nolimips

The MIPS Architecure Emulator
10 June 2014, NOLIMIPS Version 0.9a

Benoît Perrot

This manual is for Nolimips (version 0.9a, 10 June 2014), the tiny MIPS simulator.

Table of Contents

NOLIMIPS is a basic MIPS architecture simulator. It consists of an assembler which reads MIPS assembly code, and of a virtual machine which executes the instructions processed by the assembler. For the moment NOLIMIPSdoes not produce nor execute binary code, but ELF writer and loader are planned.

The main purpose of NOLIMIPSis to help students of compilers courses to evaluate the output from their "high-level language into MIPS assembler code" compilers, before and after register allocation.

Its features are:

- minimal support of MIPS instruction set
- built-in system calls (string printing and reading, etc.)
- unlimited registers ($x0, $x1, etc.).

It was written by Benoît Perrot as an LRDE member, so that EPITA students could exercise their compiler projects after the instruction selection but before the register allocation. It is implemented in C++ and Python. Be aware that NOLIMIPS does not need a Python interpreter to run, it is only required to maintain NOLIMIPS.

Information about NOLIMIPS can be found on the NOLIMIPS Home Page[1], and feedback can be sent to lrde's Projects Address[2]. LRDE stands for Laboratoire de Recherche et Dévelopment de l'EPITA[3], i.e., the Research and Development Lab of EPITA, the Ecole Pour l'Informatique et les Techniques Avancées[4].

Andrew Appel's home page[5] includes links to material related to compilers theory, and some information about the Modern Compiler Implementation[6] book series.

More information on Python can be found on Python Home Page[7].

[1] NOLIMIPS Home Page, `https://www.lrde.epita.fr/wiki/Nolimips`.

[2] lrde's Projects Address, `projects@lrde.epita.fr`.

[3] Laboratoire de Recherche et Dévelopment de l'EPITA, `http://www.lrde.epita.fr`.

[4] Ecole Pour l'Informatique et les Techniques Avancées, `http://www.epita.fr`.

[5] Andrew Appel's home page, `http://www.cs.princeton.edu/~appel/`.

[6] Modern Compiler Implementation, `http://www.cs.princeton.edu/~appel/modern/`.

[7] Python Home Page, `http://www.python.org`.

1 Invoking Nolimips

To invoke `nolimips` run:

```
nolimips options file
```

where '`file.s`' is a simple text file, and *options* is any combination of the following options:

'`-h`'
'`--help`' Display a help message and exit successfully.

'`-V`'
'`--version`'
Display the version number and exit successfully.

'`--usage`' Give a short usage message

'`--tasks-selection`'
Each task of NOLIMIPS (parsing, execution, etc.) comes with a set of prerequisites: for example the pretty printing of an input program implies the parsing of its source file. This option asks NOLIMIPS to display the tasks that must be run before to run the ones explicitly specified on command line.

'`--parse`' Parse a file.

'`--trace-scan`'
Trace the scanning.

'`--trace-parse`'
Trace the parse.

'`-N`'
'`--nop-after-branch`'
To avoid a bubble in their pipeline, MIPS processors execute the instruction immediately following a branch; this instruction is said to be in the delay slot (FIXME: see further). This option fills the delay slot of branch instructions with a NOP, disabling the delay slot, simplifying the task of assembly programmers how do not care about writing optimized code.

'`-u`'
'`--unlimited-regs`'
During last stages of a compiler, the intermediate representation of a source file (which mainly consists in a generic, architecture independent assembly code) is progressively translated into an architecture dependent assembly code. Low level (but still intermediate) representations are designed to be as near as possible of target assembly code, but generaly consider an extended target machine with an unlimited amount of registers. This option gives NOLIMIPS the ability to handle an arbitrary number of registers, allowing compiler developpers to test their low level output before implementing register allocation. These new registers may be used as any other MIPS registers through the symbols `$x0`, `$x1`, and so on. They have a general purpose and are not considered as caller save nor callee save registers.

'`--prg-display`'
Display the read program.

'`--prg-solve`'
: Resolve jump offsets and check bounds of immediates.

'`--callee-save=`*`num`*'
'`--caller-save=`*`num`*'
'`--argument-registers=`*`num`*'
: Respectively set the maximum number of callee-save, caller-save and argument registers to *num*, a positive number.

'`--check-callee-save`'
: Warn if a callee save register is not preserved across a call.

'`--profile`'
: Enable program profiling

'`-l` *`library`*'
'`--system-library=`*`library`*'
: Specify the builtin system library to use. Accepted *library* values are '`spim`' (selected by default and implementing SPIM's[1] behavior), '`nolimips`' (NOLIMIPS' own library), and '`none`' (no builtin library).

'`-e`'
'`--execute`'
: Execute the program on virtual machine.

'`-E`'
'`--trace-exec`'
: Trace the execution.

'`-i`'
'`--shell`' Launch interactive shell.

[1] SPIM's, `http://spimsimulator.sourceforge.net/`.

2 Internal of Nolimips

2.1 Steps of Nolimips

NOLIMIPS works in three steps:

- it scans and parses the file (lexical and syntaxical analysis), producing an abstract representation of a program;
- it resolves the program, checking the existence of labels used and computing branch offsets (assembly);
- it loads the resolved program in the virtual machine, search for an entry point labeled `main` and start execution (execution).

3 The Nolimips Language

NOLIMIPS supports a minimal MIPS instruction set and unlimited registers.

3.1 The Nolimips supported instruction set

3.1.1 Arithmetic instructions

3.1.1.1 ADD

Add src1 and src2 and store the result in dest (32-bit integers). If an overflow occurs, then trap.

3.1.1.2 ADDU

Add src1 and src2 and store the result in dest (32-bit integers).

3.1.1.3 ADDI

Add a constant imm and src and store the result in dest (32-bit integers). If overflow occurs, then trap.

3.1.1.4 ADDIU

Add a constant imm and src and store the result in dest (32-bit integer).

3.1.1.5 SUB

Subtract src2 from src1 and store the result in dest (32-bit integers). If an overflow occurs (FIXME), then trap.

3.1.1.6 SUBU

Subtract src2 from src1 and store the result in dest (32-bit integers).

3.1.1.7 NEG

Negate (logical 2-complement) src and store the result in dest. If an overflow occurs (FIXME), then trap.

3.1.1.8 NEGU

Negate src (logical 2-complement) and store the result in dest.

3.1.1.9 ABS

Compute the absolute value of src (32-bit integer) and write it to dest.

3.1.1.10 MUL

Multiply two words src1 and src2 and write the result to dest.

3.1.1.11 DIV

Divide src1 by src2 (32-bit signed integers), write the quotient in LO and the remainder in HI (32-bit integer).

3.1.1.12 DIVU

Divide src1 by src2 (32-bit unsigned integers), write the quotient in LO and the remainder in HI (32-bit integer).

3.1.1.13 REM

Compute the remainder from dividing src1 by src2 (32-bit signed integers) and write it to dest.

3.1.1.14 REMU

Compute the remainder from dividing src1 by src2 (32-bit unsigned integers) and write it to dest.

3.1.2 Bitwise instructions

3.1.2.1 SLL

Left-shift (logical) the word src by the fixed number imm of bits and store the result in dest.

3.1.2.2 SLLV

Left-shift (logical) the word src1 by the variable number src2 of bits and store the result in dest.

3.1.2.3 SRA

Right-shift (arithmetic) the word src by the fixed number imm of bits and store the result in dest.

3.1.2.4 SRAV

Right-shift (arithmetic) the word src1 by the variable number src2 of bits and store the result in dest.

3.1.2.5 SRL

Right-shift (logical) the word src1 by the variable number src2 of bits and store the result in dest.

3.1.2.6 SRLV

Right-shift (logical) the word src1 by the variable number src2 of bits and store the result in dest.

3.1.2.7 ROL

Left-rotate the word src by a number of bits (imm or src2) and store the result in dest.

3.1.2.8 ROR

Right-rotate the word src by a number (imm or src2) of bits and store the result in dest.

3.1.2.9 AND

Compute the bitwise logical AND between src1 and src2 and store the result to dest.

3.1.2.10 ANDI

Compute the bitwise logical AND between src and a constant imm and store the result to dest.

3.1.2.11 OR

Compute the bitwise logical OR between src1 and src2 and store the result to dest.

3.1.2.12 ORI

Compute the bitwise logical OR between src and a constant imm and store the result to dest.

3.1.2.13 XOR

Compute the bitwise logical XOR between src1 and src2 and store the result to dest.

3.1.2.14 XORI

Compute the bitwise logical XOR between src and a constant imm and store the result to dest.

3.1.2.15 NOR

Compute the bitwise logical NOR between src1 and src2 and store the result to dest.

3.1.2.16 NOT

Negate (logical 1-complement) src and store the result in dest.

3.1.3 Comparison instructions

3.1.3.1 SEQ

Set dest to 1 if src1 equals src2, else clear it.

3.1.3.2 SNE

Set dest to 1 if src1 does not equal src2, else clear it.

3.1.3.3 SGE

Set dest to 1 if src1 is greater or equal to src2 (signed comparison), else clear it.

3.1.3.4 SGEU

Set dest to 1 if src1 is greater or equal to src2 (unsigned comparison), else clear it.

3.1.3.5 SGT

Set dest to 1 if src1 is greater than src2 (signed comparison), else clear it.

3.1.3.6 SGTU

Set dest to 1 if src1 is greater than src2 (unsigned comparison), else clear it.

3.1.3.7 SLE

Set dest to 1 if src1 is lower or equal to src2 (signed comparison), else clear it.

3.1.3.8 SLEU

Set dest to 1 if src1 is lower or equal to src2 (unsigned comparison), else clear it.

3.1.3.9 SLT

Set dest to 1 if src1 is lower than src2 (signed comparison), else clear it.

3.1.3.10 SLTU

Set dest to 1 if src1 is lower than src2 (unsigned comparison), else clear it.

3.1.3.11 SLTI

Set dest to 1 if src1 is lower than a constant imm (signed comparison), else clear it.

3.1.3.12 SLTIU

Set dest to 1 if src1 is lower than a constant imm (unsigned comparison), else clear it.

3.1.4 Branch instructions

3.1.4.1 BEQ

Branch to label if src1 equals src2.

3.1.4.2 BEQZ

Branch to label if src equals zero.

3.1.4.3 BNE

Branch to label if src1 does not equal src2.

3.1.4.4 BNEZ

Branch to label if src does not equal zero.

3.1.4.5 BGE

Branch to label if src1 is greater or equal to src2 (signed comparison).

3.1.4.6 BGEU

Branch to label if src1 is greater or equal to src2 (unsigned comparison).

3.1.4.7 BGEZ

Branch to label if src is greater or equal to zero (signed comparison).

3.1.4.8 BGEZAL

Call label if src is greater or equal to zero (signed comparison)

3.1.4.9 BGT

Branch to label if src1 is greater than src2 (signed comparison).

3.1.4.10 BGTU

Branch to label if src1 is greater than src2 (unsigned comparison).

3.1.4.11 BGTZ

Branch to label if src is greater than zero (signed comparison).

3.1.4.12 BLE

Branch to label if src1 is lower or equal to src2 (signed comparison).

3.1.4.13 BLEU

Branch to label if src1 is lower or equal to src2 (unsigned comparison).

3.1.4.14 BLEZ

Branch to label if src is lower or equal to zero (signed comparison).

3.1.4.15 BLT

Branch to label if src1 is lower than src2 (signed comparison).

3.1.4.16 BLTU

Branch to label if src1 is lower than src2 (unsigned comparison).

3.1.4.17 BLTZ

Branch to label if src1 is lower than zero (signed comparison).

3.1.4.18 BLTZAL

Call label if src1 is lower than zero (signed comparison).

3.1.4.19 J

Jump to label unconditionaly.

3.1.4.20 JAL

Call label unconditionaly.

3.1.4.21 JR

Jump to address contained in dest unconditionaly.

3.1.4.22 JALR

Call address contained in dest unconditionaly.

3.1.5 Load instructions

3.1.5.1 LB

Load the 8-bit quantity at address (offset + base) into dest as a signed value.

3.1.5.2 LBU

Load the 8-bit quantity at address (offset + base) into dest as an unsigned value.

3.1.5.3 LW

Load the 32-bit quantity at address (offset + base) into dest as a signed value.

3.1.5.4 LUI

Move the constant imm into the upper half word of dest.

3.1.5.5 LI

Move the constant imm into dest.

3.1.5.6 LA

Move the computed address into dest.

3.1.6 Store instructions

3.1.6.1 SB

Store the low byte from src at address (offset + base).

3.1.6.2 SW

Store the low word from src at address (offset + base).

3.1.7 Movement instructions

3.1.7.1 MOVE

Move the contents of src to dest.

3.1.7.2 MFHI

Move the contents of HI to dest.

3.1.7.3 MFLO

Move the contents of LO to dest.

3.1.7.4 MTHI

Move the contents of dest to HI.

3.1.7.5 MTLO

Move the contents of dest to LO.

3.1.7.6 MFC0

Move the contents of control coprocessor src register to CPU dest register.

3.1.7.7 MTC0

Move the contents of CPU src register to control coprocessor dest register.

3.1.8 Syscall instructions

3.1.8.1 SYSCALL

Raise a system call exception.

3.1.9 Nop instructions

3.1.9.1 NOP

To do nothing.

3.2 Nolimips features

Appendix A Copying This Manual

A.1 GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

   If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

   If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

   It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

   You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

   A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.1.1 ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index