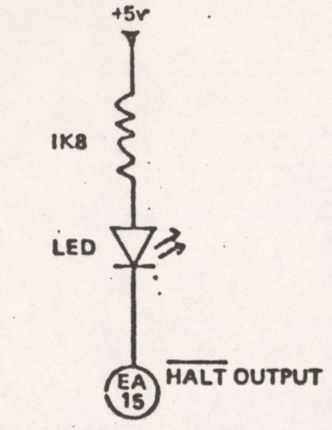

FIG 10c – $\overline{\text{HALT}}$ TEST

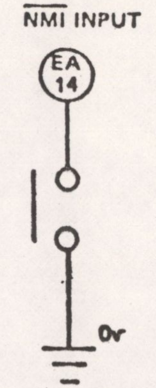

FIG 10d – $\overline{\text{NMI}}$ TEST

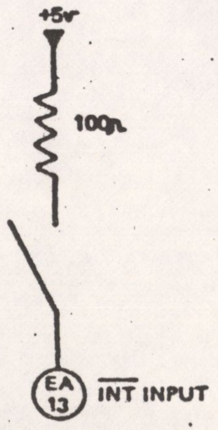

FIG 10e – $\overline{\text{INT}}$ TEST

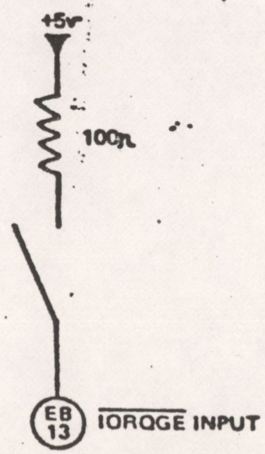

FIG 10f – $\overline{\text{IORQGE}}$ TEST

HALT EXPERIMENT

The $\overline{\text{HALT}}$ line is an output from the Z80A chip. When active (output = logic 0) it indicates that the CPU has executed a software HALT instruction and is awaiting an interrupt from an external device. Whilst the CPU is halted it executes NOP's (NO operations which do nothing). These ensure that the memory refresh mechanism continues to operate irrespective of the duration of the HALT.

The circuit for this experiment is shown in fig 10c. Any small LED (light emitting diode) will do. Now try running this little program.

```
10  CLEAR 32499
20  POKE 32500,118 :REM HALT
30  POKE 32501,201 :REM RET
40  LET a = USR 32500
50  GO TO 40
```

Line 10 stops BASIC using memory above 32499. Lines 20 and 30 set up a machine code program to execute a HALT then return control to BASIC (after an interrupt has been received from the ULA). Line 40 jumps to the small machine code program. You should notice that the LED lights when you run the program. In fact it is switched on when the program first gets to line 40. Then the ULA sends an interrupt to the CPU which turns off the HALT line. The program goes back to line 40 and turns the HALT on again, until the next interrupt is received from the ULA. You can't actually see the LED flashing on and off because it is flashing at 50 Hz (the ULA interrupts the CPU 50 times every second).

NMI TEST

The circuit for this test is shown in fig 10d. It simply requires a push switch between the $\overline{\text{NMI}}$ input on the edge connector and the 0 volt line. If you press the button to generate an $\overline{\text{NMI}}$ (non maskable interrupt) then, no matter what the CPU was doing previously, it will now start to run the interrupt service routine at address 66Hex (102 decimal). This in fact will reinitialise BASIC and so it has the same effect as pressing the RESET button.

INT AND IORQGE TESTS

In these tests we are going to use the circuits in fig 10e and fig 10f. These effectively connect the appropriate lines on the edge connector directly to +5v. The 100 ohm resistor is there to reduce the current taken from the power supply and can be omitted.

Now enter the following short program. Leave both switches for $\overline{\text{INT}}$ and $\overline{\text{IORQGE}}$ open whilst doing this.

```
10  CLS
20  PRINT AT 0,0; PEEK 23672 + 256*PEEK 23673
30  GO TO 20
```

When you run the program you will see an incrementing number at the top of your screen. This shows how many times the CPU has been interrupted since it was switched on. Now operate the INT switch. The clock stops. Try to BREAK your BASIC program. Nothing happens! All is not lost however. Switch the INT switch off again and the counter restarts where it left off. You can also use the keyboard again.

Why did this happen? Well, the way in which the counter and keyboard operate relies upon interrupts. Once every 50th of a second the ULA activates the $\overline{\text{INT}}$ line. This causes the CPU to jump to a little bit of machine code in the BASIC