bcc	title	prefix/class-number.revision
	ITP Diagnostic Program	DITP/W-39

checked	authors	approval date	revision date
Don Lodge	Butler W. Lampson	12/18/69	
checked		classification	
		Working Paper	
approved Mel		distribution: Company Private	pages: 6

ABSTRACT and CONTENTS

The comprehensive diagnostic for all non-branching ITP instructions is described. All information necessary for operating the program is (hopefully) given.

ITP DIAGNOSTIC

This program tests all ITP instructions and features except

branches
UPS, UPOT, UPIN, XCA, CCA, EXU
pops
addressing (but the program uses all addressing features except absolute indirection)
traps (illegal opcodes and memory protection)
input/output

It works in the following way.

Instructions are described by a table called INSTAB. Each entry in this tables describes one instruction and contains 8 words:

Ø) the instruction under test. If it addresses memory, its address field contains NOP2.

1) the address of the result table for the instruction
2) the address of a word containing the value which should be in A after execution of the instruction
3) same for B
4) same for X
5) same for NOP2
6) same for skip. The value is zero if the instruction does not skip, -1 if it does.
7) the number of words in a result table entry for this instruction.

The contents of A,B,X and NOP2 before instruction execution are contained in SA, SB, SX, and OP2; the addresses of these locations are therefore frequent occupants of words 2-5 of an instruction table entry. For example, the

instruction table entry for STA is

```
STA    NOP2
Ø
SA
SB
SX
SA
=Ø
Ø
```

indicating that STA has no result table, does not change ABX, never skips and leaves the old contents of A in the location addressed by it.

The result table takes care of the possibility that one of the results of an instruction might not be the same as any of its inputs. In this case, the result table contains an entry for each set of arguments which may be presented to the function. The address of the entry is put into X before the instruction table is accessed, so that an indexed address in words 2-6 will refer to the result table entry. The table is set up by running the diagnostic with the SKE instruction which does the comparison replaced by an STA. Here is the instruction table entry for SKR

```
SKR     NOP2
XXXX
SA
SB
SX
2B7
2B7+1
2
```

It has two result table words, one for the contents of the effective address and one for skip/ noskip. The address of the current result table entry is initialized from word 1 of the instruction table and is incremented by word 7 after each execution of the instruction under test. Its current value is kept in RESPTR.

Arguments are presented to instructions (the same arguments for all instructions) from a table called ARGTAB. The four words of this table addressed by ARGPTR are used to set up A, B, X and NOP2 before the instruction under test is executed. ARGPTR is initialized to ARGTAB when testing of an instruction is started and is incremented by one after each trial. NRES trials are made for each instruction. Note that each word of ARGTAB is used successively as A, B, X and NOP2 (except the first and last 3).

The following flags and pointers control execution of the diagnostic. Numbers in parentheses are the values they are initialized to

BEGINS (INSTAB) address of first INSTAB entry to test
ENDINS (EINST-8) address of last INSTAB entry to test
BEGCHM (Ø) address of first memory location included in checksum
ENDCHM (10000B) address of last memory location included in checksum
EWAITF (-1) if -1, wait for carriage return to be typed after an error before proceeding, if Ø, proceed at one.
DWAITF (-1) if -1, wait for carriage return to be typed after testing all instructions before looping to test them again; if Ø, proceed at one.

IWAITF (Ø) if -1, wait after testing each instruction before looping to test the next one; if Ø, proceed at once.

TRPTF (Ø) if -1, repeat the current test with the current operands indefinitely; if Ø, sequence through tests normally.

IRPTF (Ø) if -1, request the test of the current instruction indefinitely; if Ø, sequence through instructions normally.

CHKSF (-1) if -1, memory is checksummed after each instruction has been tested to see if execution of that instruction (or foulups in the diagnostic) improperly altered memory.

CPRINT (-1) if -1, prints a message after a compare error; if Ø, does not.

SPRINT (-1) if -1, prints a message after a checksum error; if Ø, does not.

The following locations contain useful information:

INSPTR	address of INSTAB entry for instruction under test
RESPTR	address of result table entry for operands under test
ARGPTR	address of 4 words used to initialize ABX and NOP2
ERRCNT	number of errors detected
LCTR	numberof complete loops performed
ICTR	number of instructions tested in current loop
TCTR	number of trials of current instruction

The following locations contain entry points to the various loops:

ESTART	start program
EMTEST	test all instructions
EMTLOOP	test one instruction
EMTSTLP	test one case of one instruction

There are two kinds of errors: comparison and checksum. A comparison error occurs when the new contents of the register, and effective address after execution of an instruction are not what the diagnostic thinks they ought to be. What is printed is:

```
COMPARE ON instruction AFTER nØ:n1:n2
old a  old b  old x  old memory
new a  new b  new x  new memory
```

where nØ is the number of complete cycles, n1 is the number of instructions tested in the current cycle, and n2 the number of trials of the current instruction.

A checksum error (see CHKSF above) prints

```
CHECKSUM ON instruction AFTER n1:n2
```

only.

After an error a wait occurs if the proper flag is set. Execution then continues by retrying the instruction which failed. In the case of a compare error, a checksum is tried first, which may elicit a checksum error.

The program is loaded at 10000B in the ITP address space. Since self-fill starts the ITP address space at 10000B in memory, the program itself starts at 20000B. The attached table gives the addresses of all the magic cells named above relative to 20000B.

```
PAGE       SYSTEM    (LAMPSON)/LISTING OF DITP    12/14/69 0345:33

00000                      LIST
00000  =1          940F    EQU     =1


* MAIN INSTRUCTION DIAGNOSTIC FOR ITP

* LOCATIONS OF INTEREST TO THE USER, DOCUMENTED IN THE MANUAL

00000              ESTART  BRU     START
00001              EMTEST  BRU     MTEST
00002              EMTSTLP BRU     MTSTLP
00003              EMTLOOP BRU     MTLOOP

00004              BEGINS  ZRO     INSTAB
00005              ENDINS  ZRO     EINST-8
00006              BEGCHM  ZRO     0
00007              ENDCHM  ZRO     100000B

00010              EWAITF  DATA    -1
00011              DWAITF  DATA    -1
00012              IWAITF  DATA    0
00013              TRPTF   DATA    0
00014              IRPTF   DATA    0
00015              CHKSF   DATA    -1
00016              CPRINT  DATA    -1
00017              SPRINT  DATA    -1

00020              INSPTR  ZRO     0
00021              RESPTR  ZRO     0
00022              ARGPTR  ZRO     0
00023              ERRCNT  ZRO     0
00024              LCTR    ZRO     0
00025              ICTR    ZRO     0
00026              TCTR    ZRO     0


* MACROS
MSG MACRO D,G,1
 LDA *+2
 BRU G(1)
 MSTR (D(1))
G(1) BRM PMSG
 ENDM

MSTR MACRO D
 IF 940F
 ZRO *+1
 ASC 'D(1)/'
 ELSE
 ZRO *+1,3
Q1 NCHR (D(1))
 RPT (Q2=1,Q1)
 DATA ' D($Q2)'+240B
```