* DRAFT *

IPS Assembly Language

Programmer's Reference

Issue 2.0.3
June 18, 2011

This document is subject to modification. Please contact the maintainer of the document for the latest version.

AMSAT-BDA

The first version of this document was written
by Robin A. Gape, G8DQX

This document is maintained by Paul C. L. Willmott, VP9MU.

Reports of errors should be sent to vp9mu@amsat-bda.org

DOCUMENT HISTORY LOG

Status (Baseline/ Revision/ Canceled)	**Document Revision**	**Effective Date**	**Description**
Baseline	1.0.0	September, 1985	Initial version by Robin Gape (RG).
Revision	2.0.0	October 24, 2002	Converted from WordStar format to MS Word, and reformatted by Paul Willmott (PW)
Revision	2.0.1	October 26, 2002	Document name changed from "IPS Assemblers" to "IPS Assembly Langauge Programmer's Reference" (PW). Minor edits, addition of HIER option.
Revision	2.0.2	November 19, 2002	RTX 2000/2010 and ARM added as available assemblers, 1802 and Am1601 Assemblers added (PW).
Revision	2.0.3	June 18, 2011	Minor reformatting (PW).

CONTENTS

IPS ASSEMBLY LANGUAGE

The following introduction to assembler programming in IPS explains the general principles involved. Briefly: DON'T. If you must, this introduction must be supplemented by the processor specific documentation for the processor that you will be using, and the machine specific documentation for the target that you will attempt to program on.

The principal reason for the development of the IPS assemblers is to allow the compilation of IPS for a particular target machine. (Normally this is a cross or X-Compilation)

AVAILABLE ASSEMBLERS

IPS assemblers are available for the following processors:

- RCA/COSMAC 1802
- 6502
- 6800
- 6809
- 8080/Z80
- RTX 2000/2010
- ARM
- AM1601

USE OF ASSEMBLERS IN IPS

IPS has sufficient resources to program most problems without having to be intimately familiar with either the structure or the machine language of any particular target machine. The 3 major exceptions are:

i) to interface to particular hardware. The I/O structure of a particular IPS implementation may not be capable of the desired interface without modification, particularly if interrupts are involved.

ii) to program a time critical problem which cannot be allowed any overhead. The IPS emulator has an overhead of typically 25 microseconds per executed word. Further, IPS treats all numbers as 16 bit. This results in an overhead if only bytes need to be manipulated. These two effects result in IPS programs typically running about 2 to 3 times slower than optimum machine code.

iii) to program special mathematical operations, such as rotational operations which would be inefficient using the standard IPS operators.

All these classes of problems are addressed by the ability to define new IPS words in terms of machine instructions rather than other IPS words. These newly defined words can then be used in the same fashion as other IPS words, but giving the programmer machine level access.

The IPS assembler provides the ability to define new words for a particular processor/machine. For the programmer to use an assembler successfully they must be familiar with the particular assembler that is used with their target machine.

ASSEMBLER APPLICATIONS

The IPS assemblers are intended mainly for short routines, that interface between high level IPS, the processor and application hardware. The IPS assemblers are not intended to support extended programs.

Assembly programs are hard to debug, often use more memory than the high level equivalent, and are machine specific. Assembler routines cannot easily be transferred from one processor to another.

When programming time critical routines in assembler, it is usually only worthwhile assembly coding the inner loop.

Assemblers are for consenting adults (!) only.

ASSEMBLER PHILOSOPHY

The assemblers have been kept as simple and unfussy as possible. Note that no special action is required if it is desired to assemble code for one sort of processor on a machine with a different processor.

ASSEMBLER DEFINITION

IPS allows the programmer to define new words as a sequence of assembler instructions. Note that assembly takes place in keyboard mode, compilation mode is not entered (A colon definition would result in the compiler entering compilation mode).

CODE

The introduction to an IPS named assembly routine is the word CODE followed by a name.

NEXT

The end of an assembler sequence (routine) is indicated by the word NEXT. This returns control to the emulator, and is the usual way of finishing an IPS assembly sequence.

Note that more than one exit from an assembly sequence may be provided by the multiple use of NEXT.

```
CODE a_routine (names assembly routine)

.... (first action)
....
....
.... (last action)

NEXT (returns control to the emulator)
```

Fig1: General layout of CODE definition

RCODE

The IPS assemblers allow the dubious practice of multiple entries to a routine. This is possible by marking an entry point with the word HIER/HERE in a CODE routine, and following this with the word RCODE followed by the name of the new entry point.

```
CODE a_routine

.... (a_routine actions)
....
....

HIER (deposits address on stack for RCODE following)

.... (actions common to a_routine and another_routine)

NEXT

RCODE another_routine (picks up routine address from stack)
```

Fig2: Use of RCODE construct.

HACKING

When it is necessary to write assembly routines that will be called from other assembly routines, the assembler can be used directly. This is necessary to produce replacement interrupt routines, for example.

To provide a named address, the following technique is often used:

```
0 FELD routine_name

.....     (start of subroutine)
.....

RET       (return from subroutine)
```

Fig3: Naming a directly-called assembler routine.

CODE DEPOSIT

An assembler mnemonic, e.g. LDA, results in the corresponding machine instruction being deposited at the location that $H points to (HIER/HERE) and in $H being incremented to point to the next free position.

OPERAND ORDER

The order of specifying addresses and operation follows the usual IPS principle of:

source, destination, operand.

Most conventional (algebraic) assemblers use a different order.

MNEMONICS

The IPS assemblers often use the customary mnemonics for a particular processor. Some mnemonics have been changed. Note that there is little commonality between the various IPS assemblers. Equally, there is a lack of commonality in various matters between different processors.

STRUCTURED CONTROL FLOW

The IPS assemblers use structuring words rather than labels and gotos. There are (usually) no explicit jump/branch mnemonics.

The control structures provided are (the <condition> options allowed are specific to each assembler):

a) <condition> Y? N: TH

provides an equivalent to an IF (JA? NEIN: DANN) construct;

```
....
....
      <condition> Y?
            .... (done if condition is true)
         ....
         ....
      N:
         .... (done if condition is not true)
         ....
         ....
      TH
.... (always done)
....
```

Fig4: Use of Y? N: TH construct

b) <condition> Y? TH

provides an equivalent to an IF statement without an else clause.

```
....
....
      <condition> Y?
         .... (only done if condition is true)
         ....
         ....
      TH
.... (always done)
....
```

Fig5: Use of Y? TH construct

c) BEGIN <condition> END

provides a controlled loop equivalent to the high level ANFANG ENDE? construct;

```
....
....
      BEGIN
         .... (repeated actions)
         ....
         ....
      <condition> END
.... (loop only exited if condition is true)
....
```

Fig6: Use of BEGIN END construct

d) BEGIN <condition> Y? TH/AGAIN

provides a controlled loop equivalent to the high level ANFANG JA? DANN/NOCHMAL construct.

```
....
....
      BEGIN
         ....          (start of repeated actions)
         ....
      <condition> Y?   (loop test and exit point)
         ....
         ....
         ....          (end of repeated actions)
      TH/AGAIN
....                   (loop only exited if action is false)
....
```

Fig7: Use of BEGIN Y? TH/AGAIN construct

HANDLING OF JUMP ADDRESSES

As with high level control constructs, the stack is used to manipulate jump addresses.

When the word Y? is met, the compiler deposits the appropriate jump code at HIER/HERE, increments $H and the value of $H is put on the stack. $H is then incremented to leave room for an unresolved jump address, which will be provided (resolved) later to the address specified on the stack.

When the TH is met later, the Y? jump address can now be resolved. HIER/HERE is the address that is to be jumped to. TH takes the address of the jump address off the stack, and stores HIER/HERE as the address for the jump associated with the Y?.

If a N: is met, a unconditional branch is assembled, again with an unresolved address field and with the address of the jump address left on the stack. The address left by the Y? is removed and the Y? is resolved to point to the position following the N: jump.

The word BEGIN is equivalent to HIER/HERE, and simply leaves the current address on the stack. This address is used by the END as the address of its conditional jump.

MYOPIA. It is possible to break the IPS structure rules by stack manipulation. Such deviant and devious practices are both unwise and liable to lose you friends.

TH/AGAIN

Early IPS assemblers did not support the use of TH/AGAIN. If not already available, define TH/AGAIN as follows:

```
: TH/AGAIN  ({loop start} {exit address field})
  VERT      ({exit address field} {loop start})
  NEVER END (return to loop start)
            ({exit address field})
  TH        (back fill exit address for Y?)
; (TH/AGAIN)
```

Fig8: Definition of TH/AGAIN

This definition is processor independent, and relies on the handling of the jump addresses described above.

USE OF REGULAR IPS WORDS

Because the assembler functions in interpretative mode any valid IPS words can be used in between assembly mnemonics, for instance to manipulate addresses.

The exit word NEXT can be used as often as necessary to create multiple exit points from a code word.

ASSEMBLER MACROS

A frequently required sequence of mnemonics etc. may be made a normal IPS definition. Each time the definition is invoked the sequence is run through again as if each mnemonic etc. had been entered individually. Such macros can contain low level structuring: Y? N: TH, BEGIN END. Because the high level IPS structure words are different from those used by the assemblers, it is possible to use conditional assembly within a macro.

IPS ASSEMBLER INTERFACE

Mostly, the interface between CODE definitions and IPS is via the parameter stack. At assembler level stack operations are explicit rather than automatically managed.

The stack is maintained by a pointer and appropriate machine-code instructions. If the processor already has stack-operation instructions these are usually used for the parameter stack, with the return stack simulated by a software pointer. In the case of the 6502 the machine stack is used for the return stack, with a software pointer for the parameter stack.

Before any assembly programmer is let loose at an IPS system, IPS has already taken certain machine resources into use. Particularly, certain registers and memory areas will be used for specific purposes.

The conventions should be followed, or unpredictable consequences are likely to follow.

This information can be found from

1. the relevant IPS assembler manual, and
2. the implementation notes for the particular IPS version in use.

MACHINE RESOURCES

IPS requires various machine resources to run on a particular computer. Sometimes IPS' requirements can be met by existing operating system provision, often it is necessary to bypass the host's provided operating system.

STACK AND MEMORY LAYOUT

Clearly, there are two ways that can be chosen of representing a 16 bit number on a byte oriented machine:

i) Reversed address (Little-Endian) in which the least significant byte is held in the lowest address of a byte pair.

ii) High-to-low address (Big-Endian) in which the most significant byte is held in the lowest address of a byte pair.

This is one option too many for safety!

STACK LAYOUT

The stack(s) always grow from high memory to low memory. The stack normally uses the memory convention native to the processor in use.

CODE AND DATA MEMORY LAYOUT

IPS compiled code and data grows from low memory towards high memory. IPS is designed to use the reversed address convention. In the case of those implementations which do not (6800) special conventions are necessary to allow for this.

```
           IPS word           GROWS
           <------->          ============>
_______________________________________________________
|    |    |    |    |    |    |    |    |    |    |
|    |    | LS | MS | LS | MS |    |    |    |    |
|    |    |byte|byte|byte|byte|    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
_______________________________________________________

low address                                high address

N.B. reversed address convention.
```

Fig9: IPS code and data memory layout

```
           IPS word           GROWS
           <------->          ============>
_______________________________________________________
|    |    |    |    |    |    |    |    |    |    |
|    |    | MS | LS | MS | LS |    |    |    |    |
|    |    |byte|byte|byte|byte|    |    |    |    |
|    |    |    |    |    |    |    |    |    |    |
_______________________________________________________

low address                                high address

N.B. high to low address convention.
```

Fig10: IPS code and data memory layout (6800/68000)

MEMORY MAP

The memory maps of various IPS implementations vary, but tend to have a similar pattern, as typified below:

```
#FFFF     -|   Native operating system
           |
           |
           |   Hardware ports
#CFFF     -|   x    Start of parameter stack
           |   x
           |  \x/   parameter stack grows
           |
          -|
           |   |^    code grows
           |   |
           |   |
#5FFF     -|   ||   original IPS end
           |   ||
           |   ||   (compiled IPS code)
           |   ||
#400      -|   ||   start of IPS code
           |
           |        buffers etc.
           |
#200      -|
           |
           |        return stack (depth = 128 entries)
           |
#100      -|
           |
           |        emulator, etc
           |
#80       -|
           |
           |        operating system variables
           |
#00       -|
```

Fig11: Typical IPS Memory Map

The memory map above is loosely based on the ATARI 6502 IPS implementation.

Note that the size of parameter stack is constrained by IPS code size.

RCA/COSMAC 1802 Assembler

```
(                       1802 Assembler                      )
(                    Copyright 2002 AMSAT-DL                  )
(                by Karl Meinzer, James Miller,               )

(    This program is free software; you can redistribute it   )
(    and/or modify it under the terms of the GNU General      )
(    Public License as published by the Free Software         )
(    Foundation; either version 2 of the License, or at       )
(    your option, any later version.                          )

(    This program is distributed in the hope that it will     )
(    be useful, but WITHOUT ANY WARRANTY; without even the    )
(    implied warranty of MERCHANTABILITY or FITNESS FOR A     )
(    PARTICULAR PURPOSE.  See the GNU General Public          )
(    License for more details.                                )

(    You should have received a copy of the GNU General       )
(    Public License along with this program; if not, write    )
(    to the Free Software Foundation, Inc., 59 Temple         )
(    Place, Suite 330, Boston, MA  02111-1307  USA            )

(    Assembler CDP 1802  - JRM  1997 Feb 06 [Thu] 1447 utc    )

~ CDP 1802 Cross Compiler  ~ #01D3 !t                ( Banner )

:prior i>      0 compileflag !b ;n
:int   <i      1 compileflag !b ;n
:n ,            hier $OC  !b $h incr ;n
:int rcode entrysetup ja? !O dann ;n
:int code  entrysetup ja? hier vert !O
                      dann ;n
16 feld $jerror ~ Destin. off page ~ $jerror !t
16 feld $cerror ~ PreInstr. error! ~ $cerror !t
:n $term weg $cerror syswrite ;n
:n $inadr  pdup exo #FF00 und
       =0n ja? $OC !b
          nein: weg $jerror syswrite
          dann ;n
:n $ad     zwo #FFF0 und =0n ja?   +n ,
                           nein: $term
                           dann ;n
:n $aluact zwo #FF00 und #A500 =n
          ja?   vert $ad
          nein: $term
          dann ;n

:n MX   0 $aluact ;n      :n IM    8 $aluact , ;n
```

```
:n LDN  #00 $ad ;n  :n RET   #70 , ;n     #A574 kon ADC
:n INC  #10 $ad ;n  :n DIS   #71 , ;n     #A575 kon -DC
:n DEC  #20 $ad ;n  :n LDXA  #72 , ;n     #A577 kon DC-
:n LDA  #40 $ad ;n  :n STXD  #73 , ;n     #A5F0 kon LD
:n STR  #50 $ad ;n  :n ROR   #76 , ;n     #A5F1 kon OR
:n I/O  #60 $ad ;n  :n SAV   #78 , ;n     #A5F2 kon AND
:n GLO  #80 $ad ;n  :n MARK  #79 , ;n     #A5F3 kon XOR
:n GHI  #90 $ad ;n  :n 0>Q   #7A , ;n     #A5F4 kon ADD
:n PLO  #A0 $ad ;n  :n 1>Q   #7B , ;n     #A5F5 kon -D
:n PHI  #B0 $ad ;n  :n ROL   #7E , ;n     #A5F7 kon D-
:n ->P  #D0 $ad ;n  :n NOP   #C4 , ;n     #0002 kon RS
:n ->X  #E0 $ad ;n  :n SKP2  #C8 , ;n     #0000 kon NEVER
                    :n LSIE  #CC , ;n     #0009 kon Q=1
:n IDL  #00 , ;n    :n SHR   #F6 , ;n     #000A kon D=0
:n SKP  #38 , ;n    :n SHL   #FE , ;n     #000B kon DF
:n INCX #60 , ;n    :n NEXT  #D6 , ;n     #000B kon PS

:n NOT    8 exo ;n      :n EF  11 +n ;n      :n BEGIN   hier ;n
:n Y?    #30 $ad hier $h incr ;n   :n TH    hier vert $inadr ;n
:n N:    0 Y? vert TH ;n           :n END   Y? $inadr ;n
:n $sbrt zwo 4 und >0n ja? vert ( Provoziert fehler )
                     dann $ad ;n
:n LEND      #C0 $sbrt dup 256 /n , , ;n
:n LSKP      #C4 $sbrt ;n

                    ( End 1802 Assembler )
```

Am1601 Assembler

The IPS-F1G source code must be compiled with a version of IPS-X that supports the “align” variable.

The Am1601 was designed to be IPS friendly and as such contains a number of "tricks", ... to support these "tricks" the standard IPS assembler syntax has been extended.

```
(                       Am1601 Assembler                      )
(                    Copyright 2002 AMSAT-DL                  )
(                by Karl Meinzer, James Miller,               )
(                 Lyle Johnson & Paul Willmott                )

(    This program is free software; you can redistribute it   )
(    and/or modify it under the terms of the GNU General      )
(    Public License as published by the Free Software         )
(    Foundation; either version 2 of the License, or at       )
(    your option, any later version.                          )

(    This program is distributed in the hope that it will     )
(    be useful, but WITHOUT ANY WARRANTY; without even the    )
(    implied warranty of MERCHANTABILITY or FITNESS FOR A     )
(    PARTICULAR PURPOSE.  See the GNU General Public          )
(    License for more details.                                )

(    You should have received a copy of the GNU General       )
(    Public License along with this program; if not, write    )
(    to the Free Software Foundation, Inc., 59 Temple         )
(    Place, Suite 330, Boston, MA  02111-1307  USA            )

(                Contact : vp9mu@amsat.org                    )


(    NOTE: stack comments have top on right                   )

(   "Assembler" definitions for use by IPS-X cross compiler   )

:prior i> 0 compileflag !b ;n
:int <i 1 compileflag !b ;n
:n , hier $OC !b $h incr ;n
:int code  entrysetup ja? hier vert !O
                      dann ;n
:int rcode entrysetup ja? !O dann ;n

02 align !n                      ( set even address alignment )

(                  Constants for cc codes                     )
(                  ----------------------                     )

#0 kon EQ #1 kon NE #2 kon CS #3 kon CC
```

```
#4 kon MI #5 kon PL #6 kon VS #7 kon VC
#8 kon HS #9 kon LO #A kon GE #B kon LT
#C kon GT #D kon LE #E kon AL #F kon NEF
#2 kon HI #3 kon LS

( Comparison  Unsigned   Signed )
( =           EQ         EQ     )
( !=          NE         NE     )
( >=          HS         GE     )
( >           HI CS      GT     )
( <=          LS CC      LE     )
( <           LO         LT     )


(                 Constants for PUSH & POP                    )
(                 ------------------------                    )

#00 kon PC  #10 kon PPC #20 kon HP  #30 kon FLAGS
#40 kon PSP #50 kon PSC #60 kon RSP #70 kon RSC
#80 kon EA  #90 kon RR

(     Constants for Arithmetic/Logical Instruction Operands   )
(     -----------------------------------------------------   )

#00A0 kon uN                          ( unsigned operand flag )
#00B0 kon sN                            ( signed operand flag )
#00C0 kon P0                ( parameter stack register 0 flag )
#00C0 kon P1                ( parameter stack register 1 flag )

(          Constants for SET & CLEAR Instructions             )
(          --------------------------------------             )

#00 kon FLGC #10 kon FLGZ #20 kon FLGS  #30 kon FLGO
#40 kon FLGE #50 kon FLGI #60 kon FLGIE #70 kon FLGEE

(           byte manipulation and storage primitives          )
(           ----------------------------------------          )

:n sJCODE    ( <addr> <opcode>           )
   vert      ( <opcode> <addr>           )
   dup       ( <opcode> <addr> <addr>    )
   #100 /n   ( <opcode> <addr> <MSBAddr> )
   #0F und   ( restrict range to 0-F     )
   rdo       ( <addr> <MSBaddr> <opcode> )
   oder ,    ( <LSBaddr>                 )
   #FF und , ( -                         )
;n

:n ccCODE , #0F und , ;n

:n aluCODE        ( <P1/0> <subc>             )
      zwo         ( <P1/0> <subc> <P1/0>      )
```

```
      P1 =n ja?   ( <P1/0> <subc>            )
         256 *n   ( <P1/0> <subc>*256        )
         oder     ( <opcode>                 )
         $dep     ( -                        )
      nein:       ( <numb> <uN|sN> <subc>    )
         16 /n    ( <numb> <uN|sN> <subc/16> )
         oder     ( <numb> <opcode>          )
         ,        ( <numb>                   )
         ,        ( -                        )
      dann
;n

(                      instructions                              )
(                      ------------                              )

:n sJMP   #00 sJCODE ;n
:n sJSR   #10 sJCODE ;n
:n sLOAD  #20 sJCODE ;n
:n sSTORE #30 sJCODE ;n
:n sBR    #40 sJCODE ;n
:n sBSR   #50 sJCODE ;n
:n cNLOAD #60 ccCODE $dep ;n
:n uNLOAD #70 , , ;n
:n sNLOAD #71 , , ;n

:n cJSR     #80 ccCODE $dep ;n
:n cJMP     #81 ccCODE $dep ;n
:n cLOAD    #82 ccCODE $dep ;n
:n cSTORE   #83 ccCODE $dep ;n
:n cLOADB   #84 ccCODE $dep ;n
:n cSTOREB  #85 ccCODE $dep ;n
:n cpJSR    #88 ccCODE ;n
:n cpJMP    #89 ccCODE ;n
:n cpLOAD   #8A ccCODE ;n
:n cpSTORE  #8B ccCODE ;n
:n cpLOADB  #8C ccCODE ;n
:n cpSTOREB #8D ccCODE ;n
:n cRTS     #8E ccCODE ;n
:n cpBSR    #8F ccCODE ;n

:n cBR #90 oder , , ;n

:n ADD  #00 aluCODE ;n
:n ADC  #10 aluCODE ;n

:n SBC  dup P1 =n ja?
            #50
        nein:
            #30
        dann
        aluCODE ;n
```

```
:n SUB  dup P1 =n ja?
            #40
        nein:
            #20
        dann
        aluCODE ;n

:n RSBC dup P1 =n ja?
            #30
        nein:
            #50
        dann
        aluCODE ;n

:n RSUB dup P1 =n ja?
            #20
        nein:
            #40
        dann
        aluCODE ;n

:n AND  #60 aluCODE ;n
:n OR   #70 aluCODE ;n
:n EOR  #80 aluCODE ;n
:n NOP  #90C0 $dep ;n

:n CMP  dup P1 =n ja?
            #A0
        nein:
            #B0
        dann
        aluCODE ;n

:n RCMP dup P1 =n ja?
            #B0
        nein:
            #A0
        dann
        aluCODE ;n

:n MASK #C0 aluCODE ;n
:n CPL  #D0C0 $dep ;n
:n TST  #E0 aluCODE ;n
:n NEG  #F0C0 $dep ;n

:n DUPL #00C1 $dep ;n
:n DEL  #10C1 $dep ;n
:n SWAP #20C1 $dep ;n
:n SOT  #30C1 $dep ;n
:n RTU  #40C1 $dep ;n
:n RTD  #50C1 $dep ;n
```

```
:n PTOR #60C1 $dep ;n
:n RTOP #70C1 $dep ;n
:n IDX  #80C1 $dep ;n
:n XRP  #90C1 $dep ;n
:n LSL  #00C2 $dep ;n
:n LSR  #10C2 $dep ;n
:n ROL  #20C2 $dep ;n
:n ROR  #30C2 $dep ;n
:n ASR  #90C2 $dep ;n

:n PUSHPS #D0 , , ;n
:n POPPS  #D1 , , ;n
:n PUSHRS #D2 , , ;n
:n POPRS  #D3 , , ;n
:n SET    #D4 , , ;n
:n CLEAR  #D5 , , ;n

:n cIN    #E2 ccCODE $dep ;n
:n cOUT   #E3 ccCODE $dep ;n
:n cINB   #E4 ccCODE $dep ;n
:n cOUTB  #E5 ccCODE $dep ;n
:n cpIN   #EA ccCODE ;n
:n cpOUT  #EB ccCODE ;n
:n cpINB  #EC ccCODE ;n
:n cpOUTB #ED ccCODE ;n

:n EMULATE #F0 ccCODE ;n
:n EXECUTE #F1 ccCODE ;n
:n PREPARE #F2 ccCODE ;n
:n REFRESH #F6 ccCODE ;n
:n DFX     #00F8 $dep ;n
:n 2BLIT   #00FB $dep ;n
:n JPPC    #00FC $dep ;n
:n XB      #00FD $dep ;n
:n FLAG    #FF ccCODE ;n

(                  Jump and Branch Tools                      )
(                  ---------------------                      )

:n sJSRbegin
     hier          ( push address onto IPS-X stack   )
     h2inc         ( deposit placeholder             )
;n                 ( <fixaddr>                       )

:n sJSRcomplete    ( <fixaddr>                       )
     hier          ( <fixaddr> <saveaddr>            )
     dup           ( <fixaddr> <saveaddr> <saveaddr> )
     rdo           ( <saveaddr> <saveaddr> <fixaddr> )
     $h !n         ( <saveaddr> <saveaddr>           )
     sJSR          ( <saveaddr>                      )
     $h !n         ( -                               )
;n
```

```
:n sBRbegin
     hier           ( push address onto IPS-X stack   )
     h2inc          ( deposit placeholder             )
;n                  ( <fixaddr>                       )

:n sBRcomplete      ( <fixaddr>                       )
     dup            ( <fixaddr> <fixaddr>             )
     02 +n          ( <fixaddr> <PC+2>                )
     hier           ( <fixaddr> <PC+2> <jumpadd>      )
     vert           ( <fixaddr> <jumpadd> <PC+2>      )
     -n             ( <fixaddr> <offset>              )
     hier           ( <fixaddr> <offset> <saveaddr>   )
     rdu            ( <saveaddr> <fixaddr> <offset>   )
     vert           ( <saveaddr> <offset> <fixaddr>   )
     $h !n          ( <saveaddr> <offset>             )
     sBR            ( <saveaddr>                      )
     $h !n          ( -                               )
;n

:n sJMPbegin
     hier           ( push address onto IPS-X stack   )
     h2inc          ( deposit placeholder             )
;n                  ( <fixaddr>                       )

:n sJMPcomplete     ( <fixaddr>                       )
     hier           ( <fixaddr> <saveaddr>            )
     dup            ( <fixaddr> <saveaddr> <saveaddr> )
     rdo            ( <saveaddr> <saveaddr> <fixaddr> )
     $h !n          ( <saveaddr> <saveaddr>           )
     sJMP           ( <saveaddr>                      )
     $h !n          ( -                               )

;n

:n sBSRbegin
     hier           ( push address onto IPS-X stack   )
     h2inc          ( deposit placeholder             )
;n                  ( <fixaddr>                       )

:n sBSRcomplete     ( <fixaddr>                       )
     dup            ( <fixaddr> <fixaddr>             )
     02 +n          ( <fixaddr> <PC+2>                )
     hier           ( <fixaddr> <PC+2> <jumpadd>      )
     vert           ( <fixaddr> <jumpadd> <PC+2>      )
     -n             ( <fixaddr> <offset>              )
     hier           ( <fixaddr> <offset> <saveaddr>   )
     rdu            ( <saveaddr> <fixaddr> <offset>   )
     vert           ( <saveaddr> <offset> <fixaddr>   )
     $h !n          ( <saveaddr> <offset>             )
     sBSR           ( <saveaddr>                      )
     $h !n          ( -                               )
```

```
;n

:n cJMPbegin       ( <cc>                            )
     #81 , ,       ( - ) ( cJMP opcode deposited     )
     hier          ( push address onto IPS-X stack   )
     h2inc         ( leave placeholder for address   )
;n                 ( <fixaddr>                       )

:n cJMPend         ( <fixaddr>                       )
     hier          ( <fixaddr> <saveaddr>            )
     vert          ( <saveaddr> <fixaddr>            )
     $OC !n        ( -                               )
;n

:n cJMPelse        ( <fixaddr>                       )
     AL cJMPbegin  ( <fixaddr> <fixaddr2>            )
     vert          ( <fixaddr2> <fixaddr>            )
     cJMPend       ( <fixaddr2>                      )
;n

:n cJSRbegin       ( <cc>                            )
     #80 , ,       ( - ) ( cJSR opcode deposited     )
     hier          ( push address onto IPS-X stack   )
     h2inc         ( leave placeholder for address   )
;n                 ( <fixaddr>                       )

:n cJSRcomplete    ( <fixaddr>                       )
     hier          ( <fixaddr> <saveaddr>            )
     vert          ( <saveaddr> <fixaddr>             )
     $OC !n        ( -                               )
;n


:n cBRbegin        ( <cc>                            )
     #90 oder ,    ( - ) ( cc cBR opcode deposited   )
     hier          ( push address onto IPS-X stack   )
     #0 ,          ( leave placeholder for offset    )
;n                 ( <fixaddr>                       )

:n cBRend          ( <fixaddr>                       )
     dup           ( <fixaddr> <fixaddr>             )
     01 +n         ( <fixaddr> <PC+2>                )
     hier          ( <fixaddr> <PC+2> <jumpadd>      )
     vert          ( <fixaddr> <jumpadd> <PC+2>      )
     -n            ( <fixaddr> <offset>              )
     vert          ( <offset> <fixaddr>              )
     $OC !b        ( -                               )
;n

:n cBRelse         ( <fixaddr>                       )
     AL cBRbegin   ( <fixaddr> <fixaddr2>            )
     vert          ( <fixaddr2> <fixaddr>            )
```

```
     cBRend        ( <fixaddr2>                      )
;n

( these are the traditional IPS Assembler Definitions )

:n Y? cJMPbegin ;n
:n N: cJMPelse ;n
:n TH cJMPend ;n
:n BEGIN hier ;n
:n END Y? $OC !n ;n
:n TH/AGAIN vert AL END TH ;n

(                   End Am1601 Assembler                      )
(                   --------------------                      )
```

APPENDIX A –GNU Free Documentation License

GNU Free Documentation License
Version 1.1, March 2000

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written 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 that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. 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. (For example, 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. 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 “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and 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 has been designed to thwart or discourage subsequent modification by readers is not Transparent. 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 designed for human modification. Opaque formats include PostScript, PDF, 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 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.

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 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 publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. 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 less than five).

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”, and 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. In any section entitled “Acknowledgements” or “Dedications”, preserve the section’s title, 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 as “Endorsements” or to conflict in title with any Invariant Section.

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 simply 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. 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 originalauthor 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, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they 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 quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around 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 provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

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.

APPENDIX B –GNU General Public License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all. The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.) These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program. In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable. If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program 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.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you

indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program. If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice. This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public 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. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be

guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

INDEX