The diagram shows the contents of the Z80 CPU internal registers

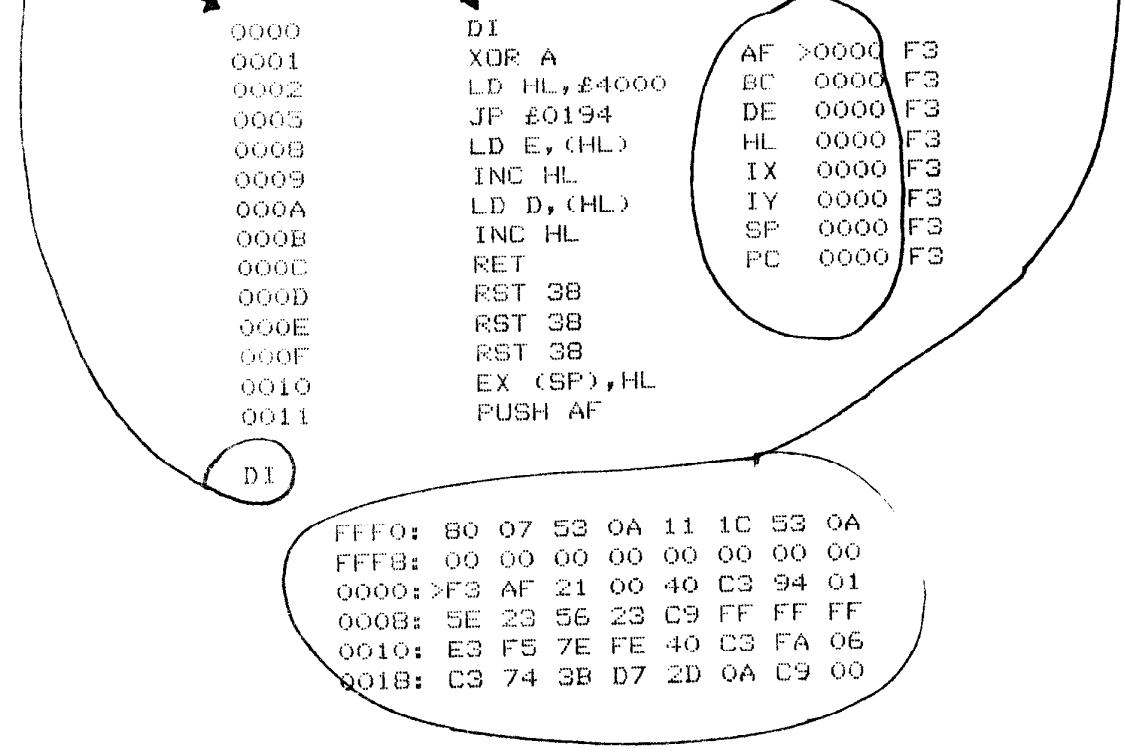
The hex addresses of each instruction

A dis - assembly of the hex codes of the instructions into their equivalent **MNEMONICS**

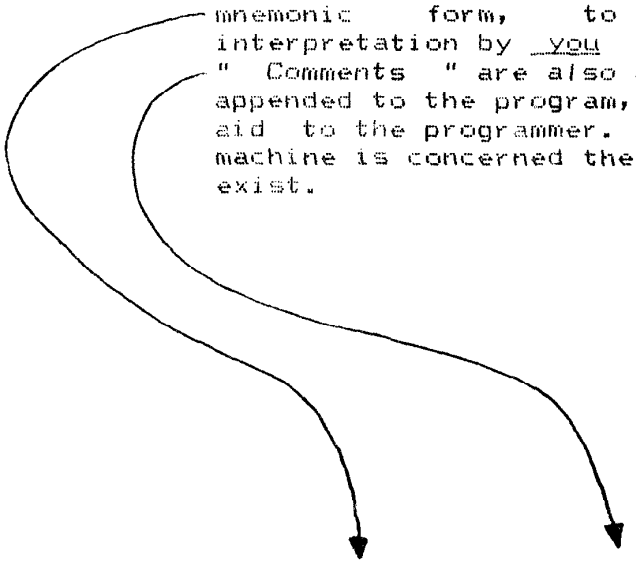
A display of an area of memory

and the next instruction in line to be executed

This is actually the state the CPU is in when power is applied. All register contents are zero, and the PC (Program Counter register) is pointing to location 0000 hexadecimal.



This is the " **Assembly Language** " representation of the program we are going to step through. Notice that the instructions are given in their mnemonic form, to aid their interpretation by <u>you</u> the programmer. " *Comments* " are also allowed to be appended to the program, again, as an aid to the programmer. As far as the machine is concerned the comments don't exist.



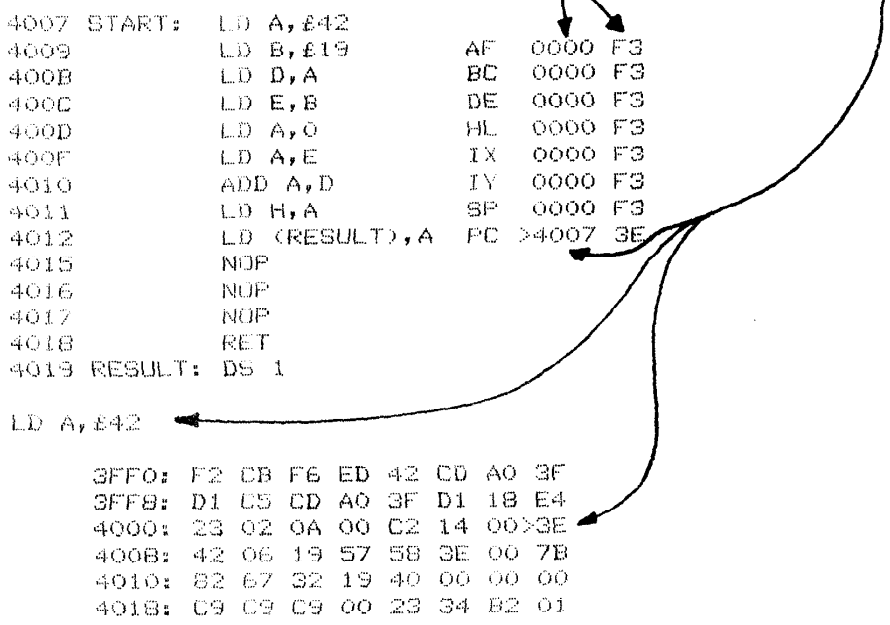
```
4007 START:  LD A,£42    ;load the Accumulator with 42 hex.
4009         LD B,£19    ;and the B register with 19 hex.
400B         LD D,A      ;move the contents of A to D
400C         LD E,B      ;move the contents of B to E
400D         LD A,0      ;put zero in the Accumulator
400F         LD A,E      ;move the contents of E to the Accumulator
4010         ADD A,D     ;add together the Accumulator and D register
4011         LD H,A      ;save the result in the H register
4012         LD (RESULT),A      ;save the result at the address RESULT
4015         NOP         ;do nothing
4016         NOP         ;and again
4017         NOP         ;and again
4018         RET         ;Return to BASIC interpreter
4019 RESULT: DS 1        ;define 1 byte of storage for the result
401A         RET

Symbols:
START   4007    RESULT  4019
```

This frame shows the state of all internal registers prior to executing the program.

Also shown here is a dis - assembly of the area of hex codes for the instructions generated by the Assembly Language program above.

The Program Counter PC has also been set to 4007 hex and shown on the left is the instruction pointed to by the PC. i.e. the next instruction to be executed

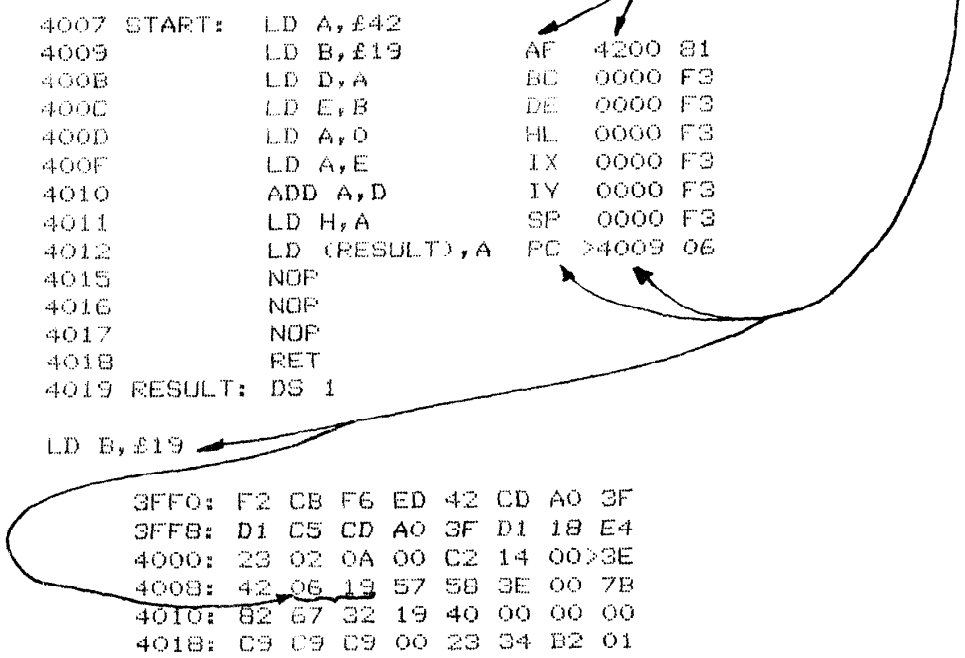
N.B. the bytes shown to the right of the register contents are the contents of the memory location pointed to by the register pair to their left. e.g. if we look at the Program Counter, the byte to the right is 3E hex. At memory location 4007 (arrowed above) the contents are 3E hex.



This is the state of things after the execution of the previous instruction, i.e. LD A,£42

Note that the Accumulator A now has 42 in it and that the Program Counter now points to location 4009 where the next instruction to be executed is to be found

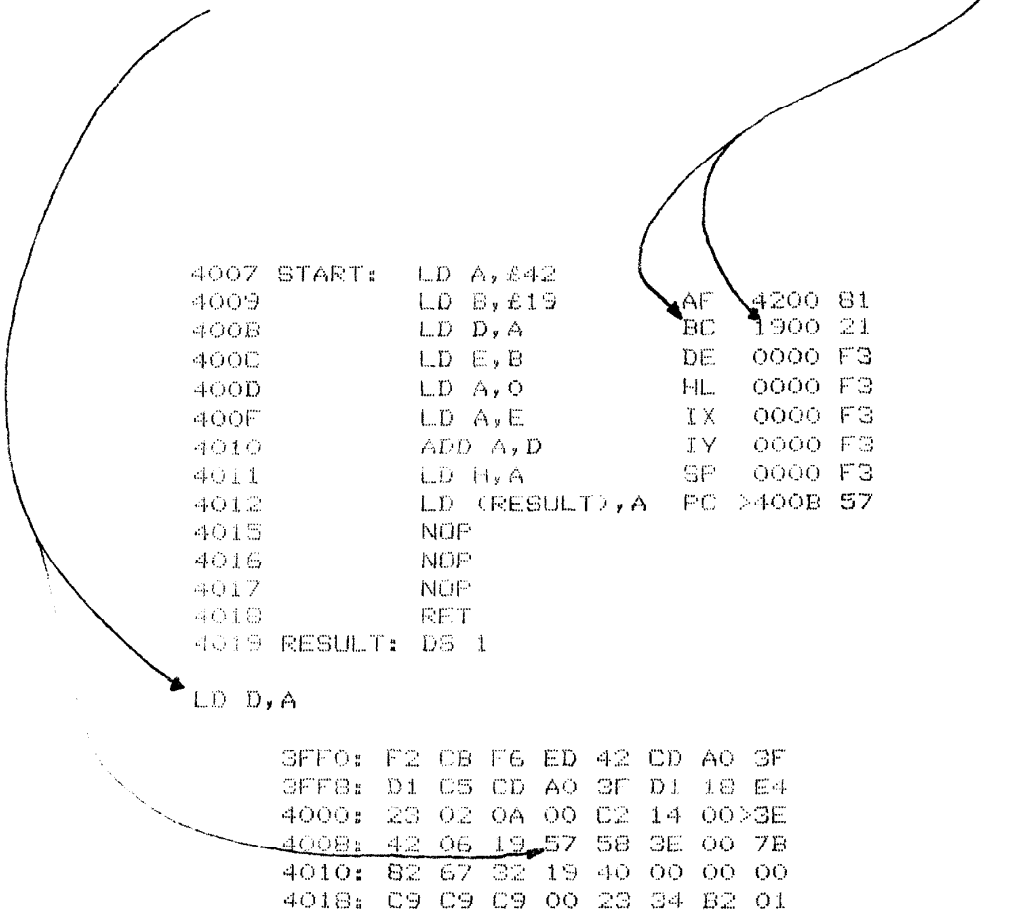
The previous instruction was a <u>two - byte</u> instruction. The data with which to load the register <u>followd *immediately after*</u> the instruction. The CPU "knows" which instructions are 1, 2, 3 or 4 bytes long and automatically adds the necessary value onto the PC after executing the instruction



Here we see the state of the CPU after executing the instruction LD B,£19

Note the contents of the B register and the Program Counter PC

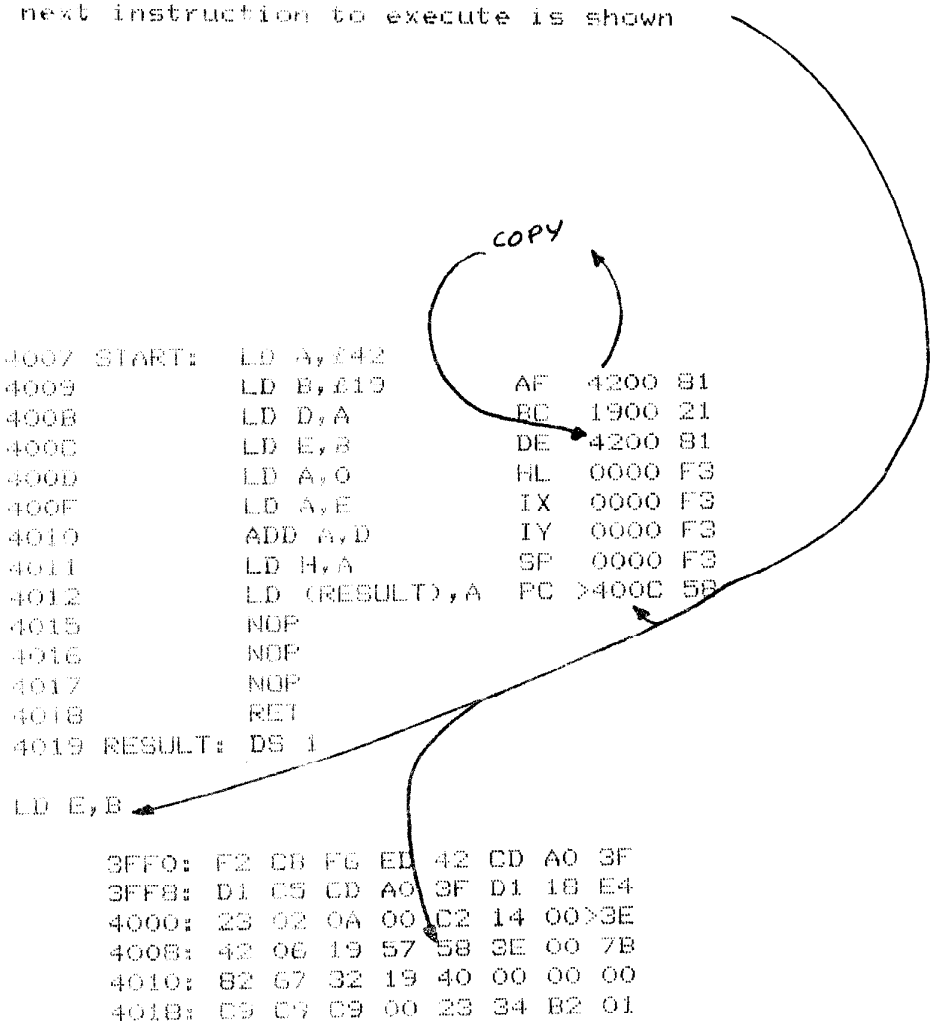
The next instruction to execute being LD D,A

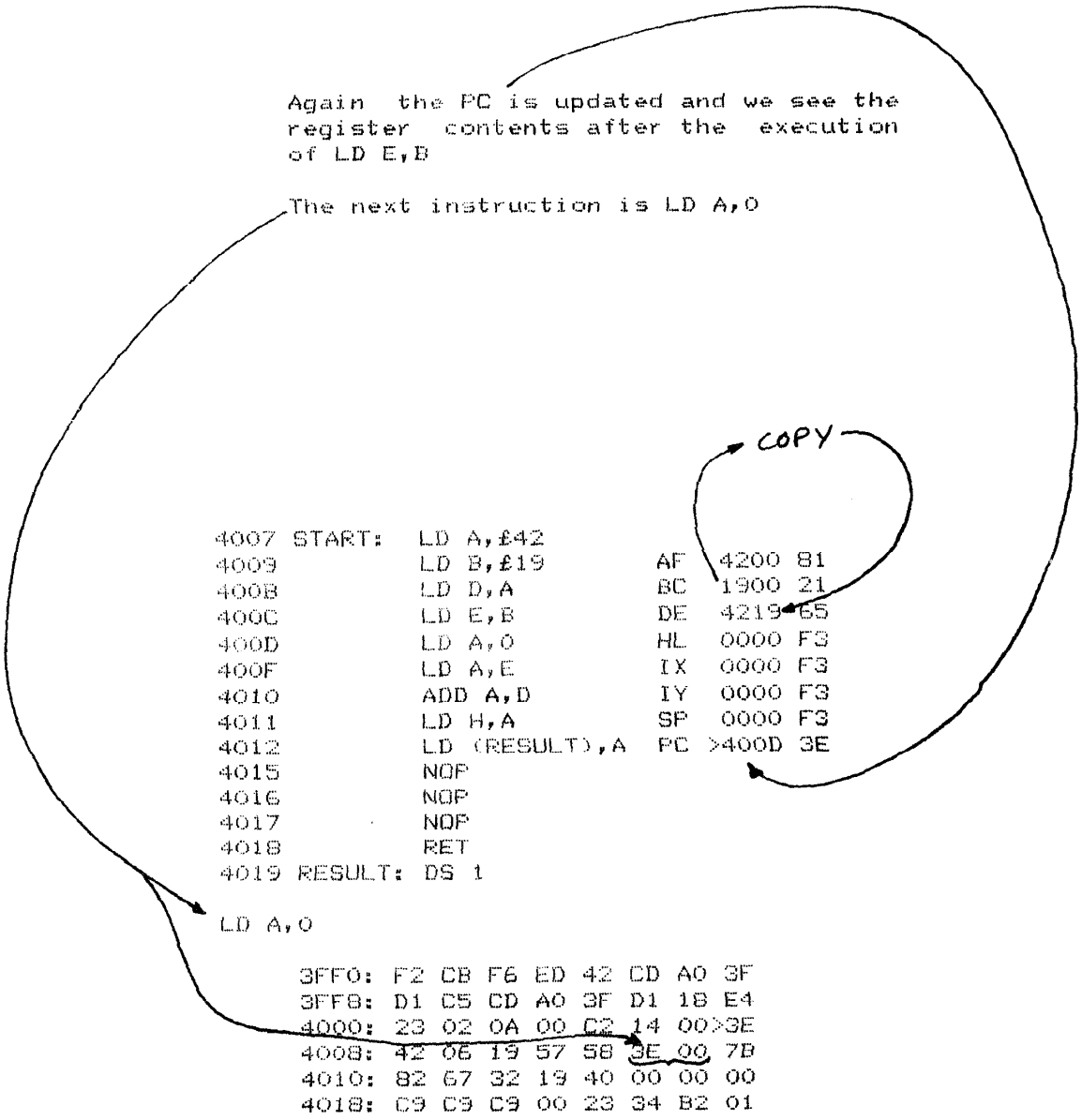


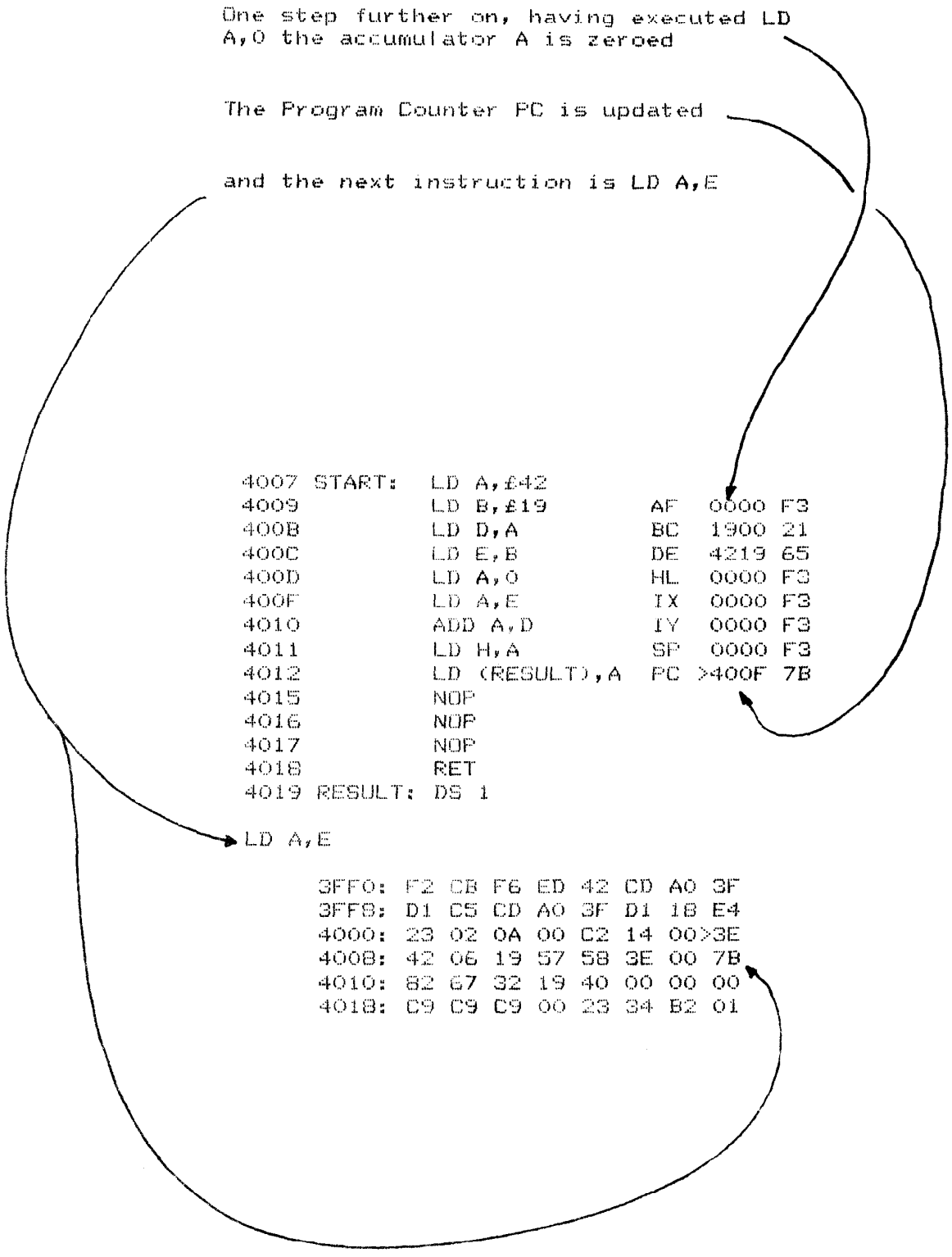
```
4007 START:  LD A,£42          AF  4200 81
4009         LD B,£19          BC  1900 21
400B         LD D,A            DE  0000 F3
400C         LD E,B            HL  0000 F3
400D         LD A,0            IX  0000 F3
400F         LD A,E            IY  0000 F3
4010         ADD A,D           SP  0000 F3
4011         LD H,A            PC >400B 57
4012         LD (RESULT),A
4015         NOP
4016         NOP
4017         NOP
4018         RET
4019 RESULT: DS 1

LD D,A

     3FF0: F2 CB F6 ED 42 CD A0 3F
     3FF8: D1 C5 CD A0 3F D1 18 E4
     4000: 23 02 0A 00 C2 14 00>3E
     4008: 42 06 19 57 58 3E 00 7B
     4010: 82 67 32 19 40 00 00 00
     4018: C9 C9 C9 00 23 34 B2 01
```

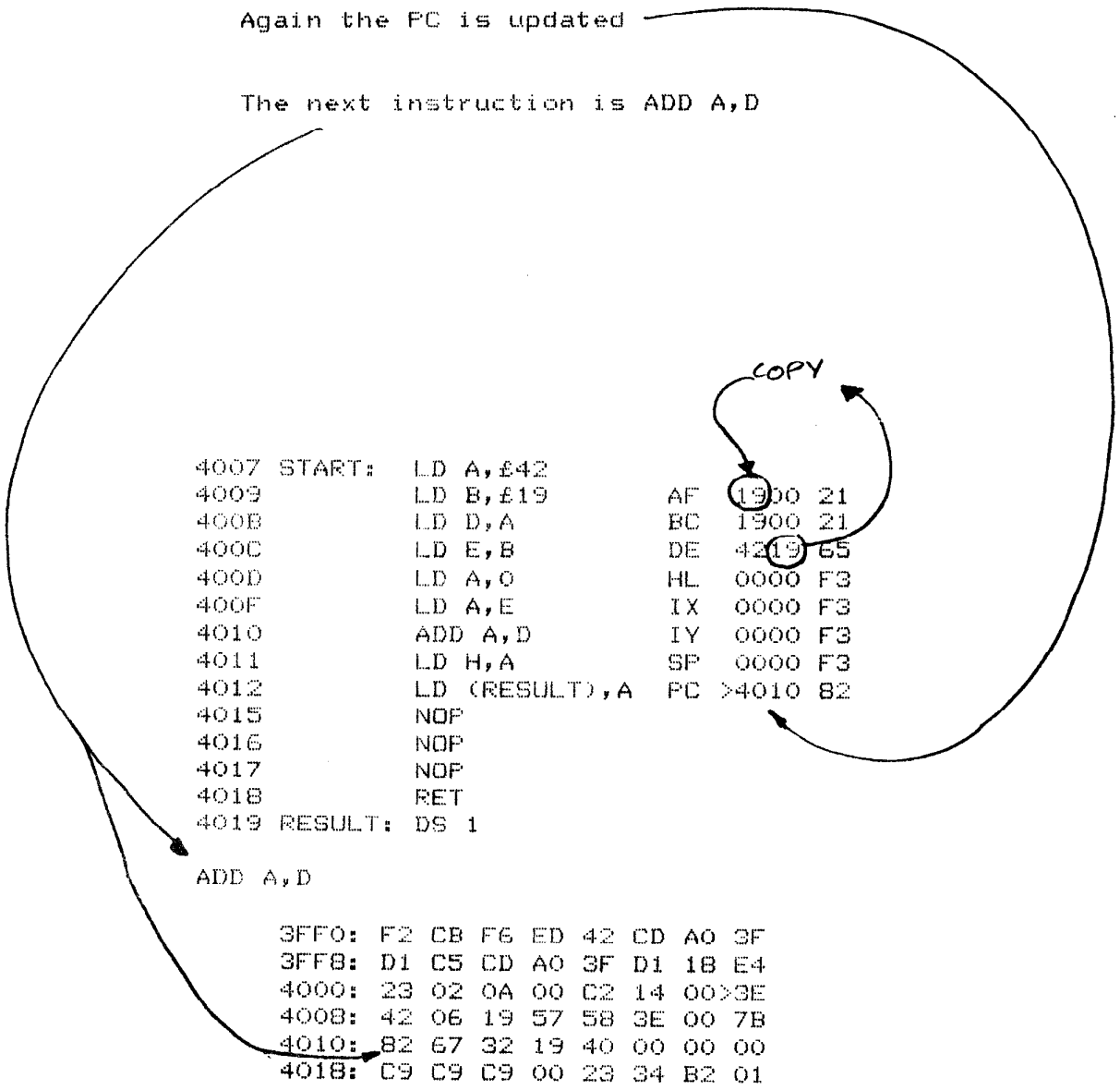
Here we are after executing LD D,A The contents of the D register have now been modified to be a COPY of what was in the Accumulator A

The PC has also been updated and the next instruction to execute is shown




Again the PC is updated and we see the register contents after the execution of LD E,B
The next instruction is LD A,0
COPY
4007 START: LD A,£42
4009 LD B,£19
400B LD D,A
400C LD E,B
400D LD A,0
400F LD A,E
4010 ADD A,D
4011 LD H,A
4012 LD (RESULT),A
4015 NOP
4016 NOP
4017 NOP
4018 RET
4019 RESULT: DS 1
AF 4200 81
BC 1900 21
DE 4219 65
HL 0000 F3
IX 0000 F3
IY 0000 F3
SP 0000 F3
PC >400D 3E
LD A,0
3FF0: F2 CB F6 ED 42 CD A0 3F
3FF8: D1 C5 CD A0 3F D1 18 E4
4000: 23 02 0A 00 C2 14 00>3E
4008: 42 06 19 57 58 3E 00 7B
4010: 82 67 32 19 40 00 00 00
4018: C9 C9 C9 00 23 34 B2 01


One step further on, having executed LD A,0 the accumulator A is zeroed
The Program Counter PC is updated
and the next instruction is LD A,E
4007 START: LD A,£42
4009 LD B,£19
400B LD D,A
400C LD E,B
400D LD A,0
400F LD A,E
4010 ADD A,D
4011 LD H,A
4012 LD (RESULT),A
4015 NOP
4016 NOP
4017 NOP
4018 RET
4019 RESULT: DS 1
AF 0000 F3
BC 1900 21
DE 4219 65
HL 0000 F3
IX 0000 F3
IY 0000 F3
SP 0000 F3
PC >400F 7B
LD A,E
3FF0: F2 CB F6 ED 42 CD A0 3F
3FF8: D1 C5 CD A0 3F D1 18 E4
4000: 23 02 0A 00 C2 14 00>3E
4008: 42 06 19 57 58 3E 00 7B
4010: 82 67 32 19 40 00 00 00
4018: C9 C9 C9 00 23 34 B2 01


Again the PC is updated
The next instruction is ADD A,D
COPY
4007 START: LD A,£42
4009 LD B,£19
400B LD D,A
400C LD E,B
400D LD A,0
400F LD A,E
4010 ADD A,D
4011 LD H,A
4012 LD (RESULT),A
4015 NOP
4016 NOP
4017 NOP
4018 RET
4019 RESULT: DS 1
AF 1900 21
BC 1900 21
DE 4219 65
HL 0000 F3
IX 0000 F3
IY 0000 F3
SP 0000 F3
PC >4010 82
ADD A,D
3FF0: F2 CB F6 ED 42 CD A0 3F
3FF8: D1 C5 CD A0 3F D1 18 E4
4000: 23 02 0A 00 C2 14 00>3E
4008: 42 06 19 57 58 3E 00 7B
4010: 82 67 32 19 40 00 00 00
4018: C9 C9 C9 00 23 34 B2 01

Again the PC is updated and we have just executed the instruction ADD A,D

This took the contents of the Accumulator A, and the D register, added them together in the ARITHMETIC & LOGIC UNIT (ALU) and put the result back into the Accumulator A

The next instruction is LD H,A

```
4007 START:   LD A,£42
4009          LD B,£19          AF  5B08 FF
400B          LD D,A            BC  1900 21
400C          LD E,B            DE  4219 65
400D          LD A,0            HL  0000 F3
400F          LD A,E            IX  0000 F3
4010          ADD A,D           IY  0000 F3
4011          LD H,A            SP  0000 F3
4012          LD (RESULT),A     PC >4011 67
4015          NOP
4016          NOP
4017          NOP
4018          RET
4019 RESULT:  DS 1
```

```
LD H,A

     3FF0: F2 CB F6 ED 42 CD A0 3F
     3FF8: D1 C5 CD A0 3F D1 18 E4
     4000: 23 02 0A 00 C2 14 00>3E
     4008: 42 06 19 57 58 3E 00 7B
     4010: 82 67 32 19 40 00 00 00
     4018: C9 C9 C9 00 23 34 B2 01
```

hex address a result

Again the PC is updated and the result of the previous addition is transferred to the H register

N.B. specific points in the program may be identified by LABELS e.g. START and RESULT in the program. These allow the programmer to define memory locations which are to be referenced by the program by names rather than by absolute hexadecimal address values

We have moved a pointer (>) in the memory display area to highlight where the Accumulator will be transferred to by the next instruction

COPY

```
4007 START:  LD A,£42          AF  5708 FF
4009         LD B,£19          BC  1900 21
400B         LD D,A            DE  4219 65
400C         LD E,b            HL  5B00 DA
400D         LD A,O            IX  0000 F3
400F         LD A,E            IY  0000 F3
4010         ADD A,D           SP  0000 F3
4011         LD H,A            PC >4012 32
4012         LD (RESULT),A
4015         NOP
4016         NOP
4017         NOP
4018         RET
4019 RESULT: DS 1
```

LD (RESULT),A

```
4008: 42 06 19 57 58 3E 00 7B
4010: 82 67 32 19 40 00 00 00
4018: C9 C9 C9 00 23 34 B2 01
4020: 01 00 00 00 02 00 00 53
4028: 54 41 52 D4 00 00 00 00
4030: 23 31 B9 01 03 00 00 00
```

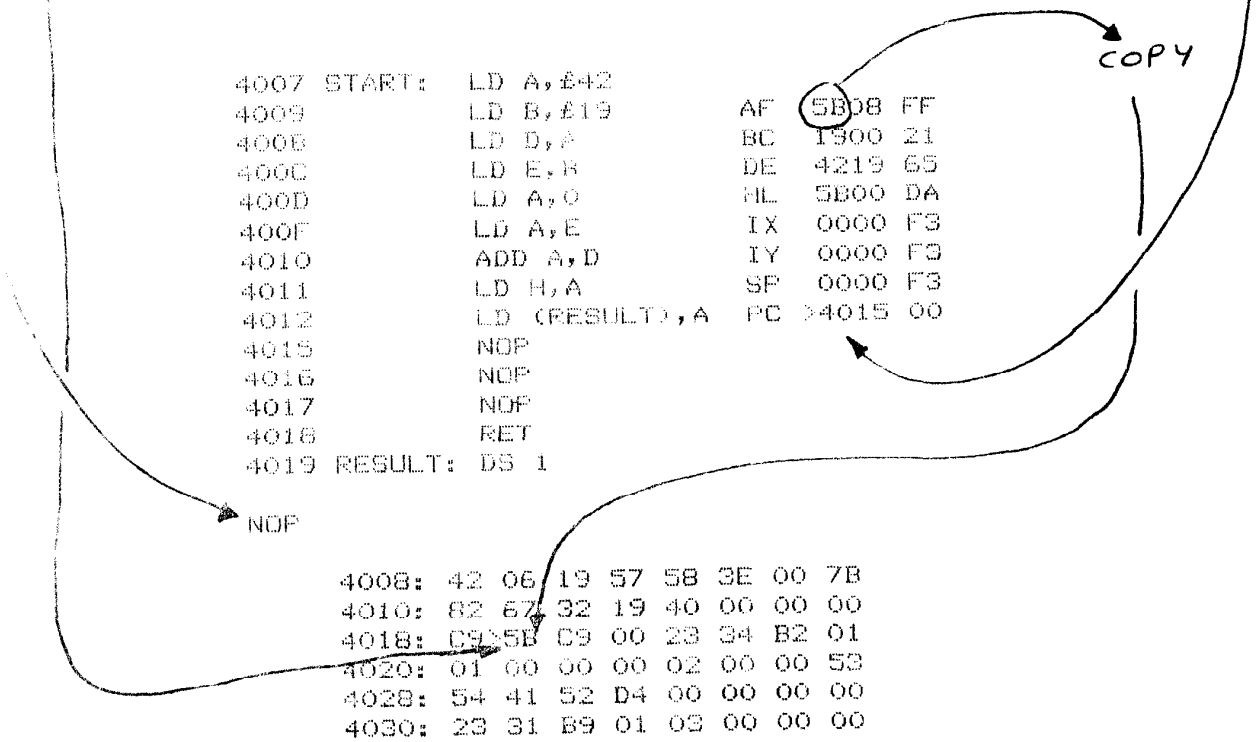
PC updated

Next instruction

We have executed the instruction LD (RESULT),A which transferred the contents of the Accumulator A to the byte at the memory location 4019 hex, labelled " RESULT "

The previous contents of which had been C9 hex, see above, the new contents are 5B hex.

Note the PC was updated by 3 as LD (RESULT),A is a 3 byte instruction



Again the PC is updated

The next instruction is shown

and we should note that having executed the previous NOP instruction all register contents and memory location values are unchanged.
NOP actually means "N o OP eration"
i.e. don't do anything

At this point we'll stop stepping through the program as we would shortly encounter a RET instruction (RETURN) which at the present time would cause us problems.

```
4007 START:  LD A,£42
4009         LD B,£19          AF  5B08 FF
400B         LD D,A            BC  1900 21
400C         LD E,B            DE  4219 65
400D         LD A,0            HL  5B00 DA
400F         LD A,E            IX  0000 F3
4010         ADD A,D           IY  0000 F3
4011         LD H,A            SP  0000 F3
4012         LD (RESULT),A     PC >4016 00
4015         NOP
4016         NOP
4017         NOP
4018         RET
4019 RESULT: DS 1
```

NOP

```
4008: 42 0E 19 57 58 3E 00 7B
4010: B2 67 32 19 40 00 00 00
4018: C9>5B C9 00 23 34 B2 01
4020: 01 00 00 00 02 00 00 53
4028: 54 41 52 D4 00 00 00 00
4030: 23 31 B9 01 03 00 00 00
```