INS8073 PROGRAMMER'S REFERENCE CARD

National
Semiconductor

March 1981

COMMAND SUMMARY

NEW expr: Establishes a new start-of-program address equal to the value of 'expr'. NSC Tiny BASIC then executes its initialization sequence. If the value of 'expr' points to a ROM address, the NSC Tiny BASIC program which begins at this address will be automatically executed. Program memory is not altered by this command.

NEW: Sets the end-of-program pointer equal to the start-of-program pointer so that a new program may be entered. If a program already exists at the start-of-program address, it will be lost.

RUN: Runs the current program.

CONT: Continues execution of the current program from the point where execution was suspended (via a STOP, console interrupt, or reset).

LIST [expr]: Lists the current program (optionally starting at the line number specified by [expr]).

STATEMENT SUMMARY

REM anything: Remark (no operation).

CLEAR: Initializes all variables to 0, disables interrupts, and resets all stacks (GOSUB, FOR-NEXT, DO-UNTIL).

[LET] var = expr: Assigns expression value to variable.

[LET] STAT = expr: Sets the STATUS word equal to the least significant byte of 'expr'. When the STATUS word is used to enable interrupts at the hardware level, interrupt processing will be deferred for one statement.

[LET] @factor = expr: Sets the memory location pointed to by 'factor' equal to the least significant byte of 'expr'.

[LET] $factor = "string": Assigns a string in RAM starting at the address 'factor'. Strings are terminated by a carriage return.

[LET] $factor = $factor: Memory to memory assignment (copy).

PRINT expr: Prints the value of 'expr'.

PRINT "string": Prints the string.

PRINT $factor: Prints the string starting at the memory address 'factor'.

IF expr [THEN] statements: Remainder of the program line is executed if 'expr' is true (non-zero).

FOR var = expr TO expr [STEP expr]: FOR loop initialization. Loops may be nested to four levels.

NEXT var: FOR loop termination.

DO: DO loop initiation. DO loops may be nested to eight levels.

UNTIL expr: DO loop termination.

GO TO expr: Transfer control to statement number 'expr'.

GO SUB expr: Call subroutine at statement number 'expr'. Subroutines may be nested to eight levels.

RETURN: Return from subroutine.

INPUT var: Read value from console into variable.

INPUT $factor: Read string from console into memory beginning at address 'factor'.

LINK expr: Links to an assembly language subroutine which begins at address 'expr'. A "RET" instruction in this routine will cause continuation of the NSC Tiny BASIC program.

ON 1 or 2 expr: Interrupt processing definition. When interrupt number 1 or 2 occurs, NSC Tiny BASIC will execute a GOSUB beginning at line number 'expr'. If 'expr' is zero, the corresponding interrupt is disabled at the software level.

DELAY expr: Delay for 'expr' time units (nominally milli-seconds, 1-1040). DELAY 0 gives the maximum delay of 1040 milliseconds.

STOP: Terminate program execution. A message is printed and the Microinterpreter returns to COMMAND mode.

OPERATOR SUMMARY

Arithmetic operators:	addition	+
	subtraction	-
	multiplication	*
	division	/
Relational operators:	less than	<
	greater than	>
	equal to	=
	not equal to	<>
	less than or equal to	<=
	greater than or equal to	>=
Logical operators:	logical AND	AND
	logical OR	OR
	logical NOT	NOT

FUNCTION SUMMARY

@factor: The memory/peripheral address for memory-I/O read/write operations.

STAT: STATUS register

TOP: Top-Of-Program address (first available memory address after end-of-program byte).

INC (x), DEC (x): Increment or Decrement a memory location (non-interruptable for multiprocessing).

MOD (x,y): Modulus function (remainder of x/y).

RND (x,y): Random number generator (in interval x,y).

ERROR CODE SUMMARY

If NSC Tiny BASIC encounters an error condition in RUN or COMMAND mode, it will alert the user by printing out 'ERROR', followed by an error number. Error numbers are defined as follows:

1. Out of memory
2. Statement used improperly
3. Unexpected character (after legal statement)
4. Syntax error
5. Value (format) error
6. Ending quote missing from string
7. GO target line does not exist
8. RETURN without previous GOSUB
9. Expression or FOR-NEXT, DO-UNTIL or GOSUB-RETURN nested too deeply
10. NEXT without previous matching FOR
11. UNTIL without previous DO
12. Divison by zero

NOTES:

1. *In the instruction syntax, items in brackets [...] are optional. '**Expr**' stands for any expression; '**var**' stands for any variable; '**factor**' stands for any of the following: a single number, a single variable, a single function, or any expression enclosed in parenthesis. The following are '**factors**':*

A decimal number	(-32767 to +32767)
A hexadecimal number	(#hexvalue)
A variable	(A to Z)
Parenthesized expressions	(expr)

2. *Variables are single-letters, A to Z.*

3. *All arithmetic is 16-bit signed integer (-32767 to +32767).*

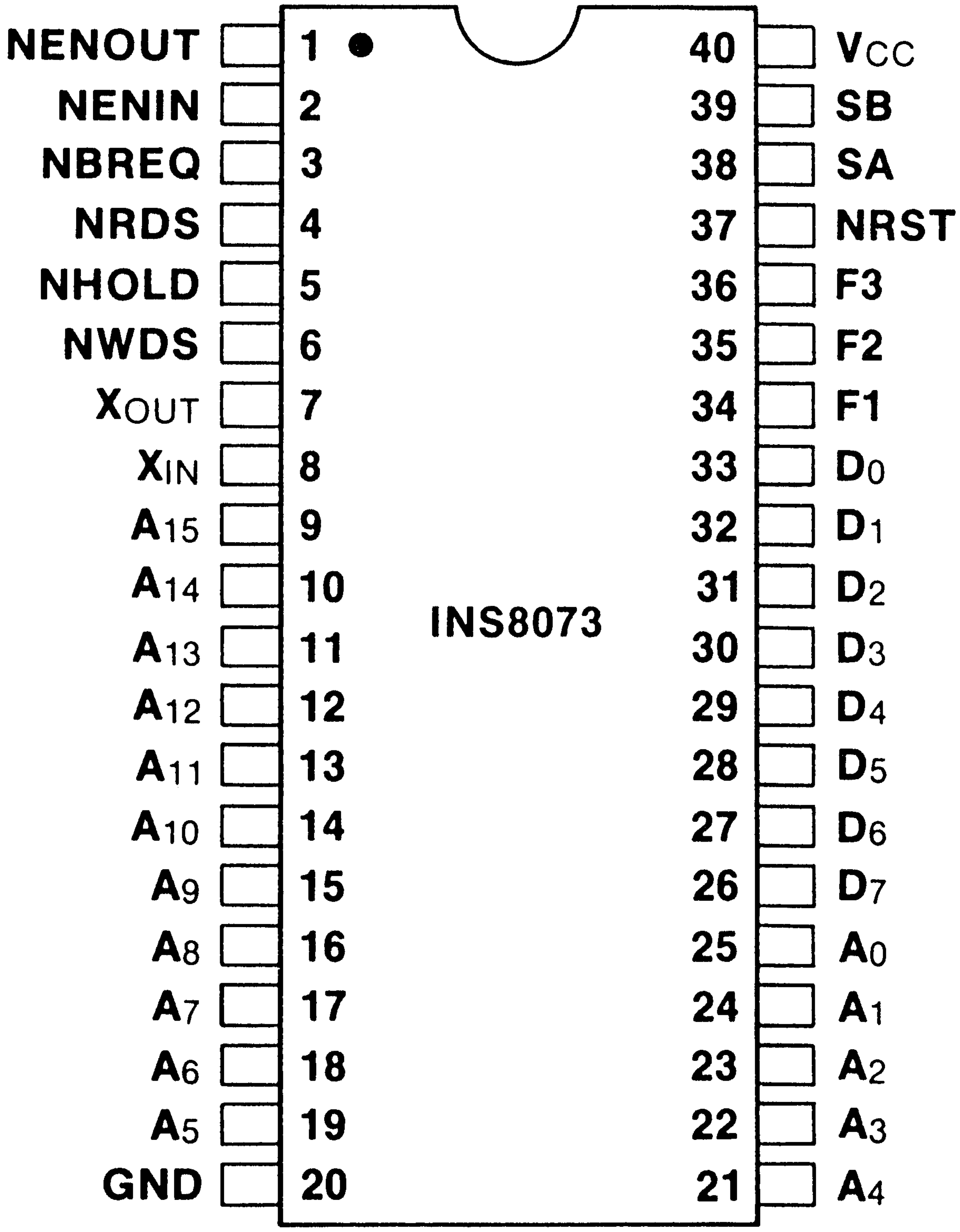
4. *All statements except INPUT may be used in COMMAND mode.*

5. *Multiple statements may be used on the same program line, with statements separated by a colon (:).* **The colon** *is especially useful as it allows FOR-NEXT and DO-UNTIL to be used in the COMMAND mode.*

6. *The PRINT and INPUT statements usually specify a list of one or more items (variables, etc.) which are separated by commas. If the PRINT list ends with a semicolon (;), the carriage return/line feed at the end of the line will be suppressed.*

number as the simplest case). *Expressions in an INPUT list may be separated by blanks or commas. There must be at least as many expressions in the input list as variables in the INPUT statement. If a console input error is detected, a message will be printed and the INPUT statement re-executed. The correct response to an 'INPUT $factor' statement is a string which is terminated by a carriage return.*

PIN CONFIGURATION



NENOUT 1
NENIN 2
NBREQ 3
NRDS 4
NHOLD 5
NWDS 6
X_{OUT} 7
X_{IN} 8
A_{15} 9
A_{14} 10
A_{13} 11
A_{12} 12
A_{11} 13
A_{10} 14
A_9 15
A_8 16
A_7 17
A_6 18
A_5 19
GND 20
INS8073
40 V_{CC}
39 SB
38 SA
37 NRST
36 F3
35 F2
34 F1
33 D_0
32 D_1
31 D_2
30 D_3
29 D_4
28 D_5
27 D_6
26 D_7
25 A_0
24 A_1
23 A_2
22 A_3
21 A_4