JMON DISASSEMBLER FOR THE TALKING ELECTRONICS COMPUTER USING THE LCD SCREEN ON THE DAT BOARD. FILE: DIS-D_1.3.ROM

CODE WRITTEN BY JIM ROBERTSON
ANNOTATION BY BRIAN CHIHA

NOTES FROM PUBLICATION:

THE DISASSEMBLER IS MEMORY MAPPED AT 3000 AND WILL NOT RUN AT ANY OTHER LOCATION. IF YOU ARE USING THE ISSUE 12 PRINTER INTERFACE, THE DISASSEMBLER ROM CAN BE FITTED INTO THE ON- HOARD ROM SOCKET. YOU WILL HAVE TO SHIFT THE CS LINE THAT GOE., TO THE 74LS138 OUTPUT TO THE SECOND TOP OUTPUT. TO CHECK IF THE DISASSEMBLER IS FITTED PROPERLY, ADDRESS 3000. THE BYTES THERE SHOULD BE 21 C0 08. WITHOUT THE PRINTER INTERFACE, THE DISASSEMBLER CAN BE STACKED AS DESCRIBED IN ISSUE 14.

THE DISASSEMBLER IS DESIGNED TO WORK WITH THE JMON ROM. BUT IF YOU ARE NOT USING JMON, THE DISASSEMBLER CAN BE USED IF YOU PROVIDE THE START ADDRESS AT 0898 AND THE FINISH ADDRESS AT 089A (FINISHING ADDRESS NOT REQUIRED FOR THE LCD VERSION) THE DISPLAY BUFFER IS LOCATED AT 08C0. MAKE CERTAIN NOTHING VALUABLE IS THERE.

FOR THOSE OF YOU USING JMON, THE DISASSEMBLER IS ENTERED AT 37B0. YOU WILL BE PROMPTED BY THE JMON PERIMETER HANDLER TO ENTER THE ADDRESS FROM WHERE TO START DISASSEMBLING FROM. ENTER THE ADDRESS AND HIT THE "+" KEY. YOU ARE NOW ASKED FOR THE END ADDRESS. ENTER THIS AND HIT "GO"

IF USING THE DISASSEMBLER WITHOUT JMON, ENTER THE START AND END ADDRESS AT 0898 AND 089A. TO ENTER THE DISASSEMBLER, CALL 37BE. THE DISASSEMBLER WILL NOW WORK THE SAME AS IF A JMON WERE FITTED. (PRINTER VERSION ONLY)

FOR ADVANCED USERS:

THE OUTPUT OF THE DISASSEMBLER IS AN ASCII STRING LOCATED AT 08C0. THE END OF THE STRING IS EASY TO FIND. IT IS POINTED TO BY THE ADDRESS LOCATED AT 08A0. THIS ADDRESS POINTS TO THE NEXT AVAILABLE DISPLAY BUFFER ADDRESS, SO IF YOU WISH TO PLACE AN END MARKER THERE FOR YOUR OWN OUTPUT ROUTINES, USE THE FOLLOWING:

```
    2A AO 08      LD HL,(08A0)
    36 FF         LD (HL),FF
```

FOR YOUR OWN CUSTOM MASTER ROUTINES, THE DISASSEMBLER CAN BE CALLED AS A SUB-ROUTINE AT ADDRESS 3000.

TWO DIFFERENCES OCCUR BETWEEN THE STANDARD ZILOG SYNTAX AND THE DISASSEMBLER OUTPUT. THEY ARE: DISASSEMBLER LEAVES OF BRACKETS FROM IN A,(00) AND OUT (01),A AND OMITS A COMMA FROM CONDITIONAL RELATIVE JUMPS. POSSIBLY, THEY MAY BE CORRECTED IN A FUTURE UP-DATE.

START OF DISASSEMBLY:

SOME IMPORTANT ADDRESS ARE:
08C0 - THE DISASSEMBLER OUTPUT STRING START LOCATION
08A0 - THE ADDRESS OF THE END OF THE DISASSEMBLER OUTPUT STRING
08A2 - FLAG TO INDICATION IF REGISTER IS (HL)=NO BIT SET,(IX)=BIT 0 SET,
       (IY)=BIT 1 SET IS USED. ALSO, IF IX OR IY BIT 7 IS SET TO INDICATE
       IT'S AN INDEX BIT INSTRUCTION.

THIS IS THE START OF THE DISASSEMBLER. IT REQUIRES THE
START ADDRESS TO BE IN 0898 AND THE END ADDRESS IN 089A.
THESE ADDRESS CAN EITHER ENTERED MANUALLY OR USING THE JMON
PERIMETER HANDLER WHICH CAN BE CALLED AT 37B0

```
3000 21 C0 08       LD HL,08C0       ;OUTPUT ADDRESS FOR ASCII STRING
3003 AF             XOR A            ;ZERO A
3004 32 A2 08       LD (08A2),A      ;CLEAR THE HL,INDEX FLAG
3007 22 A0 08       LD (08A0),HL     ;SET THE END OF OUTPUT TO START
```

```
300A 2A 98 08       LD HL,(0898)     ;LOAD THE START ADDRESS TO HL
300D E5             PUSH HL          ;SAVE HL
300E CD 8A 31       CALL 318A        ;CALL THE CONVERT HL TO ASCII ROUTINE
                                     ;WHICH PUTS THE ADDRESS IN THE OUTPUT
                                     ;STRING
```

FILL THE REST OF THE OUTPUT STRING WITH 32 SPACES. THIS IS IMPORTANT TO CLEAR THE LCD FOR THE NEXT INSTRUCTION

```
3011 06 20          LD B,20          ;32 SPACES
3013 CD B9 31       CALL 31B9        ;CALL ADD SPACE TO OUTPUT ROUTINE

3016 3E C5          LD A,C5          ;RESET OUTPUT STRING END TO BE AT 08C5
3018 32 A0 08       LD (08A0),A      ;WHICH IS 6 COLUMNS IN
301B E1             POP HL           ;RESTORE HL (CURRENT ADDRESS)
301C 7E             LD A,(HL)        ;LOAD A WITH OP CODE
301D E5             PUSH HL          ;SAVE HL
```

CHECK IF OP CODE IS ONE OF THE ONE BYTE NO VARIABLE OP CODES. THESE ARE THE EASIEST TO DISPLAY. AS NO FURTHER ADDRESS LOCATIONS ARE NEEDED.

```
301E 16 01          LD D,01          ;NUMBER OF OP CODES TO DISPLAY
3020 CD 1D 35       CALL 351D        ;CALL ONE BYTE OP CODE LOOKUP ROUTINE
3023 E1             POP HL           ;SETS CARRY FLAG IF NO OP CODE FOUND
3024 30 08          JR NC,302E       ;IF CARRY SET THEN NO ONE BYTE OP CODE FOUND
                                     ;JUMP TO LCD UPDATE IF FOUND

3026 CD 31 30       CALL 3031        ;CALL MAIN OP CODE ROUTINE
3029 4E             LD C,(HL)        ;LOAD CURRENT OP CODE TO C (FOR BACKUP)
302A 7E             LD A,(HL)        ;LOAD CURRENT OP CODE TO A
302B D4 0E 34       CALL NC,340E     ;CALL HANDLE THE REST OF THE OP CODES ROUTINE
```

LCD UPDATE AND HALT. THIS IS CALLED LAST AFTER ALL OF THE OUTPUT STRING IS PUT TOGETHER. ONCE HERE, THE HALT COMMAND IS SENT AND ONLY ON AN NMI TRIGGER THE EXECUTION WILL CONTINUE. WHEN A KEY IS PRESSED IT WILL CALL 3000 AGAIN.

```
302E C3 9E 35       JP 359E          ;INCREASE START ADDRESS AND UPDATE LCD
                                     ;EXITING OUT OF THIS ROUTINE TO UPDATE LCD
                                     ;SCREEN AFTER PERIMETER HANDLER. LDC UPDATE
                                     ;WILL RETURN
```

MAIN OP CODE ROUTINE. THE Z80 OP CODES ARE STRUCTURED IN A WAY THAT GROUPS SIMILAR OPERATIONS TOGETHER. THIS ROUTINE SPLITS THESE OPERATIONS INTO SIX GROUPINGS BASED ON THE OP CODE RANGE OR CODE. THE SPLIT FOR THE OP CODES ARE:

```
00-39 - DECREMENT AND INCREMENT OPERATIONS
40-BF - LOAD, ADD, SUBTRACT, AND, OR, XOR AND COMPARE
CB    - BIT OPERATIONS
DD    - IX OPERATIONS
FD    - IX OPERATIONS
OTHER - THE REST OF THE OPERATIONS
```

```
3031 FE 40          CP 40            ;IS OP CODE DEC OR INC?
3033 38 18          JR C,304D        ;JUMP IF TRUE AND EXIT ROUTINE
3035 FE C0          CP C0            ;IS OP CODE LD, ADD, SUB, AND, OR, XOR OR CP?
3037 38 16          JR C,304F        ;JUMP IF TRUE
3039 FE CB          CP CB            ;IS OP CODE A BIT OPERATION?
303B CA DE 31       JP Z,31DE        ;JUMP IF TRUE
303E 0E 00          LD C,00          ;SETS THE WHICH BRACKETED REGISTER TO USE C=0 (HL)
3040 FE DD          CP DD            ;IS OP CODE AN IX OPERATION? C=1 (IX+ )
3042 28 05          JR Z,3049        ;JUMP IF TRUE TO MAKE C=1
3044 FE FD          CP FD            ;IS OP CODE AN IY OPERATION? C=2 (IY+ )
3046 20 05          JR NZ,304D       ;JUMP IF NOT TRUE AND EXIT ROUTINE
3048 0C             INC C            ;ADD 1 TO C;
```

```
3049 0C            INC C           ;ADD 1 TO C;
304A C3 D7 30      JP 30D7         ;JUMP FOR IX,IY OPERATION
304D B7            OR A            ;RESET CARRY FLAG
304E C9            RET             ;EXIT
```

LOAD, ADD, SUBTRACT, AND, OR, XOR OR COMPARE ROUTINE. OP CODE IS BETWEEN 40 AND BF. ALL HAVE ONLY ONE OP CODE. IT FIRST WRITES THE OP CODE, THEN CHECK FOR PARAMETERS OF THE OP CODE IF ANY.

```
304F F5            PUSH AF         ;SAVE AF
3050 CD 55 30      CALL 3055       ;JUMP TO OP CODE WRITE TO OUTPUT
3053 18 05         JR 305A         ;IGNORE JUMP TO 31FF
3055 06 01         LD B,01         ;WRITE ONE OP CODE TO
3057 C3 FF 31      JP 31FF         ;CALL WRITE OP CODE TO OUTPUT ROUTINE
305A F1            POP AF          ;RESTORE AF
305B FE 80         CP 80           ;IS OP CODE BELOW OR 80-BF
305D 30 14         JR NC,3073      ;JUMP FOR 80-BF
```

OP CODE BETWEEN 40-7F LOAD ROUTINES. DISPLAY LD AND THEN INDEX THE FIRST AND SECOND REGISTERS. OUTPUT THE FIRST AND SECOND REGISTERS.

```
305F F5            PUSH AF         ;SAVE AF
3060 CD 05 32      CALL 3205       ;CALL MNEMONIC ENTER FOR LOAD OP
3063 F1            POP AF          ;RESTORE AF
3064 CD AC 31      CALL 31AC       ;CALL REGISTER INDEX ROUTINE
3067 F5            PUSH AF         ;SAVE AF, A=SECOND OP, C=FIRST OP
3068 CD 06 31      CALL 3106       ;CALL OUTPUT REGISTER ROUTINE
306B CD 64 31      CALL 3164       ;ADD COMMA TO OUTPUT STRING
306E F1            POP AF          ;RESTORE AF
306F 4F            LD C,A          ;LOAD SECOND OP IN C
3070 C3 07 31      JP 3107         ;CALL OUTPUT REGISTER ROUTINE AND EXIT
```

OP CODE BETWEEN 80-BF. WHICH COULDE BE ADD,SUB,AND,OR,ADC,SBC,XOR,CP

```
3073 E6 3F         AND 3F          ;MASK FOR ADD,SUB,AND OR OR
3075 CD AC 31      CALL 31AC       ;CALCULATE MNEMONIC INDEX
3078 F5            PUSH AF         ;STORE AF A=SECOND OP, C=FIRST OP
3079 3E 86         LD A,86         ;LOAD BASE MNEMONIC INDEX (ADD)
307B 81            ADD A,C         ;MULTIPLY INDEX BY 3
307C 81            ADD A,C         ;TO GET THE CORRECT
307D 81            ADD A,C         ;MNEMONIC
307E CD 07 32      CALL 3207       ;CALL THE MNEMONIC LOOKUP ROUTINE
3081 CD 86 30      CALL 3086       ;JUMP BELOW
3084 18 0C         JR 3092         ;CONTINUE WITH NEXT REGISTER
3086 CD 7B 31      CALL 317B       ;CHECK IF REQUIRES 1 OR TWO REGISTERS?
3089 C0            RET NZ          ;IF NZ THEN ONLY ONE NEEDED
308A 0E 07         LD C,07         ;INDEX FOR A REGISTERS
308C CD 0B 31      CALL 310B       ;CALL REGISTER LOOKUP TABLE
308F C3 64 31      JP 3164         ;CALL ADD COMMA ROUTINE
3092 F1            POP AF          ;RESTORE AF
3093 4F            LD C,A          ;PUT REGISTER INDEX IN C
3094 18 70         JR 3106         ;WRITE THE SECOND REGISTER AND EXIT
```

BIT OPERATOR MNEMONIC ROUTINE. FIRSTLY IT CHECK IF OP CODE IS A ROTATE/SHIFT (00-3F) OR BIT INSTRUCTION (40-FF).

```
3096 FE 40         CP 40           ;IS OP CODE 40 AND ABOVE OR BELOW?
3098 30 16         JR NC,30B0      ;ABOVE, SO JUMP TO BIT ROUTINE
```

OP CODE IS A ROTATE OR SHIFT

```
309A CD AC 31      CALL 31AC       ;CALL REGISTER INDEX ROUTINE C=LOOKUP A=TYPE
309D F5            PUSH AF         ;SAVE REGISTER TYPE
309E 79            LD A,C          ;LOAD LOOKUP INDEX IN A
309F FE 07         CP 07           ;IS IT THE A REGISTER?
```

```
30A1 20 01          JR NZ,30A4       ;NO
30A3 0D             DEC C            ;DECREASE INDEX BY 1
30A4 3E 9E          LD A,9E          ;LOAD INDEX OF 'RLC' IN A
30A6 81             ADD A,C          ;MULTIPLY INDEX BY 3 TO GET
30A7 81             ADD A,C          ;ACTUAL MNEMONIC REFERENCE
30A8 81             ADD A,C          ;
30A9 CD 07 32       CALL 3207        ;CALL ASCII MNEMONIC LOOKUP ROUTINE
30AC F1             POP AF           ;RESTORE OP CODE
30AD 4F             LD C,A           ;LOAD TYPE IN C
30AE 18 56          JR 3106          ;REGISTER OUTPUT ROUTINE AND EXIT
```

OP CODE IS A BIT,SET,RES

```
30B0 D6 40          SUB 40           ;SUBTRACT 40 FOR EASIER INDEXING
30B2 F5             PUSH AF          ;SAVE OP CODE
30B3 06 B3          LD B,B3          ;LOAD B WITH INDEX OF 'BIT'
30B5 FE 40          CP 40            ;IS IT A 'BIT' OP CODE?
30B7 38 08          JR C,30C1        ;YES, GO TO OUTPUT ROUTINE
30B9 06 B6          LD B,B6          ;LOAD B WITH INDEX OF 'RES'
30BB FE 80          CP 80            ;IT IT A 'RES' OP CODE?
30BD 38 02          JR C,30C1        ;YES, GO TO OUTPUT ROUTINE
30BF 06 B9          LD B,B9          ;MUST BE A 'SET' OP CODE
30C1 78             LD A,B           ;PUT LOOKUP INDEX IN A
30C2 CD 07 32       CALL 3207        ;CALL ASCII MNEMONIC LOOKUP ROUTINE
30C5 F1             POP AF           ;RESTORE OP CODE
30C6 E6 3F          AND 3F           ;MASK OUT BITS 7,6 FOR REGISTER INDEXING
30C8 CD AC 31       CALL 31AC        ;CALL REGISTER INDEX ROUTINE C=LOOKUP A=TYPE
30CB F5             PUSH AF          ;SAVE TYPE
30CC 79             LD A,C           ;LOAD INDEX IN A
30CD CD 9A 31       CALL 319A        ;CALL A TO ASCII CONVERTER TO OUTPUT THE BIT
30D0 CD 64 31       CALL 3164        ;ADD A COMMA
30D3 F1             POP AF           ;RESTORE INDEX
30D4 4F             LD C,A           ;LOAD INDEX IN C
30D5 18 2F          JR 3106          ;REGISTER OUTPUT ROUTINE AND EXIT
```

IX AND IY INSTRUCTION ENTRY POINT. DETERMINE IF JUST IX, IX OR THE BIT
INSTRUCTION CB. THEN EITHER HANDLE SPECIFICALLY OR JUST NORMALLY.

```
30D7 2A 98 08       LD HL,(0898)     ;LOAD CURRENT ADDRESS TO HL (EITHER DD OR FD)
30DA 23             INC HL           ;MOVE TO CHECK SECOND BYTE
30DB 7E             LD A,(HL)        ;STORE IT IN A
30DC FE CB          CP CB            ;IS IT A BIT INSTRUCTION CB?
30DE 28 0A          JR Z,30EA        ;IF IT IS THEN HANDLE IT AT 30EA
30E0 FE BF          CP BF            ;IS INSTRUCTION BETWEEN 00-BF? SET CARRY FLAG
30E2 2B             DEC HL           ;GO BACK TO DD OR FF
30E3 D0             RET NC           ;IF INSTRUCTION BETWEEN C0 OR GREATER THEN RETURN
30E4 FE 40          CP 40            ;IS INSTRUCTION BETWEEN 00-3F? SET CARRY FLAG
30E6 30 05          JR NC,30ED       ;INSTRUCTION IS BETWEEN 40-BF? THEN CONTINUE BELOW
30E8 A7             AND A            ;INSTRUCTION IS BETWEEN 00-3F CLEAR FLAGS AND
30E9 C9             RET              ;EXIT BACK
```

IX AND IY BIT INSTRUCTION.

```
30EA CB F9          SET 7,C          ;SET BIT SEVEN TO INDICATION ITS A BIT INSTRUCTION
30EC 23             INC HL           ;MOVE TO LAST BYTE IN OPERATION AT THIS INDICATES
                                     ;WHAT TYPE OF BIT INSTRUCTION IT IS.
30ED 23             INC HL           ;ENTERED HERE IF NOT BIT BUT BETWEEN 40-BF
30EE 7E             LD A,(HL)        ;LOAD OP CODE
30EF F5             PUSH AF          ;SAVE AF
30F0 06 04          LD B,04          ;NUMBER OF OP CODES TO OUTPUT
30F2 79             LD A,C           ;SAVE REGISTER FLAG TO A
30F3 32 A2 08       LD (08A2),A      ;STORE FLAG IN MEMORY.
30F6 CB 79          BIT 7,C          ;IS BIT 7 SET? IE: A BIT INSTRUCTION
30F8 20 01          JR NZ,30FB       ;YES, OUTPUT 4 OP CODES
30FA 05             DEC B            ;NO, ONLY OUTPUT 3 OP CODES
```

```
30FB CD FF 31      CALL 31FF       ;CALL WRITE OP CODES TO OUTPUT STRING ROUTINE
30FE F1            POP AF          ;RESTORE AF
30FF CB 79         BIT 7,C         ;IS BIT 7 SET? IE: A BIT INSTRUCTION
3101 20 93         JR NZ,3096      ;YES, JUMP TO BIT MNEMONIC OUTPUT ROUTINE
3103 C3 5B 30      JP 305B         ;NO JUMP TO THE LD,ADD,SUB,AND ETC OUTPUT ROUTINE
```

CALCULATE WHAT REGISTER IS BEING USED. STORED IN C, REGISTER ARE INDEXED IN THIS ORDER. B,C,D,E,H,L,(HL),A FROM 0-7

```
3106 79            LD A,C          ;LOAD INDEX TO A
3107 FE 06         CP 06           ;IS IT HL/IX/IY?
3109 28 07         JR Z,3112       ;JUMP TO THE HL/IX/IY OUTPUT ROUTINE
310B 3E 01         LD A,01         ;ONLY ONE REGISTER OR COMMA
310D 21 83 37      LD HL,3783      ;REGISTER TABLE ENTRY
3110 18 33         JR 3145         ;DO REGISTER LOOKUP AND WRITE TO OUTPUT STRING
```

OUTPUT THE REGISTERS (HL),(IX+), AND (IY+). IT USES THE HL,IX,IY FLAG TO CHECK WHICH REGISTER TO OUTPUT. IF FLAG IS 0, ITS HL, 1 FOR IX AND 2 FOR IY. CALLS THE REGISTER LOOKUP TABLE AND OUTPUTS THE ASCII TO THE OUTPUT STRING.

```
3112 3A A2 08      LD A,(08A2)     ;CHECK HL,IX,IY FLAG
3115 B7            OR A            ;IF IT 0 ITS HL
3116 20 06         JR NZ,311E      ;MUST BE IX, OR IY
3118 0E 08         LD C,08         ;SET THE INDEX OF (HL)
311A 3E 04         LD A,04         ;FOUR CHARACTERS TO PRINT
311C 18 EF         JR 310D         ;JUMP TO THE REGISTER OUTPUT ROUTINE AND EXIT
311E F5            PUSH AF         ;SAVE AF
311F 1F            RRA             ;ROTATE RIGHT 1 TO SET IF BIT 0 IS SET
3120 0E 0C         LD C,0C         ;SET THE INDEX OF (IX)
3122 38 02         JR C,3126       ;IF BIT 0 IS SET, MUST BE (IX), JUMP TO OUTPUT
3124 0E 13         LD C,13         ;SET THE INDEX IF (IY)
3126 3E 07         LD A,07         ;SEVEN CHARACTERS TO PRINT
3128 CD 0D 31      CALL 310D       ;CALL THE REGISTER OUTPUT ROUTINE
312B F1            POP AF          ;RESTORE AF
312C 17            RLA             ;CHECK IF BIT 7 IS SET (BIT INSTRUCTION)
312D ED 5B 98 08   LD DE,(0898)    ;LOAD DE WITH CURRENT ADDRESS OF VARIABLE
3131 30 01         JR NC,3134      ;IF NOT A BIT OPERATION JUMP TO 3134
3133 1B            DEC DE          ;MOVE ADDRESS BACK ONE AS BIT VARIABLE IS THERE
3134 3A A0 08      LD A,(08A0)     ;FIND END OF OUTPUT STRING
3137 D6 03         SUB 03          ;GO BACK THREE TO BE ON THE VARIABLE PART
                                   ;JUST AFTER THE + SIGN. IE: (IX+**)
3139 32 A0 08      LD (08A0),A     ;UPDATE END OF OUTPUT STRING ADDRESS
313C 1A            LD A,(DE)       ;LOAD A WITH THE BYTE AT CURRENT ADDRESS
313D CD 91 31      CALL 3191       ;CALL CONVERT TO ASCII AND WRITE TO OUTPUT STRING
3140 23            INC HL          ;HL IS NOW ON RIGHT BRACKET, INCREASE BY 1 TO
                                   ;BE ON THE END OF THE OUTPUT STRING.
3141 22 A0 08      LD (08A0),HL    ;SAVE END OF OUTPUT STRING ADDRESS
3144 C9            RET             ;EXIT
```

COPY REGISTER ASCII TO OUTPUT STRING USING C AS AN INDEX. B IS SET EXTERNALLY

```
3145 ED 5B A0 08   LD DE,(08A0)    ;LOAD DE WITH END OF OUTPUT STRING
3149 09            ADD HL,BC       ;ADD INDEX TO HL (POINTING TO REF TABLE)
314A 4F            LD C,A          ;STORE REGISTER LENGTH IN C
314B ED B0         LDIR            ;DO THE COPY
314D ED 53 A0 08   LD (08A0),DE    ;SAVE END OF OUTPUT STRING
3151 37            SCF             ;SET CARRY FLAG
3152 C9            RET             ;EXIT
```

ASCII LOOKUP INDEX ROUTINE THAT TAKES 62H FROM HL USING A 16 BIT SUBTRACTION A 16 BIT SUBTRACTION IS DONE BECAUSE THE LOOK UP TABLE IS OVER A 255 (FF) BLOCK

```
3153 26 37         LD H,37         ;SET HL TO BE 37 + A INDEX
3155 6F            LD L,A          ;LOAD L WITH A INDEX
```

```
3156 11 9E FE      LD DE,FE9E      ;16 BIT SUBTRACTION OF 62H
3159 19            ADD HL,DE       ;FROM HL
315A EB            EX DE,HL        ;STORE HL IN DE
315B 18 0C         JR 3169         ;DO THE ASCII LOOKUP BASED ON DE
```

MOVE OUTPUT STRING TO SECOND LINE. THIS IS CALLED AFTER ALL OP CODES HAVE BEEN WRITTEN. IT RETURNS BACK TO THE ROUTINE THE CALLED THE OP CODE WRITE.

```
315D 21 D2 08      LD HL,08D2      ;MOVE HL TO SECOND LINE IN OUTPUT STRING
3160 22 A0 08      LD (08A0),HL    ;PUT HL IN END OF OUTPUT STRING VARIABLE
3163 C9            RET             ;EXIT
```

ADD A COMMA "," TO THE OUTPUT REGISTER. CALL REGISTER LOOKUP TABLE WITH INDEX OF 6

```
3164 01 06 00      LD BC,0006      ;INDEX OF 6
3167 18 A2         JR 310B         ;CALL REGISTER LOOKUP TABLE
```

ASCII LOOKUP WITH DE AS THE LOOKUP TABLE ADDRESS

```
3169 2A A0 08      LD HL,(08A0)    ;LOAD HL WITH OUTPUT STRING END
316C 1A            LD A,(DE)       ;GET ASCII IN A WITH REFERENCE TO DE
316D 77            LD (HL),A       ;PUT CHARACTER IN OUTPUT STRING
316E CB BE         RES 7,(HL)      ;REMOVE 7TH BIT IF SET
3170 23            INC HL          ;MOVE TO NEXT OUTPUT STRING
3171 22 A0 08      LD (08A0),HL    ;SAVE HL TO END OF OUTPUT STRING ADDRESS
3174 13            INC DE          ;MOVE TO NEXT IN REFERENCE TABLE
3175 B7            OR A            ;CHECK A FOR SIGN FLAG (BIT 7 SET)
3176 FA B7 31      JP M,31B7       ;IF SIGN NEGATIVE, JUMP TO ADD SPACE ROUTINE
3179 18 EE         JR 3169         ;GET NEXT ASCII CHARACTER
```

CHECK IF OP CODE IS ADD,SUB,ADC,SBC OR AND,XOR,OR,CP. THIS WILL SET A TO BE ZERO OR NON-ZERO. IF ZERO THEN A REGISTER IS USED AS FIRST REGISTER.

```
317B 79            LD A,C          ;PUT INDEX IN A
317C FE 04         CP 04           ;DO COMPARISON
317E 38 02         JR C,3182       ;CARRY SET IF ADD,SUB,ADC,SBC
3180 B7            OR A            ;RESET A
3181 C9            RET             ;EXIT
3182 FE 02         CP 02           ;CHECK FOR SBC
3184 20 02         JR NZ,3188      ;MUST BE SUB
3186 3D            DEC A           ;MAKE A=1
3187 C9            RET             ;EXIT
3188 AF            XOR A           ;MAKE A=0
3189 C9            RET             ;EXIT
```

CONVERT HL TO ASCII AND STORE IT IN THE OUTPUT STRING

```
318A E5            PUSH HL         ;SAVE HL
318B 7C            LD A,H          ;STORE MSB OF HL IN A
318C CD 91 31      CALL 3191       ;CALL CONVERT A TO ASCII ROUTINE
318F E1            POP HL          ;RESTORE HL
3190 7D            LD A,L          ;STORE LSB OF HL IN A
```

CONVERT A IN HEX TO ASCII IN HEX. FIRST LOOK UP HIGH NIBBLE THEN LOW NIBBLE. IE CONVERT 0CH TO "0C" IN ASCII WHICH IS 30H AND 43H SAVE ASCII TO OUTPUT STRING AND INCREMENT END OF OUTPUT STRING ADDRESS

```
3191 F5            PUSH AF         ;SAVE A FOR LATER
3192 1F            RRA             ;MOVE HIGH NIBBLE TO LOW NIBBLE
3193 1F            RRA
3194 1F            RRA
3195 1F            RRA
3196 CD 9A 31      CALL 319A       ;CALL HEX TO ASCII CONVERTER
```

```
3199 F1            POP AF          ;RESTORE AF AND FALL THROUGH
```

HEX TO ASCII CONVERTER. A CLEVER ROUTINE THAT CONVERTS THE LOW NIBBLE OF A FROM HEXADECIMAL TO ASCII.

```
319A E6 0F         AND 0F          ;MASK ONLY LOWER NIBBLE
319C C6 90         ADD A,90        ;ADD 90H TO A
319E 27            DAA             ;BCD ADDITION WHICH COULD AFFECT CARRY FLAG
319F CE 40         ADC A,40        ;ADD 40H TO A (WITH CARRY IF ANY)
31A1 27            DAA             ;BCD ADDITION
31A2 2A A0 08      LD HL,(08A0)    ;LOAD HL WITH END OF OUTPUT STRING ADDRESS
31A5 77            LD (HL),A       ;WRITE THE ASCII HEX VALUE
31A6 23            INC HL          ;MOVE TO NEXT STRING ADDRESS
31A7 22 A0 08      LD (08A0),HL    ;SAVE HL TO END OF OUTPUT STRING ADDRESS
31AA 37            SCF             ;SET THE CARRY FLAG TO INDICATE SUCCESS
31AB C9            RET             ;EXIT
```

REGISTER AND OP MNEMONIC CALCULATION. ROTATE BITS 38 RIGHT 3 TIMES TO WORK OUT INDEX TO REGISTER OR MNEMONIC LOOKUP TABLE. LOOKUP INDEX STORE IN C, TYPE REFERENCE STORED IN A

```
31AC F5            PUSH AF         ;SAVE OP CODE
31AD E6 38         AND 38          ;CALCULATE REGISTER BASED ON THE
31AF 1F            RRA             ;BITS SET
31B0 1F            RRA
31B1 1F            RRA
31B2 4F            LD C,A          ;SAVE INDEX IN C
31B3 F1            POP AF          ;RESTORE OP CODE
31B4 E6 07         AND 07          ;MASK LAST 3 BITS
31B6 C9            RET             ;EXIT
```

ADD SPACE CHARACTER " " TO OUTPUT STRING. USES B REGISTER WHICH IS SET OUTSIDE THIS CALL IF MORE THAN 1 SPACE NEEDED (ENTER AT 31B9) INCREMENT END OF OUTPUT STRING ADDRESS

```
31B7 06 01         LD B,01         ;FALL TROUGH TO ADD 1 SPACE CHARACTER
31B9 3E 20         LD A,20         ;LOAD 20H, ASCII SPACE CHARACTER " "
31BB 2A A0 08      LD HL,(08A0)    ;LOAD HL WITH END OF OUTPUT STRING ADDRESS
31BE 77            LD (HL),A       ;SAVE SPACE
31BF 23            INC HL          ;MOVE TO NEXT STRING ADDRESS
31C0 10 FC         DJNZ 31BE       ;REPEAT B TIMES
31C2 22 A0 08      LD (08A0),HL    ;SAVE HL TO END OF OUTPUT STRING ADDRESS
31C5 C9            RET             ;EXIT
```

CONVERT CURRENT OP CODE TO ASCII AND ADD IT TO THE OUTPUT STRING. SEPARATE OP CODES WITH A SPACE. USES B FOR THE NUMBER OF OP CODES

```
31C6 ED 5B 98 08   LD DE,(0898)    ;LOAD START ADDRESS TO DE
31CA C5            PUSH BC         ;SAVE BC
31CB 1A            LD A,(DE)       ;LOAD THE OP CODE TO A
31CC F5            PUSH AF         ;SAVE AF
31CD CD 91 31      CALL 3191       ;CONVERT A TO ASCII AND STORE IN OUTPUT STRING
31D0 CD B7 31      CALL 31B7       ;ADD A SPACE " " TO OUTPUT STRING
31D3 F1            POP AF          ;RESTORE AF
31D4 13            INC DE          ;MOVE TO NEXT OP CODE ADDRESS
31D5 C1            POP BC          ;RESTORE BC
31D6 10 F2         DJNZ 31CA       ;WRITE NEXT OP CODE
31D8 1B            DEC DE          ;DECREASE START ADDRESS BY ONE
31D9 ED 53 98 08   LD (0898),DE    ;SAVE NEW START ADDRESS
31DD C9            RET             ;EXIT
```

BIT OPERATION ENTRY. BIT OPERATORS HAVE TWO OP CODES. OUTPUT OP CODES TO OUTPUT STRING AND MOVE TO NEXT ROW

```
31DE 06 02         LD B,02         ;TWO OP CODES TO OUTPUT
```

```
31E0 CD C6 31       CALL 31C6        ;CALL OP CODE TO ASCII OUTPUT ROUTINE
31E3 CD 5D 31       CALL 315D        ;CALL MOVE OUTPUT TO NEXT ROW ROUTINE
31E6 C3 96 30       JP 3096          ;JUMP TO BIT OPERATOR MNEMONIC ROUTINE
```

CHECK OP CODE (00-3E) TO SEE IF ITS A LD rr,** OR SOMETHING ELSE

```
31E9 E6 CF          AND CF           ;MASK FOR LOW NIBBLE
31EB FE 01          CP 01            ;MUST BE A LD rr,** COMMAND
31ED 20 47          JR NZ,3236       ;JUMP IF NOT
31EF CD FD 31       CALL 31FD        ;WRITE 3 OP CODES AND MOVE TO SECOND ROW
31F2 CD 05 32       CALL 3205        ;WRITE 'LD' TO OUTPUT STRING
31F5 CD 0A 32       CALL 320A        ;WRITE DOUBLE REGISTER TO OUTPUT STRING
31F8 CD 64 31       CALL 3164        ;WRITE COMMA TO OUTPUT STRING
31FB 18 24          JR 3221          ;WRITE THE NEXT TWO OP CODES AS AN ADDRESS THEN EXIT

31FD 06 03          LD B,03          ;THREE OP CODES TO PRINT ENTRY POINT FALL THROUGH
```

WRITE B NUMBER OF OP CODES TO OUTPUT STRING AND MOVE OUTPUT STRING END TO SECOND ROW. THIS ROUTINE THEN RETURNS BACK TO THE PREVIOUS CALL

```
31FF CD C6 31       CALL 31C6        ;CALL WRITE OP CODE TO OUTPUT ROUTINE
3202 C3 5D 31       JP 315D          ;MOVE OUTPUT TO SECOND LINE AND RETURN
                                     ;TO CALLED ROUTINE
```

LOAD MNEMONIC ENTRY POINT.

```
3205 3E 83          LD A,83          ;LOAD A 83 INDEX FOR "LD .."
3207 C3 53 31       JP 3153          ;JUMP TO ASCII INDEXING ROUTINE
```

FIND DOUBLE REGISTER COMBINATION BASED OFF OP CODE

```
320A 79             LD A,C           ;GET ORIGINAL OP CODE
320B F5             PUSH AF          ;SAVE IT
320C E6 30          AND 30           ;IS IT BC=0, DE=1, HL/IX/IY=2 OR SP=3
320E CD A6 35       CALL 35A6        ;CALL UPDATE INDEX FOR HL,IX,IY
3211 C6 1A          ADD A,1A         ;INDEX A FOR LOOKUP
3213 06 00          LD B,00          ;RESET B
3215 4F             LD C,A           ;SAVE INDEX IN C
3216 3E 02          LD A,02          ;TWO CHARACTERS
3218 21 83 37       LD HL,3783       ;DOUBLE REGISTER LOOKUP TABLE
321B CD 45 31       CALL 3145        ;LOOKUP REGISTER AND WRITE TO OUTPUT STRING
321E F1             POP AF           ;RESTORE AF
321F 4F             LD C,A           ;PUT OP CODE BACK IN C
3220 C9             RET              ;EXIT
```

WRITE THE NEXT TWO OP CODES TO THE OUTPUT STRING AS AN ADDRESS. THIS IS USED FOR ADDRESS REFERENCING. AS THIS IS THE OUTPUT OF THE MNEMONIC THE ADDRESS POINTING TO 0898 IS ON THE MSB OF THE 16 BIT ADDRESS. THEN DECREASE HL AND OUTPUT THE LSB. IF THE ADDRESS IS 34 12 THEN 1234 IS WRITTEN

```
3221 2A 98 08       LD HL,(0898)     ;GET MSB OF THE ADDRESS
3224 7E             LD A,(HL)        ;STORE IT IN A
3225 E5             PUSH HL          ;SAVE HL
3226 CD 91 31       CALL 3191        ;CALL CONVERT A TO ASCII AND OUTPUT ROUTINE
3229 E1             POP HL           ;RESTORE HL
322A 2B             DEC HL           ;MOVE HL BACK ONE TO GET THE LSB OF THE ADDRESS
322B 7E             LD A,(HL)        ;STORE IT IN A
322C E5             PUSH HL          ;SAVE HL
322D CD 91 31       CALL 3191        ;CALL CONVERT A TO ASCII AND OUTPUT ROUTINE
3230 E1             POP HL           ;RESTORE HL
3231 23             INC HL           ;MOVE HL BACK TO END OF OP CODE
3232 22 98 08       LD (0898),HL     ;SAVE IT BACK INTO THE START ADDRESS
3235 C9             RET              ;EXIT
```

CHECK FOR ONE VARIABLE IS NEEDED, IE TWO OP CODES. THESE ARE *6 OR *E OP CODES

```
3236 E6 C7         AND C7          ;CHECK TO SEE IF OP CODE HAS ONE VARIABLE
3238 FE 06         CP 06           ;HAS EITHER *6 OR *E ON LOW NIBBLE
323A 20 18         JR NZ,3254      ;IF NOT JUMP
323C 06 02         LD B,02         ;TWO OP CODES
323E CD BE 32      CALL 32BE       ;CALL DISPLAY OP CODES AND MNEMONIC REF
3241 79            LD A,C          ;LOAD OP CODE BACK TO A REGISTER
3242 CD AC 31      CALL 31AC       ;CALL SECOND MNEMONIC INDEX REFERENCE
3245 06 00         LD B,00         ;RESET B FOR LDIR INSTRUCTION
3247 CD E7 35      CALL 35E7       ;WRITE REGISTER TO OUTPUT STRING CHECKING FOR IX/IY
324A CD 64 31      CALL 3164       ;ADD COMMA
324D 2A 98 08      LD HL,(0898)    ;LOAD NEXT OP CODE
3250 7E            LD A,(HL)       ;SAVE IN A
3251 C3 91 31      JP 3191         ;WRITE OP CODE TO OUTPUT STRING
```

THIS WILL CHECK IF OP CODE IS A LD A,(BC) OR LD A,(DE) AND PRINT IT.

```
3254 79            LD A,C          ;RESTORE CURRENT OP CODE
3255 F5            PUSH AF         ;SAVE IT
3256 E6 EF         AND EF          ;MASK FOR OP CODE
3258 FE 0A         CP 0A           ;IS OP CODE A LD A,(BC) OR LD A,(DE)
325A 20 1F         JR NZ,327B      ;MOVE TO NEXT OP CODE CHECK
325C 06 01         LD B,01         ;ONE OP CODE TO DISPLAY
325E CD BE 32      CALL 32BE       ;CALL DISPLAY OP CODES AND MNEMONIC REF
3261 79            LD A,C
3262 01 07 00      LD BC,0007      ;INDEX FOR A REGISTER
3265 CD 0B 31      CALL 310B       ;CALL REGISTER OUTPUT ROUTINE
3268 CD 64 31      CALL 3164       ;ADD COMMA
326B 01 08 00      LD BC,0008      ;INDEX FOR LEFT BRACKET
326E CD 0B 31      CALL 310B       ;CALL REGISTER OUTPUT ROUTINE
3271 F1            POP AF          ;RESTORE A
3272 CD 0B 32      CALL 320B       ;CALL DOUBLE REGISTER OUTPUT ROUTINE
3275 01 0B 00      LD BC,000B      ;INDEX FOR RIGHT BRACKET
3278 C3 0B 31      JP 310B
```

CHECK FOR OP CODE OF LD (BC),A OR LD (DE),A

```
327B FE 02         CP 02           ;IS IT OP CODE 02 OR 12?
327D 20 11         JR NZ,3290      ;CONTINUE WITH OP CODE CHECK
327F 06 01         LD B,01         ;ONE OP CODE TO DISPLAY
3281 CD BE 32      CALL 32BE       ;CALL DISPLAY OP CODES AND MNEMONIC REF
3284 CD 03 36      CALL 3603       ;CALL HL/AF SWAP AND JUMP TO 326B ABOVE
3287 CD 64 31      CALL 3164       ;WRITE COMMA
328A 01 07 00      LD BC,0007      ;INDEX FOR A REGISTER
328D C3 0B 31      JP 310B         ;CALL REGISTER OUTPUT ROUTINE AND EXIT
```

CHECK FOR OP CODE OF LD (**),HL OR LD (**),A

```
3290 FE 22         CP 22           ;IS IT OP CODE 32 OR 22?
3292 20 30         JR NZ,32C4      ;NO. CONTINUE WITH OP CODE CHECK
3294 06 03         LD B,03         ;THREE OP CODES TO DIPSLAY
3296 CD BE 32      CALL 32BE       ;CALL DISPLAY OP CODES AND MNEMONIC REF
3299 CD AF 32      CALL 32AF       ;CALL WRITE ADDRESS IN BRACKET ROUTINE
329C CD 64 31      CALL 3164       ;ADD COMMA
329F F1            POP AF          ;RESTORE OP CODE
32A0 CB 67         BIT 4,A         ;CHECK IF ITS LD (**),A
32A2 20 03         JR NZ,32A7      ;ITS AN A, SO JUMP BELOW
32A4 C3 CD 35      JP 35CD         ;JUMP TO OUTPUT HL,IX,IY ROUTINE AND EXIT
```

DISPLAY REGISTER A IN OUTPUT STRING

```
32A7 3E 01         LD A,01         ;ONE CHARACTER TO PRINT
32A9 01 07 00      LD BC,0007      ;INDEX FOR A REGISTER
32AC C3 0D 31      JP 310D         ;CALL REGISTER OUTPUT ROUTINE AND EXIT
```

WRITE AN ADDRESS IN BRACKETS. IE "(0900)".

```
32AF 01 08 00       LD BC,0008       ;INDEX OF LEFT BRACKET
32B2 CD 0B 31       CALL 310B        ;CALL THE REGISTER ASCII LOOKUP TABLE
32B5 CD 21 32       CALL 3221        ;WRITE A 16 BIT ADDRESS IN BIG ENDIAN FORMAT
32B8 01 0B 00       LD BC,000B       ;INDEX OF RIGHT BRACKET
32BB C3 0B 31       JP 310B          ;JUMP OUT VIA THE REGISTER LOOKUP TABLE
```

DISPLAY THE OP CODES BASED ON THE NUMBER IN B. MOVE TO NEXT ROW ON OUTPUT STRING. THEN BASED ON OP CODE WRITE MNEMONIC USING THE REFERENCE LOOKUP TABLE.

```
32BE CD FF 31       CALL 31FF        ;CALL DISPLAY OP CODE ROUTINE
32C1 C3 05 32       JP 3205          ;DO MNEMONIC REFERENCING OF OP CODE
```

CHECK FOR OP CODE OF LD HL,(**) OR LD A,(**)

```
32C4 FE 2A          CP 2A            ;IS OP CODE 3A OR 2A?
32C6 20 0E          JR NZ,32D6       ;NO. CONTINUE WITH OP CODE CHECK
32C8 06 03          LD B,03          ;THREE OP CODES TO DIPSLAY
32CA CD BE 32       CALL 32BE        ;CALL DISPLAY OP CODES AND MNEMONIC REF
32CD F1             POP AF           ;RESTORE OP CODE
32CE CD A0 32       CALL 32A0        ;CALL A OR HL TO DISPLAY ABOVE AND RETURN HERE
32D1 CD 64 31       CALL 3164        ;ADD A COMMA
32D4 18 D9          JR 32AF          ;CALL WRITE ADDRESS IN BRACKET ROUTINE AND EXIT
```

CHECK FOR OP CODE OF INC rr

```
32D6 E6 CF          AND CF           ;MASK TO REMOVE HIGH NIBBLE
32D8 FE 03          CP 03            ;IS OP CODE 03,13,23,33?
32DA 20 0F          JR NZ,32EB       ;NO. CONTINUE WITH OP CODE CHECK
32DC CD E3 32       CALL 32E3        ;DISPLAY 'INC' THEN COME BACK HERE!
32DF F1             POP AF           ;RESTORE OP CODE
32E0 C3 0B 32       JP 320B          ;JUMP TO DOUBLE REGISTER OUTPUT AND EXIT
32E3 CD 55 30       CALL 3055        ;WRITE ONE OP CODE OUT
32E6 3E BC          LD A,BC          ;LOAD A WITH INDEX OF 'INC'
32E8 C3 07 32       JP 3207          ;JUMP TO ASCII LOOKUP ROUTINE AND RETURN
```

CHECK FOR OP CODE OF DEC rr

```
32EB FE 0B          CP 0B            ;IS OP CODE 0B,1B,2B,3B?
32ED 20 0C          JR NZ,32FB       ;NO. CONTINUE WITH OP CODE CHECK
32EF CD F6 32       CALL 32F6        ;DISPLAY ONE OP CODE AND 'DEC'. THEN COME BACK HERE!
32F2 F1             POP AF           ;RESTORE OP CODE
32F3 C3 0B 32       JP 320B          ;JUMP TO DOUBLE REGISTER OUTPUT AND EXIT
32F6 3E BF          LD A,BF          ;LOAD A WITH INDEX OF 'DEC'
32F8 C3 11 35       JP 3511          ;JUMP TO SPECIAL ASCII LOOKUP THEN
                                     ;DOESN'T OUTPUT ANYTHING IF A=0! AND THEN EXIT
```

CHECK FOR OP CODE OF INC r

```
32FB E6 C7          AND C7           ;MASK TO REMOVE HIGH NIBBLE AND BIT 3
32FD FE 04          CP 04            ;IS OP CODE 04,14,24,34?
32FF 20 0A          JR NZ,330B       ;NO. CONTINUE WITH OP CODE CHECK
3301 CD E3 32       CALL 32E3        ;CALL DISPLAY 'INC' ABOVE
3304 F1             POP AF           ;RESTORE OP CODE
3305 CD AC 31       CALL 31AC        ;CALL REGISTER OUTPUT CALCULATION ROUTINE
3308 C3 06 31       JP 3106          ;JUMP TO REGISTER LOOKUP BASED OF A,C FROM
                                     ;PREVIOUS ROUTINE. THEN EXIT
```

CHECK FOR OP CODE OF DEC r

```
330B FE 05          CP 05            ;IS OP CODE 05,15,25,35?
330D 20 06          JR NZ,3315       ;NO. CONTINUE WITH OP CODE CHECK
330F CD F6 32       CALL 32F6        ;CALL DISPLAY 'DEC' ABOVE
3312 F1             POP AF           ;RESTORE OP CODE
3313 18 F0          JR 3305          ;JUMP ABOVE TO DISPLAY REGISTER AND EXIT
```

CHECK FOR OP CODE ADD rr,rr

```
3315 79            LD A,C          ;RESTORE OP CODE
3316 E6 CF         AND CF          ;MASK REMOVING BITS 4,5
3318 FE 09         CP 09           ;IS OP CODE 09,19,29,39?
331A 20 0F         JR NZ,332B      ;NO. CONTINUE WITH OP CODE CHECK
331C 3E 86         LD A,86         ;LOAD INDEX OF 'ADD' TO A
331E CD 11 35      CALL 3511       ;JUMP TO SPECIAL ASCII LOOKUP OUTPUT
3321 CD A4 32      CALL 32A4       ;JUMP TO OUTPUT HL,IX,IY ROUTINE
3324 CD 64 31      CALL 3164       ;ADD COMMA
3327 F1            POP AF          ;RESTORE OP CODE
3328 C3 0B 32      JP 320B         ;JUMP TO DOUBLE REGISTER OUTPUT ROUTINE AND EXIT
```

CHECK FOR OP CODE DJNZ *. THE OFFSET WILL BE CONVERTED TO AN ACTUAL ADDRESS

```
332B F1            POP AF          ;RESTORE OP CODE
332C FE 10         CP 10           ;IS OP CODE 10?
332E 20 1B         JR NZ,334B      ;NO. CONTINUE WITH OP CODE CHECK
3330 3E D9         LD A,D9         ;LOAD INDEX OF 'DJNZ' TO A
3332 F5            PUSH AF         ;SAVE OP CODE
3333 06 02         LD B,02         ;TWO BYTES TO DISPLAY
3335 CD FF 31      CALL 31FF       ;CALL DISPLAY OP CODE ROUTINE
3338 F1            POP AF          ;RESTORE OP CODE
3339 CD 07 32      CALL 3207       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
333C C3 F3 35      JP 35F3         ;DO CONVERT RELATIVE OFFSET TO ADDRESS AND EXIT
```

THIS CODE APPEARS NOT TO BE USED. THE FIRST PART INCREMENTS THE END OF STRING ADDRESS. THE SECOND PART WRITES 'AF' DIRECTLY TO THE OUTPUT STRING. I CANT SEEM TO FIND THIS CODE BEING CALLED ANYWHERE. BUT IT FUNCTIONS ARE IN OTHER PARTS OF THE CODE

```
333F 23            INC HL          ;MOVE END OUTPUT ADDRESS TO NEXT ONE
3340 22 A0 08      LD (08A0),HL    ;SAVE IT
3343 C9            RET             ;EXIT

3344 36 41         LD (HL),41      ;WRITE 'A' TO OUTPUT STRING
3346 23            INC HL          ;MOVE TO NEXT ADDRESS
3347 36 46         LD (HL),46      ;WRITE 'F' TO OUTPUT STRING
3349 23            INC HL          ;MOVE TO NEXT ADDRESS
334A C9            RET             ;EXIT
```

CHECK FOR OP CODE JR *. THE OFFSET WILL BE CONVERTED TO AN ACTUAL ADDRESS

```
334B FE 18         CP 18           ;IS OP CODE 18?
334D 20 04         JR NZ,3353      ;NO. CONTINUE WITH OP CODE CHECK
334F 3E D5         LD A,D5         ;LOAD INDEX OF 'JR' TO A
3351 18 DF         JR 3332         ;CALL CODE IN DJNZ ABOVE TO DISPLAY OP CODE,
                                   ;MNEMONIC AND ACTUAL ADDRESS. THEN EXIT
```

CHECK FOR OP CODE JR Z,* OR JR C,*. THE OFFSET WILL BE CONVERTED TO AN ACTUAL ADDRESS AND Z AND C ARE ALSO FOUND.

```
3353 79            LD A,C          ;RESTORE OP CODE
3354 E6 C7         AND C7          ;MASK OUT BITS 3,4,5
3356 B7            OR A            ;IS A ZERO, OR OP CODE 18,28,38
3357 20 1E         JR NZ,3377      ;NO. CONTINUE WITH OP CODE CHECK
3359 79            LD A,C          ;RESTORE OP CODE
335A F5            PUSH AF         ;SAVE OP CODE
335B 06 02         LD B,02         ;TWO OP CODES TO PRINT
335D CD FF 31      CALL 31FF       ;CALL DISPLAY OP CODE ROUTINE
3360 3E D5         LD A,D5         ;LOAD INDEX OF 'JR' TO A
3362 CD 07 32      CALL 3207       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
3365 F1            POP AF          ;RESTORE OP CODE
3366 CD 6C 33      CALL 336C       ;CHECK FOR Z OR C
3369 C3 F3 35      JP 35F3         ;DO CONVERT RELATIVE OFFSET TO ADDRESS AND EXIT
```

FLAG OUTPUT ROUTINE THAT WORKS OUT WHICH FLAG CHECK IS BEING USED. ONE OF NZ,Z,NC,C,PO,PE,P,M

```
336C E6 18         AND 18          ;MASK TO CHECK FOR 28 OR 38
336E 1F            RRA             ;ROTATE BITS RIGHT 3 TIMES
336F 1F            RRA             ;CARRY WILL BE 3
3370 1F            RRA             ;AND ZERO WILL BE 1
3371 87            ADD A,A         ;DOUBLE A TO GET INDEX
3372 C6 DD         ADD A,DD        ;ADD BASE INDEX TO A
3374 C3 07 32      JP 3207         ;JUMP TO ASCII MNEMONIC LOOKUP ROUTINE AND EXIT
```

CHECK FOR OP CODE JP NZ,**

```
3377 79            LD A,C          ;RESTORE OP CODE
3378 FE C3         CP C3           ;IS OP CODE C3?
337A 28 15         JR Z,3391       ;JUMP TO JP ** OP CODE OUTPUT
```

CHECK FOR OP CODE CALL **

```
337C FE CD         CP CD           ;IS OP CODE CD?
337E 28 0D         JR Z,338D       ;JUMP TO CALL ** OP CODE OUTPUT
```

CHECK FOR OP CODE RET. NOTE* RET IS HANDLED IN A PREVIOUS ROUTINE AND IT WON'T GET HERE! THIS IS PROBABLY LEGACY CODE, ALSO THE INDEX REF OF E5 IS MEANINGLESS AND DOENS'T POINT TO ANYTHING USEFUL

```
3380 FE C9         CP C9           ;IS OP CODE C9?
3382 20 23         JR NZ,33A7      ;NO. CONTINUE WITH OP CODE CHECK
3384 3E E5         LD A,E5         ;LOAD INDEX OF 'PO'
3386 C3 11 35      JP 3511         ;OUTPUT OP CODE AND MNEMONIC ROUTINE AND EXIT
```

SET UP TO WORK OUT WHICH FLAG CONDITION TO USE. ONE OF NZ,Z,NC,C,PO,PE,P,M

```
3389 E6 38         AND 38          ;MASK OUT BITS 1,2,3,7,8
338B 18 E1         JR 336E         ;JUMP TO FLAG OUTPUT ROUTINE
```

CALL ** OP CODE OUTPUT WITH ADDRESS

```
338D 3E C8         LD A,C8         ;LOAD INDEX OF 'CALL'
338F 18 02         JR 3393         ;JUMP BELOW TO CONTINUE WITH JR OUTPUT
```

JP ** OP CODE OUTPUT WITH ADDRESS

```
3391 3E CC         LD A,CC         ;LOAD INDEX OF 'JP'
3393 06 03         LD B,03         ;THREE OP CODES TO OUTPUT
3395 F5            PUSH AF         ;SAVE INDEX
3396 CD FF 31      CALL 31FF       ;CALL DISPLAY OP CODE ROUTINE
3399 F1            POP AF          ;RESTORE INDEX
339A FE A6         CP A6           ;IS INDEX A6? JUST DISPLAY MNEMONIC AND NO ADDRESS
339C 20 03         JR NZ,33A1      ;REQUIRES ADDRESS OUTPUT
339E C3 07 32      JP 3207         ;JUMP TO ASCII MNEMONIC LOOKUP ROUTINE AND EXIT
33A1 CD E8 32      CALL 32E8       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
33A4 C3 21 32      JP 3221         ;WRITE 16 BIT ADDRESS IN BIG ENDIAN FORMAT AND EXIT
```

CHECK FOR OP CODE RET NZ, RET NC, RET PO, RET P

```
33A7 E6 C7         AND C7          ;MASK TO CHECK FOR [F-C][7-0] OP CODES
33A9 FE C0         CP C0           ;IS OP CODE C0,D0,E0,F0?
33AB 20 18         JR NZ,33C5      ;NO. CONTINUE WITH OP CODE CHECK
33AD 3E C5         LD A,C5         ;LOAD INDEX OF 'RET'
33AF 06 01         LD B,01         ;ONE OP CODE TO OUTPUT
33B1 CD B8 33      CALL 33B8       ;JUMP BELOW TO HANDLE OUTPUTTING THEN COME BACK
33B4 2B            DEC HL          ;MOVE END OF OUTPUT STRING BACK ONE
33B5 36 20         LD (HL),20      ;PUT AN ASCII SPACE " " THERE (FIX FOR P?)
```

```
33B7 C9            RET             ;EXIT
33B8 C5            PUSH BC         ;SAVE OP CODE
33B9 F5            PUSH AF         ;SAVE INDEX
33BA CD FF 31      CALL 31FF       ;CALL DISPLAY OP CODE ROUTINE
33BD F1            POP AF          ;RESTORE INDEX
33BE CD E8 32      CALL 32E8       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
33C1 C1            POP BC          ;RESTORE OP CODE
33C2 79            LD A,C          ;SAVE OP CODE IN A
33C3 18 C4         JR 3389         ;JUMP TO FLAG OUTPUT ROUTINE AND EXIT
```

CHECK FOR OP CODE CALL NZ,**, CALL NC,**, CALL PO,**, CALL P,**

```
33C5 79            LD A,C          ;RESTORE OP CODE
33C6 E6 C7         AND C7          ;MASK TO CHECK FOR [F-C][7-0] OP CODES
33C8 FE C4         CP C4           ;IS OP CODE C4,D4,E4,F4?
33CA 20 0A         JR NZ,33D6      ;NO. CONTINUE WITH OP CODE CHECK
33CC 3E C8         LD A,C8         ;LOAD INDEX OF 'CALL'
33CE 06 03         LD B,03         ;THREE OP CODES TO DISPLAY
33D0 CD B8 33      CALL 33B8       ;OUTPUT OP CODES AND CALL
33D3 C3 21 32      JP 3221         ;WRITE 16 BIT ADDRESS IN BIG ENDIAN FORMAT AND EXIT
```

CHECK FOR OP CODE JP NZ,**, JP NC,**, JP PO,**, JP P,**

```
33D6 FE C2         CP C2           ;IS OP CODE C2,D2,E2,F2?
33D8 20 04         JR NZ,33DE      ;NO. CONTINUE WITH OP CODE CHECK
33DA 3E CC         LD A,CC         ;LOAD INDEX OF 'JP'
33DC 18 F0         JR 33CE         ;JUMP ABOVE TO CALL ROUTINE AS ITS THE SAME
```

CHECK FOR OP CODE POP rr

```
33DE 79            LD A,C          ;RESTORE OP CODE
33DF E6 CF         AND CF          ;MASK TO CHECK FOR [F-C][F-0] OP CODES
33E1 FE C1         CP C1           ;IS OP CODE C1,D2,E2,F2?
33E3 20 04         JR NZ,33E9      ;NO. CONTINUE WITH OP CODE CHECK
33E5 3E CE         LD A,CE         ;LOAD INDEX OF 'POP'
33E7 18 05         JR 33EE         ;JUMP TO PUSH ROUTINE TO DISPLAY DOUBLE REGISTERS
```

CHECK FOR OP CODE PUSH rr

```
33E9 FE C5         CP C5           ;IS OP CODE C5,D5,E5,F5?
33EB C0            RET NZ          ;IF NOT EXIT (NO MORE OP CODES TO FIND!)
33EC 3E D1         LD A,D1         ;LOAD INDEX OF 'PUSH'
33EE C5            PUSH BC         ;SAVE OP CODE
33EF F5            PUSH AF         ;SAVE GROUP OP CODE
33F0 CD 55 30      CALL 3055       ;WRITE ONE OP CODE OUT AND MOVE TO NEXT ROW
33F3 F1            POP AF          ;RESTORE GROUP OP CODE
33F4 CD E8 32      CALL 32E8       ;JUMP TO ASCII LOOKUP ROUTINE AND RETURN
33F7 C1            POP BC          ;RESTORE OP CODE
33F8 79            LD A,C          ;LOAD IN A
33F9 FE F1         CP F1           ;IS IT POP AF? TEST THIS AS NO LOOKUP AVAILABLE
33FB 28 04         JR Z,3401       ;YES, MOVE TO OUTPUT 'AF'
33FD FE F5         CP F5           ;IS IT PUSH AF?
33FF 20 0A         JR NZ,340B      ;NO
3401 36 41         LD (HL),41      ;WRITE ASCII 'A' IN OUTPUT STRING
3403 23            INC HL          ;MOVE TO NEXT POSITION IN OUTPUT STRING
3404 36 46         LD (HL),46      ;WRITE ASCII 'F' IN OUTPUT STRING
3406 23            INC HL          ;MOVE TO NEXT POSITION IN OUTPUT STRING
3407 22 A0 08      LD (08A0),HL    ;SAVE END OF OUTPUT STRING
340A C9            RET             ;EXIT AS AF IS NOW OUTPUTTED
340B C3 0B 32      JP 320B         ;JUMP TO DOUBLE REGISTER OUTPUT ROUTINE AND EXIT
```

HANDLE THE REST OF THE OPCODES.

CHECK FOR OP CODE RST

```
340E E6 C7         AND C7          ;IS OP CODE A RST?
3410 FE C7         CP C7           ;A RST OP CODE?
3412 20 0D         JR NZ,3421      ;JUMP IF NOT A RST OP CODE
3414 79            LD A,C          ;LOAD OP CODE BACK INTO A
3415 F5            PUSH AF         ;SAVE AF
3416 3E C2         LD A,C2         ;INDEX REFERENCE FOR LOOKUP TABLE
3418 CD 11 35      CALL 3511       ;JUMP TO SPECIAL ASCII LOOKUP THEN
341B F1            POP AF          ;RESTORE AF
341C E6 38         AND 38          ;GET RST NUMBER 00,08,10,18,20,28,30,38
341E C3 91 31      JP 3191         ;CALL CONVERT A TO ASCII ROUTINE AND EXIT

CHECK FOR ADD * ETC..

3421 E6 C6         AND C6          ;IS OP CODE C6,D6,E6,F6,CE,DE,EE,FE
3423 FE C6         CP C6           ;AN ADD,SUB,AND,OR,ADC,SBC,XOR,CP WITH A
3425 79            LD A,C          ;LOAD OP CODE BACK INTO A AS ITS BEEN CLOBBERED
3426 20 18         JR NZ,3440      ;JUMP IF NOT AN ABOVE OP CODE
3428 AF            XOR A           ;CLEAR A
3429 CD F7 34      CALL 34F7       ;CALL OUTPUT OPCODE WITH VARIABLE ROUTINE
342C 79            LD A,C          ;RESTORE OP CODE
342D C5            PUSH BC         ;SAVE BC
342E CD AC 31      CALL 31AC       ;CALL REGISTER INDEX ROUTINE, C=INDEX
3431 3E 86         LD A,86         ;LOAD A WITH BASE INDEX
3433 81            ADD A,C         ;ADD INDEX TO A THREE TIMES TO GET CORRECT POS
3434 81            ADD A,C
3435 81            ADD A,C
3436 CD 07 32      CALL 3207       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
3439 CD 86 30      CALL 3086       ;CHECK IF ONE OR TWO REGISTERS REQUIRED
343C C1            POP BC          ;RESTORE BC
343D C3 4D 32      JP 324D         ;JUMP TO WRITE NEXT OPCODE TO ASCII AND EXIT

HANLDE EXTENDED OP CODES

3440 FE ED         CP ED           ;IS OP CODE AND EXTENDED CODE?
3442 C2 E8 34      JP NZ,34E8      ;JUMP IF NOT AN EXTENDED CODE
3445 2A 98 08      LD HL,(0898)    ;LOAD HL TO START ADDRESS (ED)
3448 23            INC HL          ;MOVE TO SECOND OP CODE
3449 7E            LD A,(HL)       ;STORE IT IN A
344A 4F            LD C,A          ;AND IN C
344B E6 C7         AND C7          ;IS IT AN LD (**),rr OR LD rr,(**)
344D FE 43         CP 43           ;
344F 20 20         JR NZ,3471      ;ITS NOT SO CONTINUE WITH OTHER 'ED' COMMANDS

OP CODE IS LD (**),rr OR LD rr,(**)

3451 06 04         LD B,04         ;4 OP CODES TO DISPLAY: IE: ED 63 ** **
3453 CD FF 31      CALL 31FF       ;CALL DISPLAY OP CODE ROUTINE
3456 CD 05 32      CALL 3205       ;DISPLAY "LD " MNEMONIC
3459 CB 59         BIT 3,C         ;CHECK IF ITS A LD rr,(**), IE: 4B,5B,6B,7B
345B 20 0B         JR NZ,3468      ;JUMP IF ITS A LD rr,(**)
345D C5            PUSH BC         ;SAVE BC
345E CD AF 32      CALL 32AF       ;WRITE ADDRESS WITH BRACKETS (**) TO OUTPUT STRING
3461 CD 64 31      CALL 3164       ;ADD A COMMA TO OUTPUT
3464 C1            POP BC          ;RESTORE BC
3465 C3 0A 32      JP 320A         ;JUMP TO DOUBLE REGISTER LOOKUP AND EXIT

OP CODE IS A LD rr,(**)

3468 CD 0A 32      CALL 320A       ;CALL DOUBLE REGISTER LOOKUP AND RETURN HERE
346B CD 64 31      CALL 3164       ;ADD A COMMAN TO OUTPUT
346E C3 AF 32      JP 32AF         ;WRITE ADDRESS WITH BRACKETS (**) TO OUTPUT STRING
                                   ;AND EXIT

                                   ;OTHER EXTENDED CODE COMMANDS
3471 06 02         LD B,02         ;TWO OP CODES TO SHOW
```

```
3473 18 6A         JR 34DF         ;CHECK IF OP CODE USES REGISTERS? EXCEPT I AND R
                                   ;IF TRUE CALL DISPLAY OP CODE ROUTINE AND
                                   ;COME BACK HERE, ELSE DO MNEMONIC LOOKUP
```

WORK OUT IF EXTENDED COMMAND IS IN,OUT OR SBC/ADC

```
3475 79            LD A,C          ;RESET A WITH CURRENT OP CODE
3476 E6 C7         AND C7          ;WORK OUT WHICH REGISTERS ARE USED
3478 FE 40         CP 40           ;IS IT 'IN'?
347A 20 1A         JR NZ,3496      ;JUMP TO CHECK IF IT'S 'OUT'
347C C5            PUSH BC         ;SAVE BC
347D 3E F0         LD A,F0         ;LOAD TABLE INDEX IN A (IN)
347F CD 07 32      CALL 3207       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
3482 C1            POP BC          ;RESTORE BC
3483 79            LD A,C          ;PUT ORIGINAL OP CODE IN A
3484 CD AC 31      CALL 31AC       ;CALL REGISTER MNEMONIC CALCULATOR ROUTINE
3487 79            LD A,C          ;C IS NOW THE INDEX INTO THE REGISTER TABLE
3488 CD 0B 31      CALL 310B       ;CALL REGISTER LOOKUP ROUTINE WITH 1 CHAR TO PRINT
348B CD 64 31      CALL 3164       ;ADD A COMMA TO OUTPUT
348E 01 22 00      LD BC,0022      ;REGISTER INDEX FOR '(C)'
3491 3E 03         LD A,03         ;NUMBER OF CHARACTERS TO PRINT
3493 C3 0D 31      JP 310D         ;CALL REGISTER LOOKUP ROUTINE

3496 FE 41         CP 41           ;IS IT 'OUT'?
3498 20 15         JR NZ,34AF      ;NO THEN JUMP
349A C5            PUSH BC         ;SAVE BC
349B 3E ED         LD A,ED         ;LOAD TABLE INDEX IN A (OUT)
349D CD 07 32      CALL 3207       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
34A0 CD 8E 34      CALL 348E       ;PRINT '(C)' TO OUTPUT (CALLS VIA IN ROUTINE ABOVE)
34A3 CD 64 31      CALL 3164       ;ADD A COMMA TO OUTPUT
34A6 C1            POP BC          ;RESTORE BC
34A7 79            LD A,C          ;PUT ORIGINAL OP CODE IN A
34A8 CD AC 31      CALL 31AC       ;CALL REGISTER MNEMONIC CALCULATOR ROUTINE
34AB 79            LD A,C          ;C IS NOW THE INDEX INTO THE REGISTER TABLE
34AC C3 0B 31      JP 310B         ;CALL REGISTER LOOKUP ROUTINE WITH 1 CHAR TO PRINT
                                   ;AND EXIT

34AF FE 42         CP 42           ;IS IT 'SBC' OR 'ADC'
34B1 20 35         JR NZ,34E8      ;NO THEN JUMP
34B3 C5            PUSH BC         ;SAVE BC
34B4 79            LD A,C          ;LOAD ORIGINAL OP CODE TO A
34B5 CB 5F         BIT 3,A         ;CHECK IF ITS ADC
34B7 20 07         JR NZ,34C0      ;IT IS ADC SO JUMP
34B9 3E 8F         LD A,8F         ;SBC INDEX REFERENCE
34BB CD 07 32      CALL 3207       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
34BE 18 05         JR 34C5         ;SKIP TO REGISTERS
34C0 3E 89         LD A,89         ;ADC INDEX REFERENCE
34C2 CD 07 32      CALL 3207       ;CALL ASCII MNEMONIC LOOKUP ROUTINE
34C5 CD A4 32      CALL 32A4       ;CALL HL,IX,IY OUTPUT ROUTINE
34C8 CD 64 31      CALL 3164       ;ADD A COMMA TO OUTPUT
34CB C1            POP BC          ;RESTORE BC
34CC C3 0A 32      JP 320A         ;CALL DOUBLE REGISTER LOOKUP AND EXIT
```

EXTENDED FUNCTION MNEMONIC LOOKUP ROUTINE. IF ENTERED AT 34CF MOVE OP CODE TO THE NEXT ONE.

```
34CF 06 1C         LD B,1C         ;28 OP CODES TO CHECK
34D1 2A 98 08      LD HL,(0898)    ;PUT START ADDRESS IN HL
34D4 23            INC HL          ;MOVE TO NEXT ADDRESS
34D5 7E            LD A,(HL)       ;STORE OP CODE IN A
34D6 21 F8 36      LD HL,36F8      ;EXTENDED INSTRUCTION LOOKUP TABLE ADDRESS
34D9 06 1C         LD B,1C         ;28 OP CODES TO CHECK
34DB 16 02         LD D,02         ;NUMBER OF OP CODES TO DISPLAY
34DD 18 43         JR 3522         ;JUMP TO OP CODE MATCHING ROUTINE
```

CHECK IF OP CODE USES REGISTERS

```
34DF E6 84         AND 84          ;DOES THE OP CODE HAVE REGISTERS?
34E1 20 EC         JR NZ,34CF      ;JUMP IF IT DOESN'T TO MNEMONIC LOOK UP ROUTINE
34E3 CD FF 31      CALL 31FF       ;CALL DISPLAY OP CODE ROUTINE
34E6 18 8D         JR 3475         ;JUMP TO REGISTER PRINTING ROUTINE

                                   ;OUT (*),A CODE HANDLER
34E8 FE D3         CP D3           ;IS OP CODE OUT (*),A?
34EA 20 16         JR NZ,3502      ;JUMP IF NOT
34EC 3E ED         LD A,ED         ;LOAD A INDEX FOR 'OUT'
34EE CD F7 34      CALL 34F7       ;CALL OUTPUT TWO OP CODES AND MNEMONIC
34F1 CD FB 34      CALL 34FB       ;OUTPUT SECOND OP CODE
34F4 C3 87 32      JP 3287         ;WRITE COMMA AND REGISTER A THEN EXIT.
```

OUTPUTS ONE OP CODE AND ONE VARIABLE TO OUTPUT STRING

```
34F7 06 02         LD B,02         ;TWO DIGITS TO OUTPUT
34F9 18 18         JR 3513         ;CALL OP CODE TO OUTPUT STRING
34FB 2A 98 08      LD HL,(0898)    ;LOAD VALUE AT ADDRESS
34FE 7E            LD A,(HL)       ;TO A
34FF C3 91 31      JP 3191         ;CONVERT A TO ASCII AND OUTPUT

                                   ;IN A,(*) CODE HANDLER
3502 FE DB         CP DB           ;IS OP CODE IN A,(*)?
3504 20 45         JR NZ,354B      ;JUMP IF NOT
3506 3E F0         LD A,F0         ;LOAD A INDEX FOR 'IN'
3508 CD F7 34      CALL 34F7       ;CALL OUTPUT TWO OP CODES AND MNEMONIC
350B 0E 00         LD C,00         ;RESET C
350D C5            PUSH BC         ;SAVE BC
350E C3 39 34      JP 3439         ;JUMP TO A,(*) OUTPUT ROUTINE AND EXIT
```

THIS ROUTINE WRITES ON OP CODE AND DOES AN ASCII OUTPUT ONLY IF THE INDEX OF A IS NOT ZERO. IF A IS NON ZERO, THE MNEMONIC IS STILL TO BE OUTPUTTED.

```
3511 06 01         LD B,01         ;ONE BYTE TO OUTPUT
3513 F5            PUSH AF         ;SAVE AF
3514 CD FF 31      CALL 31FF       ;CALL OUTPUT OP CODE TO STRING ROUTINE
3517 F1            POP AF          ;RESTORE AF
3518 B7            OR A            ;CHECK FOR A=0
3519 C2 07 32      JP NZ,3207      ;IF NOT CONTINUE WITH MNEMONIC LOOKUP
351C C9            RET             ;ELSE DO NOTHING AND RETURN.
```

ONE BYTE AND EXTENDED INSTRUCTION OP CODE MATCHING. THESE OP CODES DON'T HAVE ANY VARIABLES. THIS ROUTINE COMPARES THE CURRENT OP CODE TO THE TABLE OP CODES. IF FOUND THEN OUTPUT IT TO THE OUTPUT STRING

```
351D 06 13         LD B,13         ;NUMBER OP CODES TO CHECK
351F 21 90 36      LD HL,3690      ;LOAD HL WITH OP LOOKUP TABLE ADDRESS
3522 BE            CP (HL)         ;IS CURRENT OP IN LOOK UP TABLE?
3523 28 0A         JR Z,352F       ;IF TRUE JUMP TO 352F
3525 23            INC HL          ;MOVE TO NEXT OP
3526 CB 7E         BIT 7,(HL)      ;IF NOT ON LAST ASCII
3528 28 FB         JR Z,3525       ;CHECK NEXT ASCII CHARACTER
352A 23            INC HL          ;ON LAST CHARACTER MOVE TO NEXT ONE
352B 10 F5         DJNZ 3522       ;REPEAT FOR ALL 19 OP CODES
352D 37            SCF             ;SET CARRY FLAG TO INDICATE NO OP CODE FOUND
352E C9            RET             ;EXIT
```

DISPLAY CURRENT OP CODES MNEMONICS IN ASCII. DISPLAY THE OP CODE ON THE FIRST LINE. THEN MOVE TO THE SECOND LINE AND KEEP ITERATING THROUGH THE LOOK UP TABLE WRITING THE ACSII TO THE OUTPUT STRING UNTIL THE LAST CHARACTER IS WRITTEN. THIS IS NOTIFIED BY BIT 7 SET. THEN REMOVE THE BIT FROM THE LAST

CHARACTER AND EXIT RESETTING CARRY FLAG.

```
352F E5             PUSH HL          ;SAVE LOOKUP TABLE ADDRESS
3530 42             LD B,D           ;NUMBER OF OP CODES TO DISPLAY SAVED IN B
3531 CD FF 31       CALL 31FF        ;CALL DISPLAY OP CODE ROUTINE
3534 E1             POP HL           ;RESTORE LOOKUP TABLE
3535 ED 5B A0 08    LD DE,(08A0)     ;LOAD DE WITH END OF OUTPUT STRING ADDRESS
3539 23             INC HL           ;MOVE TO NEXT CHARACTER IN LOOKUP TABLE
353A 7E             LD A,(HL)        ;LOAD ASCII CHARACTER TO A
353B 12             LD (DE),A        ;SAVE A IN THE OUTPUT STRING
353C 13             INC DE           ;MOVE TO NEXT ENTRY IN OUTPUT STRING
353D CB 7F          BIT 7,A          ;IS ASCII CHARACTER THE LAST ONE (BIT 7 SET?)
353F 28 F8          JR Z,3539        ;NO, THEN GET NEXT CHARACTER
3541 EB             EX DE,HL         ;LAST CHARACTER HAS BIT 7 SET,
3542 2B             DEC HL           ;REMOVE IT FROM
3543 CB BE          RES 7,(HL)       ;OUTPUT STRING
3545 23             INC HL           ;ON LAST CHARACTER MOVE TO NEXT ONE
3546 22 A0 08       LD (08A0),HL     ;SAVE HL TO END OF OUTPUT STRING ADDRESS
3549 B7             OR A             ;RESETS CARRY FLAG AS OP WAS FOUND
354A C9             RET              ;EXIT
```

LD (IX/IY+*),*, DEC (IX/IY+*), INC (IX/IY+*), JP (IX/IY), EX (SP),IX/IY OP CODE HANDLER. WHICH ARE IX AND IY OP CODES. THEY HAVE SPECIFIC OP CODE OUTPUTS

```
                                     ;IX HANDLER
354B 2A A0 08       LD HL,(08A0)     ;LOAD HL WITH OUTPUT STRING END
354E FE DD          CP DD            ;IS OP CODE IX?
3550 20 07          JR NZ,3559       ;JUMP IF NOT
3552 36 44          LD (HL),44       ;WRITE 'D' TO OUTPUT STRING
3554 23             INC HL           ;MOVE TO NEXT ADDRESS
3555 3E 11          LD A,11          ;LOAD 11 IN A
3557 18 09          JR 3562          ;CONTINUE WITH IY HANDLER

                                     ;IY HANDLER
3559 FE FD          CP FD            ;IS OP CODE IY?
355B 20 33          JR NZ,3590       ;JUMP IF NOT
355D 36 46          LD (HL),46       ;LOAD 'F' TO OUTPUT STRING
355F 23             INC HL           ;MOVE TO NEXT ADDRESS
3560 3E 22          LD A,22          ;LOAD 22 IN A
3562 32 A2 08       LD (08A2),A      ;SAVE A TO HL/IX/IY FLAG
3565 36 44          LD (HL),44       ;WRITE 'D' TO OUTPUT STRING
3567 23             INC HL           ;MOVE TO NEXT ADDRESS
3568 23             INC HL           ;PUT SPACE IN OUTPUT ADDRESS
3569 22 A0 08       LD (08A0),HL     ;SAVE END OF OUTPUT ADDRESS
356C 2A 98 08       LD HL,(0898)     ;LOAD CURRENT ADDRESS TO HL
356F 23             INC HL           ;MOVE TO SECOND OP CODE
3570 4E             LD C,(HL)        ;SAVE IT IN C
3571 7E             LD A,(HL)        ;SAVE IT IN A
3572 22 98 08       LD (0898),HL     ;UPDATE CURRENT ADDRESS
3575 FE 36          CP 36            ;IS OP CODE FD/DD 36?
3577 E5             PUSH HL          ;SAVE HL
3578 CC FD 31       CALL Z,31FD      ;IF ITS 36 THEN 3 OP CODES TO PRINT
357B 79             LD A,C           ;LOAD OP CODE BACK TO A
357C E6 FE          AND FE           ;MASK OUT BIT 1
357E FE 34          CP 34            ;HAS IT GOT 2 OP CODES?
3580 06 02          LD B,02          ;TO OP CODES TO PRINT
3582 CC FF 31       CALL Z,31FF      ;PRINT TWO OP CODES
3585 E1             POP HL           ;RESTORE HL
3586 7E             LD A,(HL)        ;LOAD OP CODE IN A
3587 FE E9          CP E9            ;IS IT JP (IX) OR JP (IY)?
3589 28 02          JR Z,358D        ;CALL JP HANDLER
358B FE E3          CP E3            ;IS IT EX (SP),IX OR EX (SP),IY?
358D 28 7B          JR Z,360A        ;CALL EX/JP MOD HANDLER
358F 79             LD A,C           ;LOAD OP IN A
```

COME HERE IF NO SPECIFIC OP CODE IS FOUND

```
3590 C3 E9 31      JP 31E9         ;JUMP TO OP CODE COUNT CHECK
```

```
3593 2A 98 08      LD HL,(0898)    ;LOAD HL WITH THE CURRENT START ADDRESS
3596 2B            DEC HL          ;DECREASE IT TWICE
3597 2B            DEC HL
3598 22 98 08      LD (0898),HL    ;STORE IT BACK IN START ADDRESS
359B CD 06 31      CALL 3106       ;CALL REGISTER OUTPUT ROUTINE
```

END IF CURRENT OP CODE. INCREASE START ADDRESS BY ONE FOR NEXT ITERATION

```
359E 2A 98 08      LD HL,(0898)    ;LOAD CURRENT ADDRESS TO HL
35A1 23            INC HL          ;MOVE TO NEXT ADDRESS
35A2 22 98 08      LD (0898),HL    ;STORE HL IN START ADDRESS
35A5 C9            RET             ;EXIT
```

CALCULATE INDEX FOR HL, IX, IY DOUBLE REGISTERS. CHECK THE HL/IX/IY FLAG
TO SET THE CORRECT INDEX FOR THE DOUBLE REGISTER LOOKUP

```
35A6 1F            RRA             ;ROTATE RIGHT 3 TIMES TO GET INDEX
35A7 1F            RRA
35A8 1F            RRA
35A9 FE 04         CP 04           ;IS IT A HL,IX,IY?
35AB C0            RET NZ          ;RETURN IF NOT
35AC 3A A2 08      LD A,(08A2)     ;CHECK HL/IX/IY FLAG
35AF 0F            RRCA            ;SEE IF IX
35B0 30 03         JR NC,35B5      ;NO THEN CONTINUE
35B2 3E F3         LD A,F3         ;LOAD A WITH 'IX' INDEX
35B4 C9            RET             ;EXIT
35B5 0F            RRCA            ;SEE IF IY
35B6 30 03         JR NC,35BB      ;NO THEN CONTINUE
35B8 3E FA         LD A,FA         ;LOAD A WITH 'IY' INDEX
35BA C9            RET             ;EXIT
35BB 3E 04         LD A,04         ;MUST BE HL SO LOAD INDEX 04 BACK.
35BD C9            RET             ;EXIT
```

ENTRY POINT TO PLACE BRACKETS () AROUND HL,IX, OR IY (MIGHT NOT BE USED)

```
35BE 01 08 00      LD BC,0008      ;INDEX FOR LEFT BRACKET
35C1 CD 0B 31      CALL 310B       ;CALL REGISTER LOOKUP AND OUTPUT LEFT BRACKET
35C4 CD CD 35      CALL 35CD       ;CALL HL,IX,IY OUTPUT
35C7 01 0B 00      LD BC,000B      ;INDEX FOR RIGHT BRACKET
35CA C3 0B 31      JP 310B         ;CALL REGISTER LOOKUP AND OUTPUT RIGHT BRACKET
```

OUTPUT HL,IX OR IY DEPENDING ON THE FLAG SET IN 08A2

```
35CD 3A A2 08      LD A,(08A2)     ;LOAD A WITH HL/IX/IY FLAG
35D0 06 00         LD B,00         ;RESET B
35D2 0F            RRCA            ;SEE IF BIT 0 IS SET (IX)
35D3 30 04         JR NC,35D9      ;NOT SET CHECK IY
35D5 0E 0D         LD C,0D         ;IX REFERENCE
35D7 18 09         JR 35E2         ;JUMP TO REGISTER LOOKUP
35D9 0F            RRCA            ;SEE IF BIT 1 IS SET (IY)
35DA 30 04         JR NC,35E0      ;MUST BE HL
35DC 0E 14         LD C,14         ;IY REFERENCE
35DE 18 02         JR 35E2         ;JUMP TO REGISTER LOOKUP
35E0 0E 04         LD C,04         ;HL REFERENCE
35E2 3E 02         LD A,02         ;2 CHARACTERS TO PRINT
35E4 C3 0D 31      JP 310D         ;CALL REGISTER LOOKUP ROUTINE
```

CHECK IF REGISTER IS AN IX, OR IY, IF SO DISPLAY IT, OR JUMP TO STANDARD
REGISTER OUTPUT ROUTINE

```
35E7 3A A2 08      LD A,(08A2)     ;LOAD HL,IX,IY FLAG
```

```
35EA 0F            RRCA             ;IS BIT 1 IX SET?
35EB 38 A6         JR C,3593        ;JUMP IF IX
35ED 0F            RRCA             ;IS BIT 2 IY SET?
35EE 38 FB         JR C,35EB        ;JUMP IF IY
35F0 C3 06 31      JP 3106          ;JUMP TO REGISTER OUTPUT ROUTINE AND EXIT
```

RELATIVE OFFSET CALCULATOR. THIS ROUTINE WORKS OUT THE CORRECT JUMP ADDRESS FROM A RELATIVE OFFSET. IE '0900 DJNZ 05', MEANS JUMP 5 BYTES FORWARD. THIS ROUTINE WILL DISPLAY 'DJNZ 0905'

```
35F3 2A 98 08      LD HL,(0898)     ;GET CURRENT OP CODE ADDRESS
35F6 5E            LD E,(HL)        ;LOAD RELATIVE OFFSET BYTE INTO E
35F7 AF            XOR A            ;CLEAR A
35F8 CB 7B         BIT 7,E          ;IS IT A FORWARD OR BACKWARD JUMP?
35FA 28 01         JR Z,35FD        ;FORWARD THEN JUMP
35FC 2F            CPL              ;DO A ONE COMPLIMENT AS ITS A BACKWARD JUMP
35FD 57            LD D,A           ;LOAD 0 IN D
35FE 23            INC HL           ;MOVE TO NEXT ADDRESS
35FF 19            ADD HL,DE        ;ADD OFFSET TO HL
3600 C3 8A 31      JP 318A          ;JUMP TO DISPLAY HL TO ASCII AND EXIT.
```

SWAP HL WITH AF AND JUMP TO OUTPUT DOUBLE REGISTER IN BRACKET ROUTINE

```
3603 E1            POP HL           ;RESTORE HL
3604 F1            POP AF           ;RESTORE AF
3605 E5            PUSH HL          ;SAVE HL
3606 F5            PUSH AF          ;SAVE AF
3607 C3 6B 32      JP 326B          ;JUMP TO DOUBLE REGISTER IN BRACKET ROUTINE
```

HANDLER FOR JP (IX), JP (IY), EX (SP),IX, EX (SP),IY. THIS ROUTINE USES THE EXISTING UNIQUE ONE BYTE OP CODE LOOKUP TABLE. BUT REPLACES HL WITH IX/IY. IE: EX (SP),HL WITH EX (SP),IX AND JP (HL) WITH JP (IX)

```
360A 79            LD A,C           ;LOAD SECOND OP CODE TO A
360B CD 1D 35      CALL 351D        ;CALL UNIQUE OP CODE TO ASCII LOOKUP
360E 2B            DEC HL           ;MOVE OUTPUT STRING BACK ONE
360F 7E            LD A,(HL)        ;LOAD LAST ASCII CHARACTER IN A
3610 FE 48         CP 48            ;IS IT A 'H'
3612 20 FA         JR NZ,360E       ;NO, MOVE BACK ONE CHARACTER AND CHECK AGAIN
3614 36 49         LD (HL),49       ;REPLACE 'H' WITH 'I'
3616 23            INC HL           ;MOVE TO NEXT ASCII CHARACTER
3617 36 58         LD (HL),58       ;REPLACE 'L' WITH 'X'
3619 3A A2 08      LD A,(08A2)      ;GET HL/IX/IY FLAG
361C 0F            RRCA             ;CHECK IF ITS IX
361D D8            RET C            ;EXIT IF IX
361E 34            INC (HL)         ;CHANGE 'X' TO 'Y'
361F C9            RET              ;EXIT
```

OP CODE LOOKUP TABLES

OP CODES ARE IN ASCII AT VARIOUS INDEXING POSITIONS. IE: THE BASE INDEX FOR ADDRESS 3621 IS 83. THE END OF THE OP CODE HAS BIT 7 SET.

```
3620 49            ;I       82 < WHERE '82H' IS THE INDEX HERE
3621 4C C4 00      ;LD      83
3624 41 44 C4      ;ADD     86
3627 41 44 C3      ;ADC     89
362A 53 55 C2      ;SUB     8C
362D 53 42 C3      ;SBC     8F
3630 41 4E C4      ;AND     92
3633 58 4F D2      ;XOR     95
3636 4F D2 00      ;OR      98
3639 43 D0 00      ;CP      9B
363C 52 4C C3      ;RLC     9E
```

```
363F 52 52 C3      ;RRC    A1
3642 52 CC 00      ;RL     A4
3645 52 D2 00      ;RR     A7
3648 53 4C C1      ;SLA    AA
364B 53 52 C1      ;SRA    AD
364E 53 52 CC      ;SRL    B0
3651 42 49 D4      ;BIT    B3
3654 52 45 D3      ;RES    B6
3657 53 45 D4      ;SET    B9
365A 49 4E C3      ;INC    BC
365D 44 45 C3      ;DEC    BF
3660 52 53 D4      ;RST    C2
3663 52 45 D4      ;RET    C5
3666 43 41 4C CC   ;CALL   C8
366A 4A D0         ;JP     CC
366C 50 4F D0      ;POP    CE
366F 50 55 53 C8   ;PUSH   D1
3673 4A D2         ;JR     D5
3675 45 D8         ;EX     D7
3677 44 4A 4E DA   ;DJNZ   D9

367B 4E DA         ;NZ     DD
367D DA 20         ;Z_     DF
367F 4E C3         ;NC     E1
3681 C3 00         ;C      E3
3683 50 CF         ;PO     E5
3685 50 C5         ;PE     E7
3687 D0 00         ;P      E9
3689 CD 00         ;M      EB
368B 4F 55 D4      ;OUT    ED
368E 49 CE         ;IN     F0
```

OP CODE REFERENCE TABLES

FOR EACH OP CODE ENTRY THE FIRST BYTE IS THE OP CODE ITSELF, THEN THE MNEMONIC IN ASCII. BIT 8 IS SET ON THE LAST ASCII CHARACTER TO INDICATE ITS THE LAST CHARACTER

UNIQUE ONE BYTE WITH NO VARIABLE OPCODES

```
     OP MNEMONIC IN ASCII             ONE BYTE CODE
     -- -----------------             -------------
3690 00 4E 4F D0                      ;NOP
3694 07 52 4C 43 C1                   ;RLCA
3699 08 45 58 20 41 46 2C 41 46 A7    ;EX AF,AF'
36A3 0F 52 52 43 C1                   ;RRCA
36A8 17 52 4C C1                      ;RLA
36AC 1F 52 52 C1                      ;RRA
36B0 27 44 41 C1                      ;DAA
36B4 2F 43 50 CC                      ;CPL
36B8 37 53 43 C6                      ;SCF
36BC 3F 43 43 C6                      ;CCF
36C0 76 48 41 4C D4                   ;HALT
36C5 C9 52 45 D4                      ;RET
36C9 D9 45 58 D8                      ;EXX
36CD E3 45 58 20 28 53 50 29 2C 48 CC ;EX (SP),HL
36D8 E9 4A 50 20 28 48 4C A9          ;JP (HL)
36E0 EB 45 58 20 44 45 2C 48 CC       ;EX DE,HL
36E9 F3 44 C9                         ;DI
36EC F9 4C 44 20 53 50 2C 48 CC       ;LD SP,HL
36F5 FB 45 C9                         ;EI
```

EXTENDED INSTRUCTION. THESE HAVE OP CODE OF 'ED' THEN A SECONDARY OP CODE. THE LOOKUP USES THE SECONDARY CODE FOR ITS REFERENCE LIKE THE ABOVE TABLE

```
     OP MNEMONIC IN ASCII          EXTENDED CODE
     -- -----------------          -------------
36F8 44 4E 45 C7                   ;NEG
36FC 45 52 45 54 CE                ;RETN
3701 46 49 4D 20 B0                ;IM 0
3706 47 4C 44 20 49 2C C1          ;LD I,A
370D 4D 52 45 54 C9                ;RETI
3712 4F 4C 44 20 52 2C C1          ;LD R,A
3719 56 49 4D 20 B1                ;IM 1
371E 57 4C 44 20 41 2C C9          ;LD A,I
3725 5E 49 4D 20 B2                ;IM 2
372A 5F 4C 44 20 41 2C D2          ;LD A,R
3731 67 52 52 C4                   ;RRD
3735 6F 52 4C C4                   ;RLD
3739 A0 4C 44 C9                   ;LDI
373D A1 43 50 C9                   ;CPI
3741 A2 49 4E C9                   ;INI
3745 A3 4F 55 54 C9                ;OUTI
374A A8 4C 44 C4                   ;LDD
374E A9 43 50 C4                   ;CPD
3752 AA 49 4E C4                   ;IND
3756 AB 4F 55 54 C4                ;OUTD
375B B0 4C 44 49 D2                ;LDIR
3760 B1 43 50 49 D2                ;CPIR
3765 B2 49 4E 49 D2                ;INIR
376A B3 4F 54 49 D2                ;OTIR
376F B8 4C 44 44 D2                ;LDDR
3774 B9 43 50 44 D2                ;CPDR
3779 BA 49 43 44 D2                ;INDR
377E BB 4F 54 44 D2                ;OTDR

REGISTER COMBINATIONS MNEMONICS

3783 42 43                         ;BC
3785 44 45                         ;DE
3787 48 4C                         ;HL
3789 2C 41                         ;,A
378B 28 48 4C 29                   ;(HL)
378F 28 49 58 2B 20 1F 29          ;(IX+__)
3796 28 49 59 2B 20 1F 29          ;(IY+__)

379D 42                            ;B
379E 43                            ;C
379F 44                            ;D
37A0 45                            ;E
37A1 48                            ;H
37A2 4C                            ;L
37A3 53                            ;S
37A4 50                            ;P
37A5 28 43 29                      ;(C)

37A8 FF FF FF FF FF FF FF FF       ;FILL

ENTRY POINT INTO THE DISASSEMBLER WHEN USING THE JMON PERIMETER HANDLER

37B0 21 EB 37      LD HL,37EB      ;SOURCE DATA FOR JMON PERIMETER HANDLER
37B3 11 80 08      LD DE,0880      ;DESTINATION OF COMMAND STRING
37B6 01 0A 00      LD BC,000A      ;COMMAND STRING LENGTH 10 BITES
37B9 ED B0         LDIR            ;COPY SOURCE TO DESTINATION
37BB C3 44 00      JP 0044         ;CALL THE JMON PERIMETER HANDLER
                                   ;USER THEN ENTERS START AND END ADDRESS
                                   ;THEN PRESSES GO WHICH RETURNS BACK TO
                                   ;HERE
37BE CD 00 30      CALL 3000       ;RUN THE DISASSEMBLER
```

LDC OUTPUT ROUTINE. IT UPDATES THE LCD BY WRITING THE OUTPUT ASCII STRING IN TWO ROWS. THE TOP ROW IS THE FIRST 16 CHARACTERS OF THE OUTPUT STRING AND THE SECOND IS THE NEXT + 3 WITH TEXT PRINTED 3 COLUMNS IN. THE ROUTINE THEN HALTS THE CPU. ONCE A KEY IS PRESSED, RUNS THE DISASSEMBLER AGAIN. IT USES THE JMON RST 30 WHICH CHECK FOR THE LCD BUSY FLAG BEFORE DOING THE NEXT UPDATE

```
37C1 F7            RST 30          ;CHECK FOR LCD BUSY
37C2 3E 01         LD A,01         ;CLEAR DISPLAY CODE
37C4 D3 04         OUT (04),A      ;UPDATE LCD INSTRUCTION REGISTER
37C6 21 C0 08      LD HL,08C0      ;LOAD HL WITH OUTPUT STRING LOCATION
37C9 06 10         LD B,10         ;16 CHARACTERS LONG
37CB F7            RST 30          ;CHECK FOR LCD BUSY
37CC 7E            LD A,(HL)       ;LOAD CONTENTS OF HL TO A
37CD D3 84         OUT (84),A      ;SEND IT TO LCD (FIRST ROW BY DEFAULT)
37CF 23            INC HL          ;NEXT CHARACTER (LCD AUTO MOVES TO NEXT CHAR)
37D0 10 F9         DJNZ 37CB       ;DO THIS 16 TIMES
37D2 F7            RST 30          ;CHECK FOR LCD BUSY
37D3 3E C3         LD A,C3         ;SELECT ROW 2 COLUMN 3
37D5 D3 04         OUT (04),A      ;FOR START ENTRY OF LCD
37D7 06 10         LD B,10         ;16 CHARACTERS LONG
37D9 21 D2 08      LD HL,08D2      ;SET HL TO BOTTOM ROW OF OUTPUT STRING
37DC F7            RST 30          ;CHECK FOR LCD BUSY
37DD 7E            LD A,(HL)       ;LOAD CONTENTS OF HL TO A
37DE D3 84         OUT (84),A      ;SEND IT TO LCD (LCD AUTO MOVES TO NEXT CHAR)
37E0 23            INC HL          ;NEXT CHARACTER
37E1 10 F9         DJNZ 37DC       ;DO THIS 16 TIMES
37E3 76            HALT            ;WAIT FOR KEYBOARD INPUT
37E4 18 D8         JR 37BE         ;RUN THE DISASSEMBLER AGAIN!
```

THIS CODE BELOW ISN'T USED BUT CHECK THE LCD BUSY FLAG. THE JR NZ REFERENCE IS INCORRECT AND SHOULD BE FA. IT HAS BEEN REPLACED WITH RST 30 ON JMON. NOTES SAY THAT YOU DON'T NEED JMON FOR THE DISASSEMBLER TO WORK, MAYBE THIS ENTRY IS THERE FOR NON JMON USE???

```
37E6 DB 04         IN A,(04)       ;READ STATUS OF LCD
37E8 CB 7F         BIT 7,A         ;IF BIT 7 IS SET
37EA 20 FB         JR NZ,37E7      ;LCD IS STILL BUSY, TRY AGAIN
37EC C9            RET             ;EXIT
```

PERIMETER COMMAND STRING FOR START AND END ADDRESS. BYTES 1&2 ARE SIGNATURE BYTES (OPTIONAL), BYTES 3&4 IS ADDRESS OF THE DISPLAY CODES 37F7, BYTES 5&6 IS THE INPUT WINDOW POINTER SET AT 0899 (HIGH ORDER BYTE), BYTE 7 IS THE FIRST DISPLAY CODE, BYTE 8 IS THE SIZE OF THE DISPLAY CODES, BYTES 9&10 IS THE JUMP ADDRESS WHEN GO IS PRESSED. JUMPS TO 37BE

NOTE:
THE FIRST TWO BYTES ARE A HANGOVER FROM THE PREVIOUS UNUSED ROUTINE. IT DOESN'T MATTER WHAT THEY ARE AS THEY ARE NOT USED BY THE HANDLER, JUST THAT IT NEEDS TWO BYTES TO START WITH...

```
37EB FB C9 F7 37 99 08 00 01 BE 37
```

RESERVED FOR COMMAND STRING EXPANSION

```
37F5 FF FF
```

PERIMETER HANDLER DATA DISPLAYS

```
37F7 04 A7         "-S"            ;START ADDRESS
37F9 04 C7         "-E"            ;END ADDRESS
```

BYTES NOT USED

```
37FB C9 C9 06 06 0A
```