Z-80 ASSEMBLER ©

Assembler? What's that?

The Digital Group Z-80 Assembler is a collection of various software routines to increase an individual's software productivity. Not too clear yet? Well, this assembler gives a person the chance to build programs which are very close to the Z-80 machine language, yet isolates him from many of the problems, such as remembering a bunch of numbers, keeping track of each memory address, and often a total rewrite when a missing piece of programming must be inserted in the middle of the old program.

How? An assembler can work symbolicly. It's sort of like listening to somebody talking about "watchamacallits". The speaker may not quite grasp the proper name and precise concepts, but the general idea gets conveyed nonetheless. The programmer uses an assembler to avoid an exact address placement of each instruction; he usually locates just the beginning address; everything else just naturally following. Instead of programming using:

address	data
123345	076
123346	321
123347	323
123350	002

he could use:

```
LOADA  LD  A,321
       OUT 002
```

The assembler replaces the addresses with a label, which can be somewhat descriptive, and replaces the numeric data codes with the much more easily remembered Z-80 mnemonic.

Some sacrifices result, however. If the machine code version was directly entered into the machine, only 4 bytes of storage would be used. In the Assembler version, the assembler code itself takes up many K of storage, the "Source code" (note the term) would take up 20-30 bytes, and the resultant "Object code" (note this term-it's the resultant machine language) uses 4 bytes. One would suspect that assemblers were invented by memory manufacturers.

In addition, a certain amount of additional time is required for the assembler to be loaded, more characters to be typed and the actual assembly (or conversion to machine code) to take place. Obviously, direct machine code will often be a better choice for small programs, and mandatory for small systems. The assembler becomes very attractive for programs greater than a few hundred bytes, and the total time and effort savings in a commercial operation far outweighs the additional memory costs.

System required to run this assembler

- Digital Group Z-80 system with 18K or more memory.
- Digital Group Audio Cassette interface (standard with most systems).
- Digital Group 512 character TV Readout board.
- Standard ASCII keyboard attached to Port Ø. Upper and lower case character set, control codes, and cursor keys helpful, but not required.

Since the programming used in this Z-80 Assembler is standard Z-80 code, this assembler can be modified to run on systems which differ considerably from the intended design, or even run on non D.G. systems (shame on you!).

General Assembler Design

The Digital Group Z-80 Assembler programming occupies from address 0 to split octal address 057377 (2FFF in Hex). The programmer's Source and Object code may be placed at any address above the assembler (060000 Octal or 3000 Hex). The exact memory utilization is:

000000	- 000377	EROM
001000	- 005377	General Z-80 Op Sys
006000	- 006377	Optional Hardcopy area
007000	- 101377	TV Scroll buffer
011000	- 011377	TV & Keyboard Scrolling subroutine
021000	- 057377	Assembler Routines
060000	- 377377	User Area for Source & Object files

Several versions of page 6 have already been written for various hardcopy devices. The 256 bytes in this area should be adequate for most hardcopy devices.

General Assembler Aids to the User

The Digital Group Z-80 Assembler is designed to assemble from standard Z-80 mnemonics as listed in the Zilog User's Guide manual shipped with each Digital Group Z-80 system. This assembler uses an extensive line oriented editor which automatically generates line numbers and automatically tabs to the proper field. Other features of the editor are:

- File management
- Source code Write on cassette
- Source code Read from cassette
- Resequence line numbers
- File listing (all lines, single line, block of lines)
- Line deletion (single line, or block of lines)
- Linking ability between files
- Error messages
- Octal, Hexadecimal, Decimal, and ASCII constants
- Relative addressing
- Source code reduction

Using the Assembler

1. Place the cassette labeled "Z-80 Assembler" in the cassette recorder. When the low tone begins, turn on your 18K or greater Digital Group Z-80 system.
2. When the data burst ends, the system should display an option select list. A modified version of the Z-80 Op System is utilized for options 3 and 4 (see Appendix A).
3. Option 7 begins the assembler operation. If hardcopy is desired, Option 8 may be selected, but an initial selection of hardcopy generally wastes paper and slows the operation. Option 8 is recommended for source code listing of large programs and assembly listings. So, press 7 on your keyboard.
4. The screen will initially display "Z-80 Assembler" at the top of the screen, and an underline character (_) 4 lines below the title message.

5. Precise entries to the assembler are required to obtain proper results. The following abbreviations will be referred to in the succeeding directions:
   (sp) means a space bar depression
   (cr) means a carriage return key depression
   (x^C) means a letter X entry while the "Control" key is depressed (deletes the line of characters currently being entered)
   ($@^C$) means a letter @ entry while the "Control" key is depressed (exits from the auto numbering mode)
   (!) begins the auto numbering mode (with a line number of 0100)
   (DEL) means a delete key depression (backspaces to delete the previous character just entered - used for error correction)
   The scrolling system's input routine automatically modifies any lower case alpha entry to upper case.
6. The first operation must be the creating of a source code file. This is done by typing FILE(sp) a filename consisting of a arbitrary name 1-5 characters in length, the first character of which is alpha (sp) and the starting location address which must be greater than or equal to 060000(octal) or 3000(hex)(cr). Example:
   FILE(sp)TEST(sp)060000(cr)
7. The system will respond with the filename that you have selected, then the current beginning and ending address, which at this point would be the same. Example: TEST 060000 060000
8. The system is now ready to assist you in building your source file. To begin the auto numbering system type !
9. The system will respond with 0100(sp) you may now build your program. The system has been designed so that five fields are utilized: <u>number</u> <u>label</u> <u>mnemonic</u> <u>operation</u> <u>comment</u>
   a. The "number" field contains a four digit number between 0001 and 9999. The auto numbering system will increment by ten Note: the number entry must contain four digits.
   b. The "label" field is optional. If desired, an entry, consisting of 1-6 alphanumeric characters, the first of which is alpha, may be placed in this field. A comment statement, beginning with an * may begin in this field.
   c. The "mnemonic" field must be one of the various Z-80 Op codes shown in the Zilog users' manual or the listing included with this document.
   d. The "operation" field is required for most, but not all, op codes. The included listing shows samples. Symbolic labels may be utilized if desired.
   e. A "comment" may be included if desired. A short, meaningful comment can be a great help for later analysis of the function of each instruction.
   f. A (cr) terminates the line entry, and the system will automatically enter the next line entry. If no further entry is desired, an $@^C$ will exit from the autonumbering system.
10. After the source file has been made, several options are open to the programmer prior to assembling the source file.
   LIST(cr) will result in a listing of the source file you have been building.
   FILE(cr) will produce the name of the current file, as well as the beginning then ending address.
   RSEQ(cr) will reorder the source files numbering to start at 0100 and increment in 0010 steps.

digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

DELT(sp)XXXX
or
DELT(sp)XXXX(sp)XXXX(cr) will result in the line (first example) or a block of lines being deleted.

XXXX inserts or corrects a line of code. XXXX may be any appropriate number 0001 and 9999 inclusive.

SAVE(cr)will produce a cassette recording of the current source file. Of course, a fresh tape should be placed in the cassette recorder, and the recorder placed in record mode prior to entering the (cr). The appropriate headers will be written on the tape for subsequent loading.

11. The assembly of the source code can take eight forms.

ASSM(sp)HHHH(cr) (hex version)
or
ASSM(sp)000000(cr) (octal version)
This first form will assemble the source file starting at the indicated address. Be careful of the address. You can wind up destroying the assembler, the source code, or run out of memory.

ASSM(sp)hhhh(sp)HHHH(cr) (hex version)
or
ASSM(sp)oooooo(sp)000000(cr) (octal version)
This form will assemble the source code such that the object code is located starting at the second address given. However the actual code generated is the code as it would appear if designed to run at the first address given. Example:
ASSM(sp)006000(sp)130000(cr) will result in a octal listing of the code as if it were designed to run at page 6 and following, but a look at storage will show that the code is physically located at page 130 and following.
This form is used to generate code which will eventually be run at the area of storage where the assembler or source code is presently located.

ASSME(sp)HHHH(cr) (hex version)
or
ASSME(sp)000000 (octal version)
This form is similar to the ASSM(cr) format, except that only errors are listed.

ASSME(sp)hhhh(sp)HHHH(cr) (hex version)
or
ASSME(sp)oooooo(sp)000000(cr) (octal version)
This form will also list only errors, but will perform the dual address system above.

ASSML(sp)HHHH(cr)
ASSML(sp)000000(cr)

ASSML(sp)hhhh(sp)HHHH(cr)
ASSML(sp)000000(sp)000000(cr)

ASSMLE(sp)HHHH(cr)
ASSMLE(sp)000000(cr)

digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

ASSMLE(sp)hhhh(sp)HHHH(cr)
ASSMLE(sp)oooooo(sp)OOOOOO(cr)
These last four forms of the ASSM permit linking the assembly of a number of Source files through a common label table. This permits utilizing the same limited amount of memory for each Source file. The cassette recorder is utilized to SAVE and LOAD the various Source files. The ASSML(E) instruction then builds the desired Object code.

For example, suppose a user wishes to assemble two Source files, named TEST1 and TEST2, each requiring 4K of storage. The user has an 18K system, so making them into a single 8K Source file (plus the 12K of the Assembler code) would exceed the storage capacity of the system. TEST1 references labels in TEST2 and viceversa. The user would assemble TEST1 using the usual ASSM(E) instruction. "L" (label) errors will result, but are temporarily ignored. The TEST1 Source file is then SAVEed on a cassette. TEST2 is next made current and entered or LOADED. This time the user assembles TEST2 using the ASSML(E) instruction. There should be no "L" errors on this run if this is the last Source file. TEST1 is then made current and reLOADed. Re-assemble TEST1 using ASSML(E). This time every label of TEST1 should be listed as a "D" (duplicate) label, because all these labels are already in the label table. The assembly is now complete. Up to six Source files can be merged in this manner. The only limitation is the amount of RAM available and the maximum of 96 labels (unless the label area is enlarged).

Here are two samples of how the assembler works. The underlined characters are the operator's entries.

The first example demonstrates the ASSM(E) format.

```
FILE TEST 060000
TEST  060000  060000

! 0100 KEY    IN   000    GET DATA FROM PORT 0
0110 _       BIT  7,A
0120 _       JR   Z,KEY
0130 _       PUSH AF     SAVE DATA
0140 STROBE  IN   000
0150 _       BIT  7,A
0160 _       JR   Z,KEY
0170 _       POP  AF     RESTORE DATA
0180 _       RET  _

0090 * KEYBOARD INPUT SUBROUTINE
0160 _       JR   NZ,KEY
```


digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

```
RSEQ

LIST
0100 * KEYBOARD INPUT SUBROUTINE
0110 KEY    IN   000    GET DATA FROM PORT 0
0120        BIT  7,A
0130        JR   Z,KEY
0140        PUSH AF     SAVE DATA
0150 STROBE IN   000
0160        BIT  7,A
0170        JR   NZ,KEY
0180        POP  AF     RESTORE DATA
0190        RET

ASSM 001250 130000

001250                      0100 * KEYBOARD INPUT SUBROUTINE
001250 333 000              0110 KEY    IN   000    GET DATA FROM PORT 0
001252 313 177              0120        BIT  7,A
001254 050 372              0130        JR   Z,KEY
001256 365                  0140        PUSH AF     SAVE DATA
001257 333 000              0150 STROBE IN   000
001261 313 177              0160        BIT  7,A
001263 040 363              0170        JR   NZ,KEY
001265 361                  0180        POP  AF     RESTORE DATA
001266 311                  0190        RET

ASSM 4000

4000                 0100 * KEYBOARD INPUT SUBROUTINE
4000 DB 00           0110 KEY    IN   000    GET DATA FROM PORT 0
4002 CB 7F           0120        BIT  7,A
4004 28 FA           0130        JR   Z,KEY
4006 F5              0140        PUSH AF     SAVE DATA
4007 DB 00           0150 STROBE IN   000
4009 CB 7F           0160        BIT  7,A
400B 20 F3           0170        JR   NZ,KEY
400D F1              0180        POP  AF     RESTORE DATA
400E C9              0190        RET
```

The Second example shows the ASSML(E) operation.

```
FILE TEST2 060000
TEST2 060000  060000

! 0100 * TV OUTPUT SUBROUTINE
0110 SPACE  LD   A,' '
0120 TV     OUT  000
0130        XOR  A
0140        OUT  000
0150        RET
```


digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

```
ASSM  000370 100000

000370                          0100 * TV OUTPUT SUBROUTINE
000370 076 240                  0110 SPACE  LD   A,' '
000372 323 000                  0120 TV     OUT  000
000374 257                      0130        XOR  A
000375 323 000                  0140        OUT  000
000377 311                      0150        RET

SAVE

FILE TEST3 060000
TEST3 060000  060000

! 0100 * TV ERASE SUBROUTINE
0110 *
0120 ERASE  LD   A,127D
0130        CALL TV
0140        LD   B,000
0150 CLOAD  D LD   C,004
0160        CALL SPACE
0170        DEC  C
0180        JR   NZ,BLANK
0190        DJNZ CLOAD
0200        RET

0150 CLOAD  LD   C,004
0160 BLANK  CALL SPACE

LIST
0100 * TV ERASE SUBROUTINE
0110 *
0120 ERASE  LD   A,127D
0130        CALL TV
0140        LD   B,000
0150 CLOAD  LD   C,004
0160 BLANK  CALL SPACE
0170        DEC  C
0180        JR   NZ,BLANK
0190        DJNZ CLOAD
0200        RET

ASSML 000346 077356

000346                          0100 * TV ERASE SUBROUTINE
000346                          0110 *
000346 076 177                  0120 ERASE  LD   A,127D
000350 315 372 000              0130        CALL TV
000353 006 000                  0140        LD   B,000
000355 016 004                  0150 CLOAD  LD   C,004
000357 315 370 000              0160 BLANK  CALL SPACE
000362 015                      0170        DEC  C
000363 040 372                  0180        JR   NZ,BLANK
000365 020 366                  0190        DJNZ CLOAD
000367 311                      0200        RET
```


digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

Other Operations

Several other features of the Z-80 Assembler have been included in this initial release.

LOAD(cr) was used in the above operations to save the current source file on a cassette for later use.

EXEC(sp)HHHH(cr) (hex version)
EXEC(sp)OOOOOO(cr) (octal version)
will cause the assembler to branch to the indicated address. This address is normally the beginning address of the Object code generated by the assembly operation.

SYMB(cr) will print the label table data as shown above.

NEWF(cr) clears all RAM Source areas and label table areas. While not generally necessary, this operation is recommended when different successive assemblies (but not linked) are being run.

Arguments

Arguements consist of Register names, conditions (for Call, JR,JP, and RET), constants, and variables with or without offset.

Register names: A,B,C,D,E,H,L,AF,BC,DE,HL,SP,IX,IY. On a LD operation M may be used instead of (HL).

Conditions: Z,NZ,C,NC,PO,PE,P,M.

Constants: Octal Default e.g. 377 in Octal
Decimal indicator is D. e.g. 128 D means a decimal value of 128.
Hex indicator is H. e.g. 1EH means a Hex value of 1E. NOTE: all hex number must begin with a number e.g. ØFFH means a hex value of FF.
Ascii constants are indicated by single quotes e.g. 'B' means the Ascii value of B which would be 302 in Octal. NOTE: only one character permitted, and (cr) and control characters are not permitted.

Variables: These are the labels defined in the Source program or by a EQU statement. Those in a constant field can be offset by an Octal, Hex or Decimal constant. E.G. HERE+17 THERE-20H. The variables (labels) may be 1 to 6 characters in length. The first character must be alphabetic. The other characters may be alphabetic or numeric. Special characters are not allowed, nor may the register names be used for labels.

$ Is used to indicate the 16 bit value of the program counter after the current instruction is assembled. Offset values may be added to this pointer. Examples:
0080(sp)BEGIN(sp)JP(sp)$+100(cr)
0090(sp)LINE(sp)DS(sp)$(cr)

Pseudo Ops

ORG will reset the origin of the assembly at that point to the value indicated. Examples: 0100(sp)(sp)ORG(sp)4000(cr) (hex org)
0100(sp)(sp)ORG(sp)100123(cr) (octal org)

EQU gives a value to a symbol. Example:
0120(sp)THREE(sp)EQU(sp)003(cr)
0130(sp)ERASE(sp)EQU(sp)000346(cr)

DS leaves the indicated number bytes unchanged. Example:
0140(sp)BUFFER(sp)DS(sp)512D(cr)

DB defines the value of one byte of data. Example:
0150(sp)ROBERT(sp)DB(sp)123(cr)

DW defines a number of data words. One form has ASCII data words placed between single quotes. (cr) and control characters are not permitted. Example:
0160(sp)SUDING(sp)DW(sp)'SUDING'

Another form has a number or a label without the single quote marks. A sixteen bit number is stored corresponding to the address of the label or the value of the number. Examples:
0170(sp)Here(sp)DW(sp)123321(cr)
0180(sp)Where(sp)DW(sp)SUDING

Errors

The following errors are listed at assembly time in the space between the Object code and the Source code.

D - Duplicate label detected
M - Missing label
O - Opcode illegal
L - Label illegal or missing
V - Value of constant too large. Value of a relative jump too large.
U - Undefined Symbol
S - Syntax illegal
R - Register illegal
A - Argument illegal
N - Bit selected illegal (SET,RES,BIT)
J - Jump relative error

Some messages will occur at command times such as:

????? - The command given was illegal
USED - The file name already exists
FULL - All six files are already being used
LABEL TABLE FULL - All 96 labels permitted have been used.

Special File Handling

The command FILE can create a new Source file, delete an old Source file, or make an old Source file current for further work or available for cassette operations.

FILE(cr) will give the name of the current Source file with its beginning and ending addresses

FILES(cr) will list all files and their addresses

FILE(sp)XXXX(sp)(cr) will make the Source file name entered for XXXX current

FILE(sp)XXXX(sp)Ø(cr) will remove the Source file name entered for XXXX from the library.

FILE(sp)XXXX(sp)HHHH(cr) (hex version)
or
FILE(sp)XXXX(sp)OOOOOO(cr) (octal version)
will create a new Source file as previously described.

digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

Saving your Object file on Cassette

Often the assembler will be used to generate object code designed to be immediately run with the complete assembler and source file resident. In this case, no special object code saving is necessary. However, the code will generally be designed to run in the storage area occupied by the assembler, hence, temporary cassette loading may be desireable. Similarly, large programs may be constructed from smaller object code modules. The main operating system can be used as the new system's operating system, or, the assembler can be used to generate "a new world" of operating system.

Either way, the programmer would generally temporarily put the new data on a cassette(s) and later build upon the basic cassette reading system in the EROM area (000000 to 000377). This procedure consists of modifying the existing cassette writting routine to record only the special area just assembled, then modifying the cassette reading routine to place the new software where desired.

EXAMPLE: The Assembler assembles some code to run at page 012. Since this is the present assembler area, the assembly would be the double address type such as ASSM 012000 - 100000. The resultant code of this double address would be located at address 100000 and following. The write routine would be modified to load this area of memory on a cassette. Assuming the end to be at 101321, the start and stop addresses of 100000 and 101321 would then be loaded as replacements to the present write addresses on page 001.

```
Modify: 001044 to 000  (L Start)
        001047 to 100  (H Start)
        001052 to 321  (L End)
        001055 to 101  (H End)
```

Press R to obtain the Op System display. After placing a fresh cassette on the recorder and pressing "Record" select Option "2" and write the cassette. Stop the cassette when finished writing.

The desired op system may then be read in or the present op system may be utilized if the scrolling system is desired. The cassette read constants must be next modified to read in the special cassette to the desired location. Since in our example the code was designed to run at 011000 to 012321, these start and stop addresses must be entered.

```
Modify: 001030 to 000  (L Start)
        001031 to 011  (H Start)
        001032 to 321  (L End)
        001033 to 012  (H End)
```

Press R to obtain the Op System display. After placing the special cassette in the recorder and starting to play the cassette, select option "1" when the low tone begins. After the tone burst, the new data is loaded.

Restore the write constants at 001044 - 001055 to their original values, and the transfer is complete.

Another possible way to transfer the data is to write and execute a Z-80 memory transfer routine such as:

```
LD   HL,From   *(From = present location)
LD   DE,TO     *(TO=intended location)
DD   BC,Length* Set to code length
LDIR           *Transfer
RST  060       *Do a storage Dump
```


digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

```
ASSM 130000

130000                      0100 * Z-80 ASSM CODE TESTER
130000                      0110 *
130000                      0120 * DATA ASSIGNING
130000                      0130 *
130000                      0140 TEST   EQU  123345
130000                      0150 HEX    EQU  1234H
130000                      0160 DECMAL EQU  123D
130000                      0170 THREE  EQU  003
130000 377                  0180 DATA   DB   377
130001                      0190 EDIT   DW   'EDIT'
 305 304 311 324
130005                      0200 STORE  DS   001
130006                      0210 *
130006                      0220 *8 BIT LOAD GROUP
130006                      0230 *
130006 355 127              0240        LD   A,I
130010 355 137              0250        LD   A,R
130012 177                  0260        LD   A,A
130013 170                  0270        LD   A,B
130014 171                  0280        LD   A,C
130015 172                  0290        LD   A,D
130016 173                  0300        LD   A,E
130017 174                  0310        LD   A,H
130020 175                  0320        LD   A,L
130021 176                  0330        LD   A,(HL)
130022                      0340 * "M" MAY BE USED IN PLACE OF "(HL)" IF DESIRED:
130022 176                  0350        LD   A,M
130023 012                  0360        LD   A,(BC)
130024 032                  0370        LD   A,(DE)
130025 335 176 002          0380        LD   A,(IX+2)
130030 375 176 003          0390        LD   A,(IY+THREE)
130033 072 345 123          0400        LD   A,(TEST)
130036 076 003              0410        LD   A,THREE
130040 107                  0420        LD   B,A
130041 100                  0430        LD   B,B
130042 101                  0440        LD   B,C
130043 102                  0450        LD   B,D
130044 103                  0460        LD   B,E
130045 104                  0470        LD   B,H
130046 105                  0480        LD   B,L
```


digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

```
130350 335 041 345 123      1550        LD   IX,TEST
130354 335 052 345 123      1560        LD   IX,(TEST)
130360 335 341              1570        POP  IX
130362 375 041 345 123      1580        LD   IY,TEST
130366 375 052 345 123      1590        LD   IY,(TEST)
130372 375 341              1600        POP  IY
130374 355 103 345 123      1610        LD   (TEST),BC
131000 355 123 345 123      1620        LD   (TEST),DE
131004 042 345 123          1630        LD   (TEST),HL
131007 355 163 345 123      1640        LD   (TEST),SP
131013 335 042 345 123      1650        LD   (TEST),IX
131017 375 042 345 123      1660        LD   (TEST),IY
131023 365                  1670        PUSH AF
131024 305                  1680        PUSH BC
131025 325                  1690        PUSH DE
131026 345                  1700        PUSH HL
131027 335 345              1710        PUSH IX
131031 375 345              1720        PUSH IY
131033                      1730 * END OF 16 BIT LOAD
131033                      1740 *
131033                      1750 * END OF LOAD TEST
131033                      1760 * EXCHANGES
131033                      1770 *
131033 010                  1780        EX   AF
131034 331                  1790        EXX
131035 353                  1800        EX   DE,HL
131036 343                  1810        EX   (SP),HL
131037 335 343              1820        EX   (SP),IX
131041 375 343              1830        EX   (SP),IY
131043                      1840 *
131043                      1850 * BLOCK TRANSFER
131043                      1860 *
131043 355 240              1870        LDI
131045 355 260              1880        LDIR
131047 355 250              1890        LDD
131051 355 270              1900        LDDR
131053                      1910 *
131053                      1920 * BLOCK SEARCH
131053                      1930 *
131053 355 241              1940        CPI
131055 355 261              1950        CPIR
131057 355 251              1960        CPD
131061 355 271              1970        CPDR
131063                      1980 *
131063                      1990 * 8 BIT ARITHMETIC & LOGIC
131063 207                  2000        ADD  A
131064 200                  2010        ADD  B
131065 201                  2020        ADD  C
131066 202                  2030        ADD  D
131067 203                  2040        ADD  E
131070 204                  2050        ADD  H
131071 205                  2060        ADD  L
131072 206                  2070        ADD  (HL)
```

```
130165 154                1020        LD   L,H
130166 155                1030        LD   L,L
130167 156                1040        LD   L,(HL)
130170 335 156 002        1050        LD   L,(IX+2)
130173 375 156 003        1060        LD   L,(IY+THREE)
130176 056 003            1070        LD   L,3
130200 167                1080        LD   (HL),A
130201 160                1090        LD   (HL),B
130202 161                1100        LD   (HL),C
130203 162                1110        LD   (HL),D
130204 163                1120        LD   (HL),E
130205 164                1130        LD   (HL),H
130206 165                1140        LD   (HL),L
130207 066 003            1150        LD   (HL),THREE
130211 002                1160        LD   (BC),A
130212 002                1170        LD   (BC),A
130213 335 167 002        1180        LD   (IX+2),A
130216 335 160 002        1190        LD   (IX+2),B
130221 335 161 002        1200        LD   (IX+2),C
130224 335 162 002        1210        LD   (IX+2),D
130227 335 163 002        1220        LD   (IX+2),E
130232 335 164 002        1230        LD   (IX+2),H
130235 335 165 002        1240        LD   (IX+2),L
130240 335 066 002 003    1250        LD   (IX+2),THREE
130244 375 167 003        1260        LD   (IY+3),A
130247 375 160 003        1270        LD   (IY+3),B
130252 375 161 003        1280        LD   (IY+3),C
130255 375 162 003        1290        LD   (IY+3),D
130260 375 163 003        1300        LD   (IY+3),E
130263 375 164 003        1310        LD   (IY+3),H
130266 375 165 003        1320        LD   (IY+3),L
130271 375 066 003 003    1330        LD   (IY+3),THREE
130275 062 345 123        1340        LD   (TEST),A
130300 355 107            1350        LD   I,A
130302 355 117            1360        LD   R,A
130304                    1370 * END 8 BIT LOAD
130304                    1380 *
130304                    1390 * BEGIN 16 BIT LOAD
130304 361                1400        POP  AF
130305 001 345 123        1410        LD   BC,TEST
130310 355 113 345 123    1420        LD   BC,(TEST)
130314 301                1430        POP  BC
130315 021 345 123        1440        LD   DE,TEST
130320 355 133 345 123    1450        LD   DE,(TEST)
130324 321                1460        POP  DE
130325 041 345 123        1470        LD   HL,TEST
130330 052 345 123        1480        LD   HL,(TEST)
130333 341                1490        POP  HL
130334 371                1500        LD   SP,HL
130335 335 371            1510        LD   SP,IX
130337 375 371            1520        LD   SP,IY
130341 061 345 123        1530        LD   SP,TEST
130344 355 173 345 123    1540        LD   SP,(TEST)
```


digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

```
130047 106                0490      LD   B,(HL)
130050 335 106 002        0500      LD   B,(IX+2)
130053 375 106 003        0510      LD   B,(IY+THREE)
130056 006 003            0520      LD   B,THREE
130060 117                0530      LD   C,A
130061 110                0540      LD   C,B
130062 111                0550      LD   C,C
130063 112                0560      LD   C,D
130064 113                0570      LD   C,E
130065 114                0580      LD   C,H
130066 115                0590      LD   C,L
130067 116                0600      LD   C,(HL)
130070 335 116 002        0610      LD   C,(IX+2)
130073 375 116 003        0620      LD   C,(IY+THREE)
130076 016 003            0630      LD   C,THREE
130100 127                0640      LD   D,A
130101 120                0650      LD   D,B
130102 121                0660      LD   D,C
130103 122                0670      LD   D,D
130104 123                0680      LD   D,E
130105 124                0690      LD   D,H
130106 125                0700      LD   D,L
130107 126                0710      LD   D,(HL)
130110 335 126 002        0720      LD   D,(IX+2)
130113 375 126 003        0730      LD   D,(IY+3)
130116 026 003            0740      LD   D,THREE
130120 137                0750      LD   E,A
130121 130                0760      LD   E,B
130122 131                0770      LD   E,C
130123 132                0780      LD   E,D
130124 133                0790      LD   E,E
130125 134                0800      LD   E,H
130126 135                0810      LD   E,L
130127 136                0820      LD   E,(HL)
130130 335 136 002        0830      LD   E,(IX+2)
130133 375 136 003        0840      LD   E,(IY+3)
130136 036 003            0850      LD   E,3
130140 147                0860      LD   H,A
130141 140                0870      LD   H,B
130142 141                0880      LD   H,C
130143 142                0890      LD   H,D
130144 143                0900      LD   H,E
130145 144                0910      LD   H,H
130146 145                0920      LD   H,L
130147 146                0930      LD   H,(HL)
130150 335 146 002        0940      LD   H,(IX+2)
130153 375 146 003        0950      LD   H,(IY+THREE)
130156 046 003            0960      LD   H,THREE
130160 157                0970      LD   L,A
130161 150                0980      LD   L,B
130162 151                0990      LD   L,C
130163 152                1000      LD   L,D
130164 153                1010      LD   L,E
```


digital group software systems, inc. p.o. box 1086 arvada, colorado 80001

```
131073 335 206 002
131076 375 206 003
131101 306 003
131103 217
131104 210
131105 211
131106 212
131107 213
131110 214
131111 215
131112 216
131113 335 216 002
131116 375 216 003
131121 316 003
131123 227
131124 220
131125 221
131126 222
131127 223
131130 224
131131 225
131132 226
131133 335 226 002
131136 375 226 003
131141 326 003
131143 237
131144 230
131145 231
131146 232
131147 233
131150 234
131151 235
131152 236
131153 335 236 002
131156 375 236 003
131161 336 003
131163 247
131164 240
131165 241
131166 242
131167 243
131170 244
131171 245
131172 246
131173 335 246 002
131176 375 246 003
131201 346 003
131203 257
131204 250
131205 251
131206 252
131207 253
131210 254
131211 255
131212 256
```

```
2080    ADD  (IX+2)
2090    ADD  (IY+THREE)
2100    ADD  THREE
2110    ADC  A
2120    ADC  B
2130    ADC  C
2140    ADC  D
2150    ADC  E
2160    ADC  H
2170    ADC  L
2180    ADC  (HL)
2190    ADC  (IX+2)
2200    ADC  (IY+THREE)
2210    ADC  THREE
2220    SUB  A
2230    SUB  B
2240    SUB  C
2250    SUB  D
2260    SUB  E
2270    SUB  H
2280    SUB  L
2290    SUB  (HL)
2300    SUB  (IX+2)
2310    SUB  (IY+THREE)
2320    SUB  THREE
2330    SBC  A
2340    SBC  B
2350    SBC  C
2360    SBC  D
2370    SBC  E
2380    SBC  H
2390    SBC  L
2400    SBC  (HL)
2410    SBC  (IX+2)
2420    SBC  (IY+3)
2430    SBC  THREE
2440    AND  A
2450    AND  B
2460    AND  C
2470    AND  D
2480    AND  E
2490    AND  H
2500    AND  L
2510    AND  (HL)
2520    AND  (IX+2)
2530    AND  (IY+3)
2540    AND  3
2550    XOR  A
2560    XOR  B
2570    XOR  C
2580    XOR  D
2590    XOR  E
2600    XOR  H
2610    XOR  L
2620    XOR  (HL)
```


digital group software systems, inc. p.o.box 1086 arvada, colorado 80001

```
131213 335 256 002          2630        XOR  (IX+2)
131216 375 256 003          2640        XOR  (IY+THREE)
131221 356 003              2650        XOR  THREE
131223 267                  2660        OR   A
131224 260                  2670        OR   B
131225 261                  2680        OR   C
131226 262                  2690        OR   D
131227 263                  2700        OR   E
131230 264                  2710        OR   H
131231 265                  2720        OR   L
131232 266                  2730        OR   (HL)
131233 335 266 002          2740        OR   (IX+2)
131236 375 266 003          2750        OR   (IY+THREE)
131241 366 003              2760        OR   THREE
131243 277                  2770        CP   A
131244 270                  2780        CP   B
131245 271                  2790        CP   C
131246 272                  2800        CP   D
131247 273                  2810        CP   E
131250 274                  2820        CP   H
131251 275                  2830        CP   L
131252 276                  2840        CP   (HL)
131253 335 276 002          2850        CP   (IX+2)
131256 375 276 003          2860        CP   (IY+THREE)
131261 376 003              2870        CP   THREE
131263 074                  2880        INC  A
131264 004                  2890        INC  B
131265 014                  2900        INC  C
131266 024                  2910        INC  D
131267 034                  2920        INC  E
131270 044                  2930        INC  H
131271 054                  2940        INC  L
131272 064                  2950        INC  (HL)
131273 335 064 002          2960        INC  (IX+2)
131276 375 064 003          2970        INC  (IY+THREE)
131301 075                  2980        DEC  A
131302 005                  2990        DEC  B
131303 015                  3000        DEC  C
131304 025                  3010        DEC  D
131305 035                  3020        DEC  E
131306 045                  3030        DEC  H
131307 055                  3040        DEC  L
131310 065                  3050        DEC  (HL)
131311 335 065 002          3060        DEC  (IX+2)
131314 375 065 003          3070        DEC  (IY+THREE)
131317                      3080 *
131317                      3090 * 16 BIT ARITHMETIC
131317                      3100 *
131317 011                  3110        ADD  HL,BC
131320 031                  3120        ADD  HL,DE
131321 051                  3130        ADD  HL,HL
131322 071                  3140        ADD  HL,SP
131323 335 011              3150        ADD  IX,BC
```

```
131325 335 031          3160        ADD  IX,DE
131327 335 071          3170        ADD  IX,SP
131331 335 051          3180        ADD  IX,IX
131333 375 011          3190        ADD  IY,BC
131335 375 031          3200        ADD  IY,DE
131337 375 071          3210        ADD  IY,SP
131341 375 051          3220        ADD  IY,IY
131343 355 112          3230        ADC  HL,BC
131345 355 132          3240        ADC  HL,DE
131347 355 152          3250        ADC  HL,HL
131351 355 172          3260        ADC  HL,SP
131353 355 102          3270        SBC  HL,BC
131355 355 122          3280        SBC  HL,DE
131357 355 142          3290        SBC  HL,HL
131361 355 162          3300        SBC  HL,SP
131363 003              3310        INC  BC
131364 023              3320        INC  DE
131365 043              3330        INC  HL
131366 063              3340        INC  SP
131367 335 043          3350        INC  IX
131371 375 043          3360        INC  IY
131373 013              3370        DEC  BC
131374 033              3380        DEC  DE
131375 053              3390        DEC  HL
131376 073              3400        DEC  SP
131377 335 053          3410        DEC  IX
132001 375 053          3420        DEC  IY
132003                  3430 *
132003                  3440 * GENERAL PURPOSE OPS
132003                  3450 *
132003 047              3460        DAA
132004 057              3470        CPL
132005 355 104          3480        NEG
132007 077              3490        CCF
132010 067              3500        SCF
132011                  3510 *
132011                  3520 * JUMP, CALL, & RETURN
132011                  3530 *
132011 303 345 123      3540        JP   TEST
132014 332 345 123      3550        JP   C,TEST
132017 322 345 123      3560        JP   NC,TEST
132022 312 345 123      3570        JP   Z,TEST
132025 302 345 123      3580        JP   NZ,TEST
132030 352 345 123      3590        JP   PE,TEST
132033 342 345 123      3600        JP   PO,TEST
132036 372 345 123      3610        JP   M,TEST
132041 362 345 123      3620        JP   P,TEST
132044 030 376          3630 HERE   JR   HERE
132046 070 374          3640        JR   C,HERE
132050 060 372          3650        JR   NC,HERE
132052 050 370          3660        JR   Z,HERE
132054 040 366          3670        JR   NZ,HERE
132056 020 364          3680        DJNZ HERE
```

```
132060 351                  3690        JP   (HL)
132061 335 351              3700        JP   (IX)
132063 375 351              3710        JP   (IY)
132065 315 345 123          3720        CALL TEST
132070 334 345 123          3730        CALL C,TEST
132073 324 345 123          3740        CALL NC,TEST
132076 314 345 123          3750        CALL Z,TEST
132101 304 345 123          3760        CALL NZ,TEST
132104 344 345 123          3770        CALL PO,TEST
132107 354 345 123          3780        CALL PE,TEST
132112 374 345 123          3790        CALL M,TEST
132115 364 345 123          3800        CALL P,TEST
132120 311                  3810        RET
132121 330                  3820        RET  C
132122 320                  3830        RET  NC
132123 310                  3840        RET  Z
132124 300                  3850        RET  NZ
132125 340                  3860        RET  PO
132126 350                  3870        RET  PE
132127 370                  3880        RET  M
132130 360                  3890        RET  P
132131 355 115              3900        RETI
132133 355 105              3910        RETN
132135                      3920 *
132135                      3930 * RESTART
132135                      3940 *
132135 307                  3950        RST  0
132136 317                  3960        RST  8D
132137 327                  3970        RST  16D
132140 337                  3980        RST  24D
132141 347                  3990        RST  32D
132142 357                  4000        RST  40D
132143 367                  4010        RST  48D
132144 377                  4020        RST  56D
132145                      4030 *
132145                      4040 * INPUT
132145                      4050 *
132145 333 003              4060        IN   THREE
132147                      4070 * ALTERNATE INPUT (A) FORMAT:
132147 333 003              4080        IN   A,THREE
132151 355 170              4090        IN   A,(C)
132153 355 100              4100        IN   B,(C)
132155 355 110              4110        IN   C,(C)
132157 355 120              4120        IN   D,(C)
132161 355 130              4130        IN   E,(C)
132163 355 140              4140        IN   H,(C)
132165 355 150              4150        IN   L,(C)
132167 355 160              4160        IN   F,(C)
132171                      4170 * BLOCK INPUT
132171 355 242              4180        INI
132173 355 262              4190        INIR
132175 355 252              4200        IND
132177 355 272              4210        INDR
```

```
132201                          4220 *
132201                          4230 * OUTPUT
132201                          4240 *
132201 323 003                  4250        OUT  THREE
132203                          4260 * ALTERNATE OUTPUT (A) FORMAT:
132203 323 003                  4270        OUT  A,THREE
132205 355 171                  4280        OUT  (C),A
132207 355 101                  4290        OUT  (C),B
132211 355 111                  4300        OUT  (C),C
132213 355 121                  4310        OUT  (C),D
132215 355 131                  4320        OUT  (C),E
132217 355 141                  4330        OUT  (C),H
132221 355 151                  4340        OUT  (C),L
132223                          4350 * BLOCK OUTPUT
132223 355 243                  4360        OUTI
132225 355 263                  4370        OTIR
132227 355 253                  4380        OUTD
132231 355 273                  4390        OTDR
132233                          4400 *
132233                          4410 * CPU CONTROL
132233                          4420 *
132233 000                      4430        NOP
132234 166                      4440        HALT
132235 363                      4450        DI
132236 373                      4460        EI
132237 355 106                  4470        IM   0
132241 355 126                  4480        IM   1
132243 355 136                  4490        IM   2
132245                          4500 *
132245                          4510 * ROTATES & SHIFTS
132245                          4520 *
132245 007                      4530        RLCA
132246 017                      4540        RRCA
132247 027                      4550        RLA
132250 037                      4560        RRA
132251 313 007                  4570        RLC  A
132253 313 000                  4580        RLC  B
132255 313 001                  4590        RLC  C
132257 313 002                  4600        RLC  D
132261 313 003                  4610        RLC  E
132263 313 004                  4620        RLC  H
132265 313 005                  4630        RLC  L
132267 313 006                  4640        RLC  (HL)
132271 335 313 002 006          4650        RLC  (IX+2)
132275 375 313 003 006          4660        RLC  (IY+THREE)
132301 313 017                  4670        RRC  A
132303 313 010                  4680        RRC  B
132305 313 011                  4690        RRC  C
132307 313 012                  4700        RRC  D
132311 313 013                  4710        RRC  E
132313 313 014                  4720        RRC  H
132315 313 015                  4730        RRC  L
132317 313 016                  4740        RRC  (HL)
```

```
132321 335 313 002 016      4750        RRC  (IX+2)
132325 375 313 003 016      4760        RRC  (IY+THREE)
132331 313 027              4770        RL   A
132333 313 020              4780        RL   B
132335 313 021              4790        RL   C
132337 313 022              4800        RL   D
132341 313 023              4810        RL   E
132343 313 024              4820        RL   H
132345 313 025              4830        RL   L
132347 313 026              4840        RL   (HL)
132351 335 313 002 026      4850        RL   (IX+2)
132355 375 313 003 026      4860        RL   (IY+THREE)
132361 313 037              4870        RR   A
132363 313 030              4880        RR   B
132365 313 031              4890        RR   C
132367 313 032              4900        RR   D
132371 313 033              4910        RR   E
132373 313 034              4920        RR   H
132375 313 035              4930        RR   L
132377 313 036              4940        RR   (HL)
133001 335 313 002 036      4950        RR   (IX+2)
133005 375 313 003 036      4960        RR   (IY+THREE)
133011 313 047              4970        SLA  A
133013 313 040              4980        SLA  B
133015 313 041              4990        SLA  C
133017 313 042              5000        SLA  D
133021 313 043              5010        SLA  E
133023 313 044              5020        SLA  H
133025 313 045              5030        SLA  L
133027 313 046              5040        SLA  (HL)
133031 335 313 002 046      5050        SLA  (IX+2)
133035 375 313 003 046      5060        SLA  (IY+THREE)
133041 313 057              5070        SRA  A
133043 313 050              5080        SRA  B
133045 313 051              5090        SRA  C
133047 313 052              5100        SRA  D
133051 313 053              5110        SRA  E
133053 313 054              5120        SRA  H
133055 313 055              5130        SRA  L
133057 313 056              5140        SRA  (HL)
133061 335 313 002 056      5150        SRA  (IX+2)
133065 375 313 003 056      5160        SRA  (IY+THREE)
133071 313 077              5170        SRL  A
133073 313 070              5180        SRL  B
133075 313 071              5190        SRL  C
133077 313 072              5200        SRL  D
133101 313 073              5210        SRL  E
133103 313 074              5220        SRL  H
133105 313 075              5230        SRL  L
133107 313 076              5240        SRL  (HL)
133111 335 313 002 076      5250        SRL  (IX+2)
133115 375 313 003 076      5260        SRL  (IY+THREE)
133121 355 157              5270        RLD
133123 355 147              5280        RRD
```

```
133125                          5290 *
133125                          5300 * BIT MANIPULATION
133125                          5310 * (PARTIAL LISTING)
133125                          5320 *
133125 313 107                  5330        BIT  0,A
133127 313 100                  5340        BIT  0,B
133131 313 101                  5350        BIT  0,C
133133 313 102                  5360        BIT  0,D
133135 313 103                  5370        BIT  0,E
133137 313 104                  5380        BIT  0,H
133141 313 105                  5390        BIT  0,L
133143 313 106                  5400        BIT  0,(HL)
133145 335 313 002 106          5410        BIT  0,(IX+2)
133151 375 313 003 106          5420        BIT  0,(IY+THREE)
133155 313 117                  5430        BIT  1,A
133157 313 120                  5440        BIT  2,B
133161 313 131                  5450        BIT  3,C
133163 313 142                  5460        BIT  4,D
133165 313 153                  5470        BIT  5,E
133167 313 164                  5480        BIT  6,H
133171 313 175                  5490        BIT  7,L
133173 313 176                  5500        BIT  7,(HL)
133175 335 313 002 176          5510        BIT  7,(IX+2)
133201 375 313 003 176          5520        BIT  7,(IY+THREE)
133205 313 207                  5530        RES  0,A
133207 313 200                  5540        RES  0,B
133211 313 201                  5550        RES  0,C
133213 313 202                  5560        RES  0,D
133215 313 203                  5570        RES  0,E
133217 313 204                  5580        RES  0,H
133221 313 205                  5590        RES  0,L
133223 313 206                  5600        RES  0,(HL)
133225 335 313 002 206          5610        RES  0,(IX+2)
133231 375 313 003 206          5620        RES  0,(IY+THREE)
133235 313 217                  5630        RES  1,A
133237 313 220                  5640        RES  2,B
133241 313 231                  5650        RES  3,C
133243 313 242                  5660        RES  4,D
133245 313 253                  5670        RES  5,E
133247 313 264                  5680        RES  6,H
133251 313 275                  5690        RES  7,L
133253 313 276                  5700        RES  7,(HL)
133255 335 313 002 276          5710        RES  7,(IX+2)
133261 375 313 003 276          5720        RES  7,(IY+THREE)
133265 313 307                  5730        SET  0,A
133267 313 300                  5740        SET  0,B
133271 313 301                  5750        SET  0,C
133273 313 302                  5760        SET  0,D
133275 313 303                  5770        SET  0,E
133277 313 304                  5780        SET  0,H
133301 313 305                  5790        SET  0,L
133303 313 306                  5800        SET  0,(HL)
133305 335 313 002 306          5810        SET  0,(IX+2)
```