P E R M A N E N T M E M O R A N D U M # M-1090

DATE October 10, 1960

SUBJECT. FRAP[1] - AN ASSEMBLY FOR PDP-1

TO PDP Distribution List FROM Edward Fredkin

Frap is designed to operate on a PDP-1 computer with at least 1024 words of memory and the minimum set of in-out equipment. The assembly program will accept as input, a paper tape, produced by a flexowriter. The tape is an isomorphic representation of the information contained in a typewritten program.

The symbolic program has two levels of structure. (1) The program is a sequence of statements and (2) a statement is a sequence of words. We adopt the following conventions to delimit statements; the words on one line of the typewritten program constitute one statement. We see that the carriage return is a delimiter with respect to statements.

To delimit words, we allow a larger set of delimiters. A word is, at present, that ordered string of characters[2] occurring between any two delimiters. We do not admit that string consisting of zero characters to the class of words. One may overdefine by saying that a delimiter followed by a delimiter is again - a delimiter. The delimiters with respect to words are: the space, the tab, the comma, and the carriage return. The comma is used to initiate a comment - the rule is that all words and delimiters following a comma (in a statement) are ignored by Frap. A statement may start with a comma, in which case the whole statement is ignored.

The Use of Words as Symbols

Most words are used as symbols that will have a numerical value. A typical example would be the use of the word "add" as a symbol

---

1. Acknowledgment
   The FRAP program and this memorandum were supplied to DEC by Mr. Edward Fredkin of Bolt, Beranek, & Newman, Cambridge, Mass.

2. A "character" being any key on the typewriter, including such as upper case, backspace, etc.

whose value equals 400000_8.[1] To represent an octal number, we use a word that contains only digits. The word "120377," which is a number, is a symbol whose value equals 120377_8. (In general, numbers will be symbols whose value is indicated by the number itself and all numbers are octal.) Some words (control words) are used as symbols for procedures that one wishes Frap to follow in order to control or modify various aspects of the assembly.

We may say that the function of the compiler is to: (1) accept a tape representing a program and to establish a correspondence between each type of word and some numerical value, (2) to replace each token of a word by the value assigned to its type, (3) to combine, within each statement, the values assigned into an output item, (4) to produce an output tape which, when read into the computer, will result in a block of successive registers containing the successive output items, and (5) to allow variations of and exceptions to the above rules by means of control words.

Each word in a statement may contribute some numerical value to the item that will be put out to correspond to the statement. We adopt the convention of combining values of words within a statement by means of the inclusive or instruction (ior).

Some words do not actually contribute numerically to the output item. Examples are labels and control words. A label is always the first word in a statement. There is a rule that if the first word in a statement is undefined, i.e., Frap does not have a numeric value that corresponds to it, Frap considers it to be the label of that statement. This label may be used anywhere in the program to refer to the statement so labeled. If the first word in a statement is defined, it is not a label.

Writing A Program

The first statement in a program is usually a comment that includes the name of the program, the date, and perhaps a description of the program. Next there might be some opd statements defining various non-standard symbols. (When Frap is loaded into the computer it has in its tables the definitions of all of the standard operation codes, as listed in the PDP-1 Manual published by the Digital Equipment Corporation.) This is most often followed by an org statement which designates the area of memory into which the program is to be

---

1. The number is not decimal, but rather octal; to the base 8.

placed Not starting with an org statement is equivalent to starting with "org 0," i.e., the program will start at register zero. The program follows, as a series of statements, and may be terminated by an end statement.

The Output

Frap will produce an output tape in read in mode. This tape will load itself in the proper area of memory if it is placed in the reader, and if the read in switch is activated.

It is desirable to terminate a read in mode tape by a jump to some location. The control word "end" will punch a final item on the tape equivalent to the words, in the statement, to the left of the word "end." This final item is not one that is stored in memory, but rather one that is interpreted by the read in mode.

The Procedure for Assembling a Program

1. If Frap is not already in memory, place the Frap tape in the reader and activate READ IN. A stop with 1 or 7725 in the program counter is normal; a stop with 7745 in the program counter indicates a check sum error.

2. Set sense switches to desired positions.

3. Turn the punch on.

4. Place the symbolic tape in the reader and start at register 1. After the tape stops reading (the completion of the first pass) reposition the tape and activate the continue switch. The computer will read the symbolic tape and punch the output tape (the second pass). At the end of each pass, the computer will stop with 27_8 in the program counter.

The Control Words Available in Frap

1. opd (operation definition) This operator is used to assign a value to the symbol following "opd." The value assigned to the symbol is that value indicated by all items to the right of the symbol being defined. "opd" is used to define the operation codes, shift numbers, etc. The opd statement must precede all other occurrences of the symbol being defined.

```
e.g.  "opd jmp 600000, defines the operation jmp
       opd * 010000, the defer bit
       opd minus 1-1,  constant -1
       opd multiply jsp mult, why write jsp?"
```

An opd statement does not result in an item on the output tape.

2. org (origin). This operator is used to reset the compiler's location counter. The location counter is made equal to the value of the org statement. The effect of an org statement is to cause Frap to place the following program in a block that starts in the location specified by the origin. "org" may be used anywhere in the program, and as often as desired.

```
e.g.  normal use:  "org 172, program starts in 172"
      other uses:  "org A + 3
                   jmp patch, jmp to patch
                   org 1000, insert patch
                   patch lac C
                    dac D
                   jmp A + 4, return"
```

An org statement does not result in an item on the output tape.

3. * (indirect addressing) The "*" is defined as a ONE in the indirect addressing bit position (bit 5). The definition statement is opd * 10000.

```
e.g.  "lac * A"
```

4. s1, s2, s3 ...s9 (shift and rotate numbers) Since a shift or rotate requires a number in the unary* number system rather than binary, "s1" through "s9" are defined as 1-9, in unary. The definition statements are:

---

* In the unary number system, the number of ones, or markers, is equal to the value of the number.

```
"opd s1    1
 opd s2    3
 opd s3    7
 opd s4   17
 opd s5   37
 opd s6   77
 opd s7  177
 opd s8  377
 opd s9  777"
```

e.g. "ral s8, rotate accumulator to the left, eight bits."

5. loc (the location of the following symbol) Normally Frap assumes that every symbol in a statement is to be combined into the output word. The exception is when a symbol is used as the label for the statement. Frap identifies labels on the following bases: (1) it is not defined by an "opd" and (2) it is the first symbol in the statement. If one wishes to have a statement with the first symbol undefined, and that symbol is not meant to be a label, then the symbol should be preceded by "loc."

```
e.g.  jsp mult
      loc A, the location of A
```

6. end (the end of the program) "end" serves three functions: (1) punching of a 6-inch leader and trailer, (2) punching of a special item at the end of the read-in-mode tape, and (3) signaling the end of the program. It is important to remember that the symbol "end" must be terminated by a delimiter, such as space or carriage return, as is true of all symbols.

Examples of various uses of "end" are:

(1) "jmp end, this causes read-in-mode to terminate and jump to zero."
(2) "jmp 1277 end, this causes jmp to location 1277."
(3) "jmp begin end, this causes a jump to the statement labeled 'begin'"

7. "+" (the plus symbol) Normally the values of every symbol on a line are combined by means of the "ior" instruction. The operator "+" causes that rule to be suspended for the symbol following the "+" and substitutes the instruction "add" for the "ior."

```
e.g.   "dac 10a + 1"  If 10a = 145, then the statement is
        equivalent to:
       "dac 145 + 1"
    or "dac 146."
```

The character "+" is not on the Flexowriter; the "&" is used in typing programs.

8. - (the minus symbol) This operator causes the "sub" instruction to be used in combining the following symbol.

```
e.g.   If 10a = 145, then
          "dac 10a - 3"
       and "dac 145 - 3"
       and "dac 142"      are equivalent.
```

9. ¢ (location counter) This symbol always has the value currently equal to the compiler's location counter. This allows symbolic reference to the location of the statement containing the operator "¢".

```
e.g.   "A szf 1
       jmp ¢ + 2, skip around
       jmp ¢ - 2, go back to the szf."
```

The program above is equivalent to the following one:

```
"A szf 1
 B jmp D
 C jmp A
 D"
```

10. con (concise code) This operator causes the concise code representation of the first three characters of the following symbol to be combined with the rest of the statement. If the symbol following "con" has less than 3 characters, the one or two righthand characters are left zero. The three characters are packed into one word, into bits 0-5, 6-11, 12-17, respectively.

```
e.g.   "ab con xyz" results in
        273031 in register "ab"
        27 is concise code for x
        30 is concise code for y
        31 is concise code for z
       "con x" has the value 270000.
```

11. pause The pause operator causes Frap to stop operating until the continue button is pushed. "pause" has no other effects. It may be used to determine when Frap has reached some point in compiling, or else to prevent a tape from running off the reader when there is no "end" on the tape.

12. fix (fix symbol table) When Frap is loaded into the computer, some definitions are permanently in the symbol table, i.e., the operation codes, the shift numbers, etc. The operator "fix" allows one to add operators to that permanent set.

```
e.g.  "opd multiply jsp 1600
       opd divide jap 1727
       opd stop hlt cla cli
       fix pause"
```

13. ext (external symbol) This operator is used to indicate symbols that are to be referenced by some other program (compiled separately).

```
e.g.  "ext mult dap return"
```

Here "mult" is the name of a routine that other programs will want to refer to.

```
e.g.  "jsp mult"
```

During a compilation, "ext" prevents the combining of the value of the symbol following it with the value of the rest of the statement The values of external symbols may be defined by means of "opd" statements, and the "fix" operator, or by a special zero pass.

Format Control

In order to allow a neat and standard listing (printout of the program) the following conventions may be used.

Set up the off-line flexo-writer in the following manner:

1. left-hand margin at 8
2. 1st tab at 24
3. right-hand margin at 78

Always tab prior to the first operation code, and nowhere else, i.e one tab per line.

e.g. "10a (tab) jmp (space) A (carriage return)
(tab) hlt"

Zero pass

Zero pass and "ext" are used in order to allow the compilation of multi-program systems. This allows some programs to make references to other programs, while allowing various programs to use internal names that are not unique. The mode of operation is as follows:

1. All external symbols are prefixed by "ext" when they are used as a label.

```
e.g.  "ext mult dap return
       lac mult -1
       .
       .
       .
       etc."
```

2. All origins of subroutines are given in symbolic form if one does not desire a fixed location for the program.

```
e.g.  "ext divide org divide
       dap
       .
       .
       etc."
```

3. When compiling, turn sense switch 3 up (on) and run each tape through the zero pass once, and in the order desired.

4. With sense switch 3 down, compile each program (two passes each), in any order.

5. Reload Frap prior to compiling some other system, or programs not connected to the set just compiled.

Sense Switch Usage

1. Sense switch 1. If this switch is down, no effect. If it is up, then value used to reset the location counter in an "org" statement is taken from the test-word switches rather than from the value of the origin statement.

2. Sense switch 2. If this switch is down, no effect. If it is up, then on the second pass a listing will be typed out on the on-line typewriter, simultaneously with the punchout. The listing is an identical copy of the program, as typed, with, in addition, two octal numbers typed at the extreme right. These two numbers are (1) the location of the output data (the instruction) and (2) the actual data put out, in octal.

3. Sense switch 3. If this switch is down, no effect. If it is up, then the computer will perform a zero pass.

4. Sense switch 6. If sense switch 6 is down, no effect. If it is up, then the symbol table will be printed out at the end of the pass (first or second pass).

Error Indications

If the computer stops running prior to completion of a pass, the cause may usually be deduced by the location of the halt (the number in the program counter) and by the number in the Accumulator. Below is a table of standard error halts.

Program Counter	Accumulator	Cause and Remedy
152	unimportant	Turn the typewriter on
200	71 or 765	Machine error on Frap not in memory correctly. Reload Frap and try again.
200	145 or 315	No more room in table area. Divide the program into smaller routines and re-assemble.
200	763	The symbol being defined in an opd statement was previously defined. To redefine the symbol, continue.
251	unimportant	Parity error (the bad character is in register 400) Continue or start at 251 to use the bad character, start at 234 to skip the bad character.
260	unimportant	Turn the typewriter on.
604	unimportant	Turn the punch on.
625	unimportant	Turn the reader on.
706	unimportant	Turn the typewriter on.
1016	unimportant	The pause caused by a pause control word; continue.

ADDENDUM

Changes to FRAP 22 January 1961 -

The following changes have been made to FRAP and they are presently incorporated in the copies of FRAP which are in the tape bin in the Computer Room at BBN.

1. FRAP now produces leader and trailer on the read in mode output tape with the seven hole punched.

2. The function of sense switch #1 is changed in the following manner.

   a. Sense switch #1 will have no effect on origin statements.

   b. If sense switch #1 is up FRAP will set the location counter to the value in the address part of the test word at the beginning of pass 0, 1, or 2.

3. The "ext" symbol has been fixed so that a tape incorporating "ext" symbols will assemble properly if it is assembled by itself with only a first and second pass.

4. The "ext" symbol has been fixed so that a line with only the ext and a symbol will assemble properly.

5. The following new symbols have been defined in FRAP.

xec — This symbol means the same as "xct" and is defined in addition to "xct" as having the octal value 100000.

nop — This symbol is defined to have the octal value 760000. It is a command for no operation.

.. — This symbol is defined as having the octal value 000000. It is a null symbol.

sb0, sb1, sb2, ..., ..., sb17 — These symbols are defined as having the octal values 000000, 000004, 000010,.....000074. They designate the register into which the accumulator will be deposited upon actuation of a sequence break on the channel corresponding to the number in the symbol. For any particular sequence break channel the accumulator will be deposited in the register

defined by the symbol for that channel, the program counter will be deposited in the register defined in the symbol plus 1, the in-out register will be deposited in the location defined by the symbol plus 2, and the computer will jump to the location defined by the symbol plus 3.

6. The symbols which previously designated the sequence break registers (registers 0 to 77) have been eliminated.

7. The symbols tyo-1, tyi-1, ppa-1, ppb-1 have been eliminated. To designate typewriter or punch 1, use the symbol defining the operation desired such as tyo or ppb followed by the symbol "c1".

8. The position of the equivalence table in FRAP has been moved down in memory so that the binary punch load package in the high position may occupy core at the same time as FRAP. The present top of the equivalence table is 7377.

9. During a listing FRAP will space over page boundaries. The listing will type out 60 lines then space 6 so that if the listing is started 3 lines below the top page boundary it will thereafter space over succeeding boundaries.

The new sense switch #1 option will be useful in two areas.

a. When making the 0 pass on a group of programs all having symbolic origins sense switch #1 may be switched up and the test word set to the desired origin of the first tape in a group while reading in the first tape. Then sense switch #1 should be switched down and succeeding tapes will have their origins defined at locations immediately following the previous program.

b. For test assembly of a program having a symbolic origin sense switch #1 may be switched up and the test word set to the value desired for the origin of the tape on the test assembly and a pass 1 and 2 performed. The sense switch need not be left up during the second pass; however, it will not affect the assembly in any way if it is.

SYMBOLS DEFINED IN FRAP 22 Jan 1961

Instruction List

Symbol Octal Value

Operation Group

add	400000
and	020000
cal	160000
dac	240000
dap	260000
dio	320000
dip	300000
dis	560000
dzm	340000
idx	440000
ior	040000
isp	460000
jda	170000
jmp	600000
jsp	620000
lac	200000
law	700000
lio	220000
mus	540000
sad	500000
sas	520000
sub	420000
xct	100000
xor	060000

Skip Group

skp	640000

basic skip group code

sma	640400
spa	640200
spi	642000
sza	640100
szf	640000
szo	641000
szs	640000
skip	640600

a sma and spa to make an unconditional skip

In-Out Transfer Group

iot	720000
rpa	720001
rpb	720002
rrb	720030
tyo	720003
tyi	720004
ppa	720005
ppb	720006
dpy	720007
srb	720021
rcb	720031
cnv	720040

Sequence Break Group

esm	720055
lsm	720054
asc	720051
dsc	720050
isb	720052
*	010000

the defer or wait symbol

loc	000000

the null symbol

..	000000

another null symbol

nop	760000

a no operation symbol

Shift-Rotate Group

ral	661000
rar	671000
rcl	663000
rcr	673000
ril	662000
rir	672000
sal	665000
sar	675000
scl	667000
scr	677000
sil	666000
sir	676000

The "channel" symbols for SB

c0	0
c1	000100
c2	000200
c3	000300
c4	000400
c5	000500
c6	000600
c7	000700
c10	001000
c11	001100
c12	001200
c13	001300
c14	001400
c15	001500
c16	001600
c17	001700
c20	000003

Symbols for the SB Storage Positions

sb0	000000
sb1	000004
sb2	000010
sb3	14
sb4	20
sb5	24
sb6	30
sb7	34
sb10	40
sb11	44
sb12	50
sb13	54
sb14	60
sb15	64
sb16	70
sb17	74

for RLS and ALT

xec	100000

Operate Group		The "extent of shift-rotate" symbols	
opr	760000	s1	1
cla	760200	s2	3
clf	760000	s3	7
cli	764000	s4	17
cma	761000	s5	37
hlt	760400	s6	77
lat	762200	s7	177
stf	760010	s8	377
lap	760300	s9	777

Control Symbols Meaning

Symbol	Meaning
→	Location Counter
+	Plus
-	Minus
org	Origin
con	Concise
ext	External
end	End
pause	Pause
opd	Operation Define