Working with the A-ONE

WOZ MONITOR

The original Apple 1 did not come with a reset circuit, which means that the user has to press the RESET switch in order to get the machine started. Once you do that a back slash '\' is printed on the screen and the cursor drops down one line. The cursor position is represented by a flashing '@' symbol.

You can now type address, data and commands which will be executed as soon as you press the RETURN key. The input buffer can hold up to 127 characters; if you type more characters before pressing the RETURN key the input line will be erased and will start again from scratch. This overflow situation is indicated by a new back slash after which the cursor again drops one line.

Because of the primitive nature of the terminal there are not many line-editing features available. You can press the back arrow key to erase characters from the input buffer, but the erased characters will not be erased from the screen nor will the cursor position back-up. You'll have to keep track of the changes yourself. It's obvious that you can easily get confused when a line contains too many corrections or when an error is detected all the way at the other end of the input line. In that case it would be easiest to cancel the input and start all over again. Cancelling the input is done by pressing the ESC key.

Address inputs are truncated to the least significant 4 hexadecimal digits. Data inputs are truncated to the least significant 2 hexadecimal digits. Thus entering 12345678 as an address will result in the address 5678 being used. This feature can be used to your advantage to correct typing errors instead of using the back arrow key.

If an error is made while parsing the input line then the rest of the line is simply ignored without warning even though commands executed before the error are executed normally.

Examining memory (memory dump)

You can examine the contents of a single memory location by typing a single address followed by a `RETURN`:

```
4F

004F: 0F
```

Note: The **bold** typed characters are what the user types. All other characters are responses from the Apple 1.

Now let us examine a block of memory from the last opened location to the next specified location:

```
.5A
```

```
0050: 00 01 02 03 04 05 06 07
0058: 08 09 0A
```
Note: `004F` is still considered the most recently opened location.

We can also combine the previous two examples into one command:
`4F.5A`

```
0040: 0F
0050: 00 01 02 03 04 05 06 07
0058: 08 09 0A
```
Note: Only the first location of the block `4F` is considered opened.

You can also examine several locations at once, with all addresses on one command line:
`4F 52 56`

```
004F: 0F
0052: 02
0056: 06
```
Note: `0056` is considered the most recently opened address.

Let's take this concept to the extreme and combine some block and single address examinations on one command line:
`4F.52 56 58.5A`

```
004F: 0F
0050: 00 01 02
0056: 06
0058: 08 09 0A
```
Note: By now you won't be surprised that `0058` is considered the most recently opened location.

Finally let's examine some successive blocks of memory. This can be handy if you want to examine a larger block of memory which will not fit on one monitor screen. Remember that there is no way to halt a large examine list other than hitting the RESET button!
`4F.52`

```
004F: 0F
0050: 00 01 02
.55

0053: 03 04 05
.5A

0056: 06 07
0058: 08 09 0A
```

Depositing memory (changing memory contents)

This is how to change a single memory location (provided it is RAM memory of course):
`30:A0`

```
0030: FF
```
Note: `FF` is what location `00300` contained before the operation, from now on it contains `A0`.
Location `0030` is now considered the most recently opened location.

Now we're going to deposit some more bytes in successive locations, starting from the last deposited location:

```
:A1 A2 A3 A4 A5
```

Note: Location `31` now contains `A1`, location `32` contains `A2` and so on until location `35` which now contains `A5`.
Combining these two techniques will give us the next example:

```
30:A0 A1 A2 A3 A4 A5
```

```
0030: FF
```

Note: Location `0030` used to contain `FF` in this example.
Breaking up a long entry into multiple command lines is done like this:

```
30:A1 A2
```

```
0030: FF
:A2 A3
```

```
:A4 A5
```

Note: A colon in a command means "start depositing data from the most recently deposited location, or if none, then from the most recently opened location.

Now we're going to examine a piece of memory and then deposit some new data into it:

```
30.35
```

```
0030: A0 A1 A2 A3 A4 A5
:B0 B1 B2 B3 B4 B5
```

Note: New data deposited beginning at most recently opened location, which is `0030` in this example.

Running a program

To run a program at a specified address:

```
10F0R
```

```
10F0: A9
```

Note the cursor is left immediately to the right of the displayed data; it is not returned to the next line. It's the program's responsibility to control the rest of the output.

From now on the user program is in control of the Apple 1. If the user program does not return to the Woz monitor (by jumping to address $FF1F) you'll have to press the RESET key to stop your program and return to the Woz Monitor.

You can also enter a program and run it all from the same command line. Please note that this only works for very short programs of course.

```
40: A9 0 20 EF FF 38 69 0 4C 40 0 R
```

```
40: FF
```

Note: `FF` is the previous contents of location `0040`.