

Marinchip Spellins Checker

User Guide

For Release 1.0

by John Walker

(C) Copyright 1981 Marinchip Systems, Ltd.

All Rights Reserved

Revised June 1981

Marinchip Systems 16 St. Jude Road
Mill Valley, CA 94941 ■ (415) 383-1545

Marinchip Spellings Checker User Guide

1. Introduction	1
2. Using SPELL	2
3. How SPELL Operates	3
3.1. What is a word?	3
3.1.1. Apostrophes	3
3.1.2. Hyphens	3
3.1.3. Other punctuation	3
3.1.4. Upper and lower case	3
3.1.5. Numbers and alphanumeric symbols	3
3.2. What is a line?	4
3.3. What SPELL doesn't do	4
4. How SPELL is used	5
4.1. Checking correspondence and memoranda	5
4.2. Technical writing	5
4.3. Maintenance of large documents	6
5. Standard Dictionary Composition	7
6. Creating and updating custom dictionaries	8
6.1. Exception word lists	8
6.2. Merging exception word lists	8
6.3. Creating custom dictionaries	8
6.4. Adding words to a dictionary	9
6.5. Deleting words from a dictionary	9
6.6. Creating a compressed dictionary	9
6.6.1. What is a compressed dictionary?	10
6.6.2. How much does it save?	10
6.6.3. Should I compress my custom dictionary?	10
6.6.4. How do I compress a dictionary?	10
6.6.5. Using compressed dictionaries	12
6.6.6. If you have trouble	12
7. Support utilities	13
7.1. DCHECK - Dictionary Validator	13
7.2. WC - Word Count	13
8. Installing SPELL under NOS/MT	14
9. References	15

Marinchip Spellins Checker User Guide

1. Introduction

The Marinchip Spellins Checker, SPELL, is a tool for checking the spelling of words in text prepared on Marinchip computers. SPELL may be used to check spelling of words in any text file, and is not limited to processing files intended for use with WORD.

SPELL operates by reading a file, forming a list of all words used in the file, then looking up each word in a dictionary. Words from the file which are not found in the dictionary are output, either to the user's terminal, the printer, or to a file, so that the user may check them to make sure they are not spelling errors.

SPELL is supplied with a dictionary of the English language containing more than ten thousand words including the basic vocabularies of technology and computing. SPELL is supplied with the tools necessary to build lists of words to add to the dictionary, merge them with other lists of words, and add them to the dictionary. These tools may be used to develop custom dictionaries for a user, for a style of writing, for a field of knowledge, for a specific document, or even for an entire new language.

SPELL was designed to always err in the direction of doubting a word rather than accepting it. Unlike some other products of this kind, SPELL contains absolutely no heuristics regarding plurals, prefixes, or suffixes. The only way SPELL will accept a word is if the exact word appears in the dictionary. This guarantees that SPELL will never allow a misspelled word to be missed because it is an irregular form in English. The dictionary is compressed to save the space occupied by common forms of root words, but this is just compression; each and every form appears in the dictionary as a separate entry.

Marinchip Spelling Checker User Guide

2. Using SPELL

SPELL is supplied on a Disc Executive disc. SPELL itself may be used under either the Disc Executive or under NOS. The following instructions apply to use of SPELL under either system: for Disc Executive we assume the SPELL release disc has been installed in Drive 1. For NOS/MT, we assume SPELL has been installed on the system as described in the section "Installing SPELL under NOS/MT" below.

SPELL <file>

This command causes SPELL to read the named file and print all words it cannot find in the dictionary on the user's terminal.

SPELL +<file>

This command causes SPELL to print the words it cannot find on the standard printer device, PRINT.DEV.

SPELL <error file>=<text file>

SPELL will read the named <text file> and write a list of words not found to the named <error file>. The words not found will also be printed on the user's terminal or, if a plus sign precedes the <error file> name, on the standard printer, PRINT.DEV.

SPELL <file>,<dict1>,<dict2>,...

SPELL will read the named file and look for words from the file in the named dictionaries <dict1>, <dict2>, and so on, in addition to the standard dictionary file. The dictionary files must be in "dictionary format", prepared according to the instructions given later in this manual.

These forms are all examples of the general form of the SPELL call statement. The general form is:

SPELL [+][<error file>=][<text file>][,<dict1>,...]

As illustrated by the examples above, the plus sign, if specified, directs the listing of possible misspelled words to the printer rather than the user's terminal. The <error file>=, if present, names a file where a list of the words not found in the dictionary is to be written. The <text file>, always required, names the file containing the text to be checked. The <dict1>,... specifications name any auxiliary dictionaries in which words not found in the master dictionary are to be looked up.

Marinchip Spelling Checker User Guide

3. How SPELL Operates

In implementing a program like SPELL, numerous decisions must be made regarding the definitions used in the program. Natural languages do not obey the inviolable rules of computer languages, so these decisions must, to some extent, be arbitrary. We have tried to make our choices so as to maximize the usefulness of the tool.

3.1. What is a word?

SPELL defines a word as one or more alphabetic characters, possibly with embedded apostrophes. The following are considered words:

 this brutzfleesle Charley's L'n'h's'r

3.1.1. Apostrophes

An apostrophe will be considered part of a word only if it is surrounded on both sides by a letter. Hence, in the following sentence:

 One of 'em, Dan'l, said "It's 'whoa', not 'woe', idiot".

"Dan'l" and "It's" are considered words, but "em", "whoa", and "woe" are considered words by themselves, as the apostrophes before and after them, not being surrounded by letters, are treated as punctuation and ignored.

3.1.2. Hyphens

Hyphenated words, such as:

 drouht-stricken half-baked Franco-Prussian

are treated as separate words, that is, the hyphen is treated exactly as if it were a space.

3.1.3. Other punctuation

All other characters will be treated as blanks by SPELL. Thus, words will be read regardless of punctuation.

3.1.4. Upper and lower case

All letters in words will be converted to lower case before being tested against the dictionary. Thus, all the following words are considered identical by SPELL.

 input Input INPUT INput InPuT

3.1.5. Numbers and alphanumeric symbols

Numbers and alphanumeric symbols are ignored by SPELL. When SPELL reads words from input text, numbers are initially considered as letters in forming words. Once a word has been read, it is checked to see if it contains any numeric digits. If so, it is discarded and not

Marinchip Spelling Checker User Guide

checked against the dictionary. This seemingly confusing rule prevents the symbol "TST5IO" from being considered as two words, "TST" and "IO", as it would be if numbers were treated as punctuation by SPELL. The following symbols are examples of those ignored by SPELL.

18 1E17 ENTRY5 100F

3.2. What is a line?

As it reads the input text, SPELL ignores lines which have either a period, ".", or an exclamation point, "!" in column one. This causes SPELL to ignore command lines for WORD. All other lines will be read by SPELL and scanned for words. If this rule is undesirable for your application (for example, if you want to check a long comment for an assembly program which might have periods in column one), you can simply edit the program and change the periods to some other character (or just insert a space at the start of each line).

3.3. What SPELL doesn't do

It is important to remember exactly what SPELL does. It simply reads words from text, looks them up, and outputs the ones it does not find. This task is an important part of proofreading, and a part that is very hard for humans to do (because the brain tends to see the correct word even if a spelling error is made, especially in the case of duplicated, missing, or transposed letters). It is not, however, a replacement for proofreading, as there are many errors SPELL cannot find. For example, consider the following sentence:

There affects where too, its said.

which is a pathetic attempt to convey the thought:

Their effects were two, it's said.

Both sentences would be considered correct by SPELL, because every word in each sentence is a valid English word which appears in the SPELL dictionary. Only human proofreading can find errors of usage and meaning. What SPELL does do is eliminate the tedious and exhausting task of character by character proofreading of words. It is an interesting exercise to run SPELL on some text that has been carefully proofread. It is not at all uncommon to find one or two transposition errors that were missed.

Marinchip Spelling Checker User Guide

4. How SPELL is used

SPELL is a simple tool. Like other simple tools, SPELL may be used in various different ways depending on the size and nature of the job being done. The descriptions below cover various ways in which SPELL is applied to different problems.

4.1. Checking correspondence and memoranda

When working on a small volume of text, such as a business or personal letter, SPELL is normally run on the text using the simplest form of the call statement:

```
SPELL <file>
```

SPELL will produce a list of possible misspelled words and display them on the user's terminal. The user will then glance over them, eliminating the obvious proper nouns, place names, special terms, and abbreviations that appear. Usually a few obviously misspelled words will appear, and sometimes a few in the category, "Gee, I think that's right". After noting the obvious errors and checking the questionable ones in a dictionary, the file can be corrected. After making the corrections, SPELL may be run against the file again to make sure no errors have been introduced in making the corrections. Then the printed copy of the letter may be made.

4.2. Technical writing

In technical writing, there is usually a special vocabulary of terms specific to the field for which the writing is being done. In applications such as preparing legal documents, preparing articles for submission to technical journals and so on, the best approach is to start out by placing the output of SPELL into a file by using the form of the command:

```
SPELL <error file>=<text file>
```

This command also prints the possible errors on the terminal or printer, so the error file need not be immediately examined. After collecting error files for several documents of this type, you can use the MERGE program (described later) to merge them. Then, taking great care not to allow a misspelled word to get by, you can edit the merged file and delete all errors and words not generally applicable to the subject matter. Finally, you will have constructed a file containing the specialized vocabulary for the field in which you are writing. You can use MERGE to build a custom dictionary for that vocabulary, and henceforth you can call SPELL with a command such as:

```
SPELL <file>,LEGALESE
```

where LEGALESE is the file where you have placed the dictionary of legal terms you have compiled. SPELL will then check words against both the master dictionary of English and against your dictionary of legal terms. You can build as many custom dictionaries as you like. For example, a legal secretary preparing a filing in a patent case concerning a heart pacemaker might use:

Marinchip Spelling Checker User Guide

SPELL <file>,LEGALESE,MEDICAL,ELECTRONICS

to apply legal, medical, and electronics special vocabularies against the document.

Of course, you can explicitly create a vocabulary for a field just by typing in all the words you need rather than by scanning text and proofreading SPELL-generated error lists. This approach is most convenient when the terms needed are available in tabular form (as a physicist might, for example, type in the names of the elementary particles).

4.3. Maintenance of large documents

Large documents subject to frequent revision present a difficult obstacle to the proofreader. It is easy for a small change to be made inadvertently and not be noticed until it is too late to correct. Many technical manuals and user guides fall into this class of documents.

SPELL may be used in an ingenious way to make quality control of these large documents very easy. When the document is initially prepared, SPELL is used in the conventional way to find spelling errors. These errors are corrected and the document is proofread normally. When the document is considered "clean", SPELL is run on it again. The output of SPELL will then consist of all those terms present in the correct version of the manual but not in the dictionary. These will include abbreviations, acronyms, special vocabulary, and so on. This may then be MERGED into a dictionary which is used only for this single document. The dictionary is saved with the document.

When the document is revised, SPELL is run on the revised document, using the document dictionary as an alternate dictionary in the SPELL command. SPELL will output only questionable new words added to the document in this revision. If no spelling errors or new terms have been introduced, SPELL will output nothing. If new special words have been added, they may be checked in the output from SPELL and then be added to the document dictionary for use the next time a revision is made. Just compare this to manually proofreading a 300 page technical manual after little changes have been made all over it.

Of course, proofreading must still be done to guard against errors SPELL cannot find (such as errors of sense and grammar), but generally that proofreading is much faster and less prone to error than proofreading for spellings.

Marinchip Spelling Checker User Guide

5. Standard Dictionary Composition

SPELL is released with a standard dictionary used to look up words. This dictionary was produced by scanning various machine-readable documents and extracting a list of unique words. These words were then manually checked by looking them up in the references cited at the end of this manual.

Many different types of documents were scanned to form the dictionary. In selecting words for inclusion, the dictionary was "slanted" toward computer and technical terminology. Thus, while the dictionary works quite well for almost any general type of writing, preparers of technical manuals, especially in the computer field, will find that many of the words they use are present in the dictionary and hence do not have to be manually checked. We have tried to include the general vocabulary of technology, but obviously cannot include the vocabulary of each field, as that would cause the dictionary to grow to unmanageable size. Users who do technical writing in a specific field may use the MERGE utility to prepare word lists specific to their own fields, and optionally merge them into a new master dictionary for their own use.

In addition to the words found from scanning input text, several sets of words were included explicitly. These are:

- Greek letters (alpha, tau, upsilon)
- World countries
- World cities with populations above 3,000,000
- U.S. and Canadian cities with populations above 1,000,000
- States and Possessions of the United States
- Chemical elements
- Days, months, seasons, parts of the day
- Colors
- Units of measurement, British and Metric
- Compass points (north, southwest, etc.)
- Cardinal and ordinal numbers (fourteen, sixtieth)

Criteria for inclusion or exclusion of words were their general frequency, applicability, and possible confusion for other words. Very few abbreviations were included; only the most common, such as "etc." and "Mr." were included. Including many abbreviations greatly increases the probability that a typographical error may be a valid abbreviation and not be found by SPELL. (For example, I often mistype "at" as "st" (look at your keyboard to see why). If "st" were in the dictionary as the abbreviation for "street" or "saint", this error would not be found.)

The dictionary includes only standard American spellings. British and Canadian forms such as "colour", "centre", "cheque", "aluminium", "tyre", "storey", and "civilise" are not included. We would welcome submission of an additional word list containing these forms, so they could be supplied as an addendum which users could, at their option, add to the master dictionary.

Barbarisms such as "thru", "tho", and "enuf" have been excluded from the dictionary.

Marinchip Spelling Checker User Guide

6. Creating and updating custom dictionaries

SPELL is supplied complete with the tools you need to create lists of words, merge them with additions, and use them to build custom dictionaries for use with SPELL.

6.1. Exception word lists

As described in the chapter on use of SPELL, the list of words not found in the dictionary may be written into a file by specifying the file name on the call on SPELL. This list of words is referred to as an "exception word list" and is written to the file in alphabetic order, one word per line. These word lists are text files and may be edited with EDIT or WINDOW, or processed with any other Marinchip software which accepts text files.

6.2. Merging exception word lists

The MERGE program, supplied on the SPELL release disc, may be used to merge two or more word lists into one exception word list. For this process, MERGE is called with the command:

```
MERGE <output file>=<input>,<input>,...
```

where <output file> is the file where MERGE will write the merged word lists, and the <input> specifications name the files containing the word lists to be merged. The output file will be written in the format described above for exception word lists: one word per line with words sorted in alphabetic order. The input files need not be in this format (although they may be), as MERGE uses the same rules for scanning words as SPELL and any file acceptable to SPELL may be used as input to MERGE.

When merging exception word lists, MERGE is limited to processing the number of words that will fit in memory at one time. How many words this represents depends upon the lengths of the words and the amount of memory available on the system on which MERGE is run. The dictionary merge facilities of MERGE, described below, should be used to maintain lists too large to fit in memory.

6.3. Creating custom dictionaries

Before SPELL can use a list of words as a dictionary it must be reformatted in "dictionary file order". A dictionary file acceptable to SPELL is a normal text file. The words within it are sorted first by length (number of letters); then, within a given length, in alphabetic order. As many words of a given length as will fit within 80 characters will be placed on a line, but words of differing length will never be mixed on one line (when the length changes, a new line is begun). Words are separated by spaces. The following lines illustrate a very small custom dictionary in the correct format:

```
my no us  
but buy can cat top  
try tus vow  
zero
```

Marinchip Spelling Checker User Guide

mustard

These format restrictions allow SPELL to search the dictionary efficiently without using an unreasonably large amount of memory. There is no memory-imposed limit on dictionary size.

Once a list of words has been created, it may be placed into dictionary file order by using the MERGE utility with the following command:

```
MERGE <output dictionary>=+<input>,...
```

where <output dictionary> is the file where the dictionary is to be written and <input> is the input file. As noted, more than one <input> file may be specified (only one plus sign is used, before the first file). The dictionary created by this command may be specified as a custom dictionary on a call to SPELL.

When creating a dictionary with MERGE, all words from the <input> files must fit in memory. If this causes problems when creating large dictionaries, you may make a smaller dictionary using part of the input, then add the remaining words to it as described below.

6.4. Adding words to a dictionary

Once a dictionary has been created with MERGE, you may use MERGE to add words to it by using the command:

```
MERGE <output dictionary>=<input dictionary>+<input>,...
```

Note that this command is the same as the command used to create a dictionary except that in this command an <input dictionary> file name is specified. MERGE will read in the words from the <input> files, sort them into dictionary order (as always, the new words being added must fit in memory), then merge them with the <input dictionary> and write the result in the <output dictionary> file. The <input dictionary> and <output dictionary> must not be the same file.

The only limits on the sizes of the input and output dictionaries are those imposed by the amount of disc storage available on the system and the user's patience.

6.5. Deleting words from a dictionary

If an incorrect word is discovered in a dictionary, or you wish to remove words from a dictionary for some other reason, you may do so by calling MERGE with the command:

```
MERGE <output dictionary>=<input dictionary>-<input>,...
```

This command will delete all the words in the <input> file(s) from the <input dictionary> and write an <output dictionary> with those words removed. You would use this command rather than just editing the dictionary with EDIT or WINDOW when working with a dictionary so large that a text editor could not efficiently handle it.

6.6. Creating a compressed dictionary

Marinchip Spelling Checker User Guide

English text contains much redundancy. Hence, dictionaries stored as lists of words occupy much more space than information theory fundamentally requires. To allow large dictionaries to be stored in a small amount of disc space, and to speed up SPELL's searching of such dictionaries, a dictionary compression program is supplied on the SPELL release disc.

6.6.1. What is a compressed dictionary?

A compressed dictionary is meaningful only to SPELL. It is not a text file and may not be used as input to MERGE. Hence, the original uncompressed dictionary must be kept for future updates. Compressing a dictionary loses no information; SPELL will act exactly the same regardless of whether a dictionary is compressed.

The compression program relies on detection of standard plurals, common initial strings, and polygrams common in English text. It assumes that its input is a file already in dictionary order (if not, the output will be useless). The output of the compression program is a file in a special format acceptable to SPELL. SPELL may be given either a text file or a compressed file; it detects which format is supplied and adapts itself.

6.6.2. How much does it save?

The amount of space and time saved by compressing the dictionary depends on the nature of the words in the dictionary (especially the number of words for which both the singular and plural forms are present). Compressing the master dictionary supplied with SPELL reduced the size of the dictionary file by about 55% and speeded up the search by about the same factor (most of the time spent searching the dictionary is in simply reading the file).

6.6.3. Should I compress my custom dictionary?

If a custom dictionary is small (say, 1500 words or fewer), compression is usually not worthwhile. The dictionary takes little disc space and time to search, so compression is not worth the bother. For larger dictionaries, you must consider how often the dictionary will be changed. Compression is very time consuming, and if the dictionary will be frequently revised, you may spend more time compressing it than you save searching it. A large dictionary which is revised only rarely should be compressed.

6.6.4. How do I compress a dictionary?

Dictionaries are compressed by running the DMASH program (supplied on the SPELL release disc). All files used as output from DMASH must be created before running DMASH, even if it is used under NOS/MT. DMASH is called simply by typing its name:

DMASH

DMASH will first ask you:

Input dictionary file name:

Marinchip Spellings Checker User Guide

You should respond with the name of the uncompressed dictionary file, which must be in dictionary file order (as output by MERGE with the +). Next DMASH will ask:

Dictionary paging file:

This is a file used to page the dictionary through memory, and must be about 15% larger than the input dictionary file. It must have been created before DMASH is called, and will be used throughout the execution of DMASH. Once DMASH has finished, the contents of this file will have no meaning.

After you answer these two questions, DMASH will read in the dictionary and store it in the paging file. When this is completed, DMASH will print:

Done with dictionary file: <file name>

and ask:

File for spurious singular
forms not found, or CR:

At this time, DMASH has finished with the dictionary file, and the volume containing this file may be removed, if desired. The dictionary paging file volume must remain mounted. Part of the compression process performed by DMASH is forming the standard singulars of all words in the dictionary that "look" like plurals. If the singular form is also in the dictionary, it is marked as having a standard plural and the plural form is removed. If a file is specified for "spurious singular forms", DMASH will write in that file all suspected singulars which were not found in the dictionary. This file will consist of many non-words formed as "singulars" of words that looked like plurals but were not, or of nonstandard plurals. It will also contain the singular forms of words whose plurals were in the dictionary but whose singulars were not. After running DMASH, you can edit the "spurious singular" file, discard all the meaningless words, and add the missing singular forms to your dictionary, if you desire. If you answer the question with just a carriage return, DMASH will not write a "spurious singular" file. Creating this file really has to do with dictionary completeness, not compression. It is provided as an option in DMASH because it can be done with no expense in additional time as a by-product of the compression process.

After answering this question, DMASH will run for a long time, performing the plural compression function. When this is done, the following messages will be displayed:

Done with spurious singulars file: <file name>
Eliminated <number> plurals, saving <number> bytes.

and you will be asked:

File for compressed dictionary:

At this time, DMASH is done with the "spurious singular" file, and the volume containing it may be dismounted, if desired. The volume

Marinchip Spelling Checker User Guide

containing the dictionary paging file must remain mounted. You should now enter the name of the file where the compressed dictionary is to be written. Remember that this file must have been created before DMASH was initially called, and just to be safe, should be as large as the input dictionary file.

After you specify the output file, DMASH will write the compressed dictionary to it, performing the rest of the compression process. This often takes as long as the plural compression phase did. Finally, DMASH will display the message:

Compressed <number> polygrams, saving <number> bytes.

and exit. The compressed dictionary will be stored in the output file.

6.6.5. Using compressed dictionaries

Although compressed dictionaries are not text files, they are structured so that the SIZE utility may be used to determine their size. Although SIZE will say that a compressed dictionary is a text file, a compressed dictionary should be copied only with BCOPY. Using TCOPY to copy it, or attempting to edit it will destroy control information in the file which SPELL needs to decompress the data while reading it. Other than these precautions, once a dictionary is compressed, it may be used just like an uncompressed dictionary in calls on SPELL.

6.6.6. If you have trouble

DMASH can only produce a proper compressed dictionary if its input is in correct dictionary order. If you have trouble with DMASH, try running DCHECK (described below) to make sure the input dictionary is properly formatted. If the dictionary is not properly formatted, run it through MERGE using the instructions for creating a dictionary. This will restore the correct format.

7. Support utilities

7.1. DCHECK - Dictionary Validator

DCHECK <file>

DCHECK reads a dictionary in word list (not compressed) form and makes sure that the rules of construction expected by SPELL are met. That is, words must be sorted first by length, then by alphabetic order, with no two words of differing length on the same line. If the dictionary file is properly formed, DCHECK will produce no output. If the dictionary contains format errors, DCHECK will print messages identifying the misplaced word or words.

If the MERGE utility is properly used, there should be no need to run DCHECK, as the output of MERGE is always a properly formatted word list dictionary. However, if you have modified the dictionary using some other tool or have developed your own program which constructs dictionary files, DCHECK should be run on the dictionary before using it with SPELL or compressing it with DMASH, just to make sure the dictionary is correctly formatted.

7.2. WC - Word Count

WC <file>

WC reads the file named on its call statement and prints the number of lines and the number of words in the file.

Marinchip Spelling Checker User Guide

8. Installing SPELL under NOS/MT

SPELL will work under either the Disc Executive or under NOS/MT. When used under NOS/MT, SPELL uses special rules to search for dictionary files which make it more convenient to use in multi-user environments.

The executable file SPELL may be placed in any directory. To allow all users to access it conveniently, it is normally placed in the system executable library file, 1:BIN. All users may then call SPELL with the commands described in this manual. Any dictionary file used by SPELL, whether the standard dictionary "DICT" or any alternate dictionary file named by the user, will first be looked for in the user's assumed directory. If a file with the given name is found there, it will be used. If the dictionary is not found in the user's working directory, the file will be looked up in the directory "1:SOURCE". Thus, standard dictionaries such as the master dictionary "DICT" and any special dictionaries to be made available to all users should be placed in "1:SOURCE". Users may override any of these standard dictionaries or use custom dictionaries of their own creation by placing them in their own working directories or other files.

The automatic search rule described above applies only to dictionary files named with no explicit directory, volume, or unit specification. If the explicit location of the file is given, as in:

```
SPELL ZORCHMANUAL,V18:SPDICT/URDU
```

the dictionary file will be looked up with that name. If the file cannot be found an error message will be given and no further search will be done.

Marinchip Spelling Checker User Guide

9. References

20,000 Words (Seventh Edition), Louis A. Leslie, McGraw-Hill, 1977 (ISBN 0-07-037392-2).

Oxford English Dictionary, Oxford University Press, 1971.

The Washington Post Deskbook on Style, Robert A. Webb, McGraw-Hill, 1978 (ISBN 0-07-068397-2).

The New International Atlas, Rand McNally, 1980.

Handbook of Chemistry and Physics (53rd Edition), Chemical Rubber Company, 1972.