

EDUCATOR-Z-80<sup>© ®</sup>

INTRODUCTION

EDUCATOR-Z-80 is a self-bootstrapping software package designed to run on The Digital Group, Inc., (DGI) Z-80 micro-computer under the TV-Cassette Operated System. It should be immediately operable on any DGI Z-80 system which has been wired according to the DGI recommended standards. EDUCATOR-Z-80 requires a minimum of 5K of RAM.

INSTRUCTION SET

The EDUCATOR-Z-80 command/instruction set is a subset of the Z-80 instruction set. The commands implemented with EDUCATOR-Z-80 include commands which do not in any way effect the content of memory or require memory access, program sequence control, the stack or stack-pointer, input/output or interrupt control. All other commands and their variants are included in the instruction set (over 400 commands and variants are implemented.) Since instructions are to be executed singly from the keyboard, there was no need to incorporate any of the Jump, Jump-Relative, Call, Return or Resart instructions. However, since the Flags are displayed dynamically, it is very easy to determine whether or not a given conditional Jump, Call or Return would cause a transfer of control by observing the setting for the Flag whose status is being tested. The commands which alter memory were not implemented because anything which can be done to memory can be done to the registers and because it was not considered practical to attempt to display both memory and the registers - Flags. Stack and Stack-Pointer operations were excluded for the same reasons as memory alteration instructions. Finally, no instruction whose action could not be readily observed was implemented; e.g., Input/Output, Halt, DI/EI, IMx, etc.)

In the interest of showing the operation of as many instructions as possible, all of the working registers except IX and IY are shown on the screen and can be manipulated by the instruction set appropriately. In addition, the "PRIME" register set is also available through the use of the appropriate Exchange Commands. All instructions are shown as they would appear in the Z-80 Assembly Language.

The instruction subset and the valid operands for each instruction are shown as follows:

REGISTERED TRADE MARK and  
Copyright \* \* \* \* Digital Group Software Systems, Inc., P.O. Box 1086,  
Arvada, Colorado 80001.

COMMAND	OPERANDS	DESCRIPTION OF OPERATION	-----FLAGS-----							
			S	Z	Ø	H	Ø	V	N	C
ADC	n	ADD the value of the CARRY-FLAG and the value of the immediate operand n to the contents of the Accumulator	D	D	Ø	D	Ø	V	Ø	D
ADC	s	ADD the value of the CARRY-FLAG and the contents of the single-register s to the contents of the Accumulator	D	D	Ø	D	Ø	V	Ø	D
ADC	HL,dd	ADD the value of the CARRY-FLAG and the contents of the double-register pair dd to the contents of the double-register pair HL	D	D	Ø	X	Ø	V	Ø	D
ADD	n	ADD the value of the immediate operand n to the contents of the Accumulator	D	D	Ø	D	Ø	V	Ø	D
ADD	s	ADD the contents of single-register s to the contents of the Accumulator	D	D	Ø	D	Ø	V	Ø	D
ADD	HL,dd	ADD the contents of double-register pair dd to the double-register pair HL	N	N	Ø	X	Ø	N	Ø	D
AND	n	LOGICALLY-AND the value of the immediate operand n to the contents of the Accumulator	D	D	Ø	1	Ø	P	Ø	Ø
AND	s	LOGICALLY-AND the contents of single-register s to the contents of the Accumulator	D	D	Ø	1	Ø	P	Ø	Ø
BIT	#,s	TEST bit # of single-register s and set the ZERO-FLAG accordingly	X	D	Ø	1	Ø	X	Ø	N
CCF		COMPLEMENT the value of the CARRY-FLAG; if it is zero make it one or if it is one, make it zero	N	N	Ø	X	Ø	N	Ø	D
CP	n	COMPARE the value of the immediate operand n with the contents of the Accumulator	D	D	Ø	D	Ø	V	1	D
CP	s	COMPARE the contents of the single-register s with the contents of the Accumulator	D	D	Ø	D	Ø	V	1	D
CPL		COMPLEMENT the contents of the Accumulator by changing all the ones to zeros and all the zeros to ones	N	N	Ø	1	Ø	N	1	N
DAA		DECIMAL-ADJUST the value in the Accumulator (after an arithmetic command using decimal numbers)	D	D	Ø	D	Ø	P	N	D
DEC	dd	DECREMENT (subtract 1 from) the contents of double-register pair dd	N	N	Ø	N	Ø	N	N	N
DEC	s	DECREMENT (subtract 1 from) the contents of single-register s	D	D	Ø	D	Ø	V	Ø	N
EX	AF,AF <sup>1</sup>	EXCHANGE the Accumulator and FLAGS of the current set with the Accumulator and FLAGS of the PRIME set	?	?	Ø	?	Ø	?	?	?

COMMAND	OPERANDS	DESCRIPTION OF OPERATION	-----FLAGS-----										
			S	Z	H	V	N	C	P				
EX	DE,HL	EXCHANGE the contents of double-register pair DE with the contents of double-register pair HL	N	N	Ø	N	Ø	N	N	N	N	N	N
EXX		EXCHANGE the double-register pairs BC, DE and HL of the current set with the double-register pairs BC, DE and HL of the PRIME set	N	N	Ø	N	Ø	N	N	N	N	N	N
INC	dd	INCREMENT (add 1 to) the contents of double-register pair dd	N	N	Ø	N	Ø	N	N	N	N	N	N
INC	s	INCREMENT (add 1 to) the contents of single-register s	D	D	Ø	D	Ø	V	Ø	N	Ø	N	N
LD	dd,nn	LOAD the double-register pair dd with the <u>two</u> byte immediate value nn	N	N	Ø	N	Ø	N	N	N	N	N	N
LD	s,n	LOAD the single-register s with the immediate value n	N	N	Ø	N	Ø	N	N	N	N	N	N
LD	s,s	LOAD the <u>first</u> single-register with the contents of the <u>second</u> single-register	N	N	Ø	N	Ø	N	N	N	N	N	N
NEG		NEGATE the contents of the Accumulator (perform the two's complement; can be accomplished by a CPL followed by an INC A or by subtracting the contents of the Accumulator from zero)	D	D	Ø	D	Ø	V	1	D	Ø	N	N
NOP		NO-OPERATION do nothing	N	N	Ø	N	Ø	N	N	N	N	N	N
OR	n	LOGICALLY-OR the value of the immediate operand n to the contents of the Accumulator	D	D	Ø	Ø	Ø	P	Ø	Ø	Ø	N	N
OR	s	LOGICALLY-OR the contents of single-register s with the contents of the Accumulator (Use OR A to clear the CARRY-FLAG and set the SIGN-FLAG and ZERO-FLAG accordingly to the contents of the Accumulator)	N	N	Ø	N	Ø	N	N	N	N	N	N
RES	#,s	RESET (force to be zero) bit # of single-register s	N	N	Ø	N	Ø	N	N	N	N	N	N
RL	s	ROTATE the contents of single-register s left one bit position with the 7-bit going to the CARRY-FLAG and the CARRY-FLAG going to the zero bit	D	D	Ø	Ø	Ø	P	Ø	D	Ø	N	N
RLA		ROTATE the contents of the Accumulator left one bit position with the 7-bit going to the CARRY-FLAG and the CARRY-FLAG going to the zero bit	N	N	Ø	Ø	Ø	N	Ø	D	Ø	N	N
RLC	s	ROTATE the contents of single-register s left one bit position with the 7-bit going to both the CARRY-FLAG and the zero bit	D	D	Ø	Ø	Ø	P	Ø	D	Ø	N	N

COMMAND	OPERANDS	DESCRIPTION OF OPERATION	-----FLAGS-----													
			S	Z	Ø	H	Ø	V	N	C	P	D				
RLCA		ROTATE the contents of the Accumulator left one bit position with the 7-bit going to both the CARRY-FLAG and the zero-bit	N	N	Ø	Ø	Ø	N	Ø	D						
RR	s	ROTATE the contents of single-register s right one bit position with the zero-bit going to the CARRY-FLAG and the CARRY-FLAG going to the 7-bit	D	D	Ø	Ø	Ø	P	Ø	D						
RRA		ROTATE the contents of the Accumulator right one-bit position with the zero-bit going to the CARRY-FLAG and the CARRY-FLAG going to the 7-bit	N	N	Ø	Ø	Ø	N	Ø	D						
RRC	s	ROTATE the contents of single-register s right one-bit position with the zero-bit going to both the CARRY-FLAG and the 7-bit	D	D	Ø	Ø	Ø	P	Ø	D						
RRCA		ROTATE the contents of the Accumulator right one-bit position with the zero-bit position going to both the CARRY-FLAG and the 7-bit	N	N	Ø	Ø	Ø	N	Ø	D						
SBC	n	SUBTRACT the values of the CARRY-FLAG and the immediate value n from the contents of the Accumulator	D	D	Ø	D	Ø	V	1	D						
SBC	s	SUBTRACT the values of the CARRY-FLAG and the contents of single-register s from the contents of the Accumulator (Try SBC A to set all the bits in the Accumulator to the value of the CARRY-FLAG)	D	D	Ø	D	Ø	V	1	D						
SBC	HL,dd	SUBTRACT the values of the CARRY-FLAG and the contents of double-register pair dd from the contents of double-register pair HL	D	D	Ø	X	Ø	V	1	D						
SCF		SET the CARRY-FLAG to a 1	N	N	Ø	Ø	Ø	N	Ø	1						
SET	#,s	SET bit # of single-register s to a 1	N	N	Ø	N	Ø	N	N	N						
SLA	s	SHIFT the contents of single-register s left one-bit position and insert a 0 into the zero-bit; the 7-bit goes to the CARRY-FLAG	D	D	Ø	Ø	Ø	P	Ø	D						
SRA	s	SHIFT the contents of single-register s right one-bit position and insert a value equal to what the 7-bit was into the 7-bit; the zero-bit goes into the CARRY-FLAG	D	D	Ø	Ø	Ø	P	Ø	D						
SRL	s	SHIFT the contents of single-register s right one-bit position, shift a 0 into the 7-bit and the zero-bit to the CARRY-FLAG	D	D	Ø	Ø	Ø	P	Ø	D						

<u>COMMAND</u>	<u>OPERAND</u>	<u>DESCRIPTION OF OPERATION</u>	<u>-----FLAGS-----</u>
			P
			S Z Ø H Ø V N C
SUB	n	SUBTRACT the value of the immediate value n from the contents of the Accumulator	D D Ø D Ø V 1 D
SUB	s	SUBTRACT the contents of single-register s from the contents of the Accumulator	D D Ø D Ø V 1 D
XOR	n	LOGICALLY-EXCLUSIVE-OR the value of the immediate operand n to the contents of the Accumulator	D D Ø D Ø P Ø Ø
XOR	s	LOGICALLY-EXCLUSIVE-OR the contents of single-register s to the contents of the Accumulator (Try XOR A to clear the Accumulator)	D D Ø D Ø P Ø Ø

Where: n = a one-byte immediate value  
 nn = a two-byte immediate value  
 s = any one of the single registers displayed B,C,D,E,H,L and A  
 dd = any of the double-register pairs displayed BC, DE, or HL

Values for FLAGS: D = Depends on command and results  
 X = Meaningless  
 N = Not Changed  
 V = Depends on Overflow  
 P = Depends on Parity  
 Ø = Flag is reset  
 1 = Flag is set

## IMMEDIATE OPERANDS

Many of the instructions supported by EDUCATOR-Z-80 require immediate operands. An immediate operand is a constant value which is part of the instruction being executed and, usually, immediately follows the operation code of the instruction; hence the name immediate.

Whenever a single byte or sometimes double byte "constant" is required in a program, its inclusion as the immediate value of an appropriate instruction will serve to reduce the length of a program because there is no need to address the value directly. Immediate values have an implied address which is the address of the byte following the opcode. This address is supplied by the Program-Counter Register automatically whenever an immediate-type instruction is executed.

EDUCATOR-Z-80 supports six different types of immediate values as part of the input command and defaults to split-octal. The format of the immediate operands are consistent with the format for the Assembler. There are two general kinds of immediate values: Character and Numeric.

CHARACTER IMMEDIATE VALUES - A character immediate value is shown as a character string from 1 to X characters long enclosed in single quotes (example - 'A'). Whenever the length of the string is greater than the length required by a specific instruction, only the rightmost n character(s) will be used with space padding on the left hand end.

```
Thus: LD A,'?'      Load a ? into A
      LD B,'ABC'    Load a C into B
      LD HL,'Z80'   Load an 80 into HL
      LD DE,'X'     Load sp X into DE
```

The only values which cannot be specified as a character immediate value are: Carriage Return (215 octal); Control-L (214 octal); Delete (377 octal or 233 octal); and the single quote (247 octal). Use one of the numeric forms for the values.

NUMERIC IMMEDIATE VALUES - There are five forms in which to enter numeric immediate values. Four of the five must be specified and the fifth (split-octal) is the default option. All numeric immediate values must begin with a numeric character, 0 through 9, appropriate to the type being specified (e.g., binary immediate values can only begin with a 0 or a 1). Any one of the four specified forms may be entered by entering the numeric string followed by a type indicator. The format of a numeric immediate value is:

numeric-string T

Where: numeric-string is a string of digits of value appropriate to the to the type of numeric value being specified

T is the type of indicator and has the following values and meanings:-

```
T=B  Binary
T=D  Decimal
T=H  Hexadecimal
T=Q  Pure Octal
T=C/R Split-Octal
```

All numeric immediate values are zero padded on the left hand end. Only as many significant digits as needed should be entered. There is no necessity to enter leading zeros except as required by the rule that all numeric immediate values must begin with a numeric character when entering a hexadecimal immediate value such that the first hex digit is not a 0 through 9.

Examples of the specification of numeric character values are shown below:

7FH	yields	7F
ABH	yields	error
0ABH	yields	AB
1110010B	yields	01110010
1777Q	yields	3/377 split
1777C/R	yields	1/377 split
6C/R	yields	006 split
58D	yields	072 split

Each digit of a numeric immediate value is checked for validity within its specified radix.

Whenever numeric immediate values are longer than necessary, only the appropriate number of the least significant digits are used as required by the command.

Examples of the use of immediate character values are as follows:

LD A,216	Load 216 octal into A
LD HL,0FF3FH	Load FF3F hex into HL
ADD 36D	ADD 032 octal to A
LD D,'-'	Load 255 octal into D

## ENTERING COMMANDS

Commands are entered into EDUCATOR-Z-80 as a string of characters (e.g., letters, numbers, spaces and punctuation) followed by a command-terminator. As written, EDUCATOR-Z-80 assumes that the command terminator will be a carriage-return (octal 215).

Since it is not uncommon to make errors when keying information into a computer, two provisions have been made in EDUCATOR-Z-80 for correcting or eliminating errors. The delete-character code (octal 377) is used to delete the last preceding existing character in the input string. Since a deleted character is not considered to exist, n consecutive character-deletes will delete the n consecutive characters preceding the n deletes. For example, RAX←L will be reduced to RAL and CQP←←MA will result in CMA.

Characters which have been keyed in are displayed after they have been tested. The display function used the octal value 377 as a clear-screen control code. Therefore, character-deletes are transformed into the back-arrow before they are displayed and stored. EDUCATOR-Z-80 users who have a back-arrow (octal 233) key on their keyboards may use it as a character-delete code and it will have the same effect as the delete key. The other---and somewhat more drastic---method of eliminating keying errors is to delete the entire input line.

This is usually done when an error is detected before the command-terminator character is input but several characters after the error occurred. The procedure for deleting an entire line is to enter a line-delete code (octal 214) which is an ASCII CONTROL-L. This will clear the input line and restart the command entry procedure.

A special feature of EDUCATOR-Z-80 permits the user to execute the last command which was input several times. This is accomplished by hitting "Return" again. In order to provide this facility, the input buffer is not cleared prior to calling for the entry of a new command and, since the the command-terminator is not stored in the buffer, the last previously entered command is still in the buffer. This feature is especially handy when demonstrating the effect of multiple executions of the Rotate, Increment, Decrement, Arithmetic and Logical commands.

The general format for entering a command is as follows:

OPCODE sp OPERAND-1 ,OPERAND-2 t

Where: OPCODE is the mnemonic opcode for the command.  
(Example: LD,SGC,etc.)

sp is one or more spaces.

OPERAND-1 is the first or only operand required by an instruction. It may be a register identification or an immediate value. See the implemented instruction set for the operand requirements.

OPERAND-2 is the second operand where required by a specific instruction. See the implemented instruction set.

t is the command-terminator, usually a carriage-return.

See ENTERING COMMANDS

The brackets shown in the general format are used to indicate that the items within them are optional since some commands do not require any operands (e.g., RLA, SCF, CPL, etc.), some require one operand only (e.g., AND, CP, DEC, etc.), and some commands require two operands (e.g., LD, EX).

## ERROR MESSAGES

There are a number of errors which can occur during the process of entering and executing commands under EDUCATOR-Z-80. When this happens an error message is displayed on the output device. A delay of approximately two seconds occurs as the message is being displayed to provide time to read it for those with TV-Monitors. The normal EDUCATOR-Z-80 display is outputted and the command entry mode is re-entered after the two second delay.

The errors which can occur are as follows:

<u>ERROR MESSAGE</u>	<u>MEANING</u>
INVALID COMMAND	The input command mnemonic is not one of the mnemonics permitted and/or implemented under EDUCATOR-Z-80 or the command is too long.
INVALID FORMAT	The format of the input command operand is inconsistent with the command.
INVALID IMMEDIATE VALUE	One of the characters in the immediate operand string was inconsistent with the immediate type code. For example, a digit in a binary input string was not a zero or a one.
INVALID SINGLE-REGISTER	The character specified where a single-register specification is required is not A, B, C, D, E, H or L.
INVALID DOUBLE-REGISTER	The character set specified where a double-register pair specification is required is not BC, DE, HL or, for the EX command, AF or AF <sup>1</sup> .