruction()` and `gen_shr_instruction()`.

Changes to asm:

- Fixed handling of absolute addresses.

ACSE 1.1.0

Released on 2008-12-27

Changes to acse:

- Renamed the following functions to increase clarity of the code:
 - `reserveLabel()` to `newLabel()`
 - `reserveLabelID()` to `newLabelID()`
 - `fixLabel()` to `assignLabel()`
 - `fixLabelID()` to `assignLabelID()`
 - `perform_bin_numeric_op()` to `handle_bin_numeric_op()`
 - `perform_binary_comparison()` to `handle_binary_comparison()`
 - `load_immediate()` to `gen_load_immediate()`
- Correctly handle invalid tokens.
- Minor fixes to the Control Flow Graph (CFG) generation algorithm.

- Minor fixes to the register allocator.
- Translate all Italian comments to English.
- Corrections to the documentation and comments.

Changes to asm:

- Fixed parsing of comments in assembly listings.

Changes to mace:

- Fix PSW updating behavior of unary instructions to conform to the specification.

Other changes:

- Introduced unified Makefile for tests.
- Minor code cleanup across the project.

ACSE 1.0.0

Released on 2008-01-07

First public release.