**Base Machine**

The base machine is a simple SC/MP machine, which does not use the extended 64k layout, i.e. it is set to 4k of program space, clocked at 1Mhz (e.g. 500Khz cycle)

The basic machine allows for 1k bytes of ROM and 128 bytes of RAM, and input through a bank of toggle switches accessed via a tri-state buffer. Output is provided by a bank of 8 LEDs, which can be either 8 LEDs displaying binary values, or 2 x 7 Segment LEDs with 9368 decoder/latches (binary to hexadecimal), which is mapped onto the ROM space when you write to it

* 000-7FF up to 2k x 8 EPROM (read) (Monitor), output LEDs (write)
* 800-BFF up to 1k x 8 RAM (Program space)
* C00-FFF toggle switch input (read only)

Active switches are :-

* SA Key “A”
* SB Key “B”
* NRST Key “R”

Unusually, possibly because it was never actually built, it had the option of a monitor ROM, rather than keying in via toggle switches. This was not that common in the earliest computers, especially SC/MP designs.

The weird amount of RAM and ROM is because of the use of 6800 technology, from the ETI Microprocessor series which was 6800 based. They also have lots of chip select inputs.

**Expansion (Actual)**

1. The machine’s memory was expanded, up to 2k of ROM and 1k of RAM by adding a second 6830 and 6810s to suit (up to 8)
2. A cassette interface was added connected to SIN and SOUT, whose design leaves much to be desired, let’s draw a veil over that bit.
3. An octal keyboard was added using the toggle switches as push buttons 0-7.
4. A keyboard interface was added using bit 0 on the same read as the toggle addressing (bit 0 = toggles, bit 1 = keyboard) – e.g. you read $0C01. The keyboard is standard ASCII with a positive going strobe on D7.
5. A video interface was added to the same space as the toggle addressing (1k of RAM space). This was a single 1k x 1 RAM driven with counters and multiplexers, one pixel mapped to each byte between C00 and FFF, connected to bit 0. Because the keyboard and toggles were mapped onto this, this was write only, and in fact the design was based around a 2102 1k x 1 RAM ; this has separate in and out pins, the in pins being connected to the databus and the out pin driving the video circuitry. There were 2 x 6 bit counters and 3 quad multiplexers switching between the address bus and the counters with no syncing, so it would have flashed on writing. The timing wouldn’t have worked, but then designing PAL video at 12/13 is probably beyond almost anyone. The actual design apart from the timing is not at all bad.
6. I recall, but have lost, a printer that design worked on strip paper ; the idea was there was a solenoid print head with 7 actuators that was pulled through by a motor (not even a stepper motor) and the code fired the solenoids as the paper went past ; a bit like an automatic Dymo Labeller crossed with a paper tape punch. A real Sinclair design, I think ;-)
7. I do recall designing a memory expansion board which was just lots and lots of RAM memory. Would have cost a fortune in the mid 1970s.

**Software**

**Remembered**

* Toggle switch Monitor that used the LEDs (survived)
* Octal keypad Monitor (lost – do not recall whether it used the display or LEDs, but given the relative complexity of coding for the display, it probably used the LEDs)
* PILOT interpreter (lost, but presumably used ASCII keyboard and display)

**Considered**

* Tiny BASIC
* FORTH

**Expansion (new)**

Any expansion must be in keeping with the time, (e.g. no AY-3-8910 chips or more modern equivalents to generate sound effects)

1. The obvious expansion is to decode A12-A15 and use them to expand ROM and RAM beyond the (then) astronomical 2k and 1k levels. I must have done this because I remember designing a RAM board that wouldn’t fit in the design unless A12-A15 were latched. It is worth recalling that in the mid 1970s 1k of program memory was colossal.
2. There is no sound. Most of these early computers didn’t have any sound (apart from the Apple 2 which just Beeped – single board microcontrollers, Pet, Apple, TRS80, Nascom etc. The sound is a NE555 running at 1274Hz which is modulated by pin 5 via a simple D/A network connected to F0/F1/F2 (see NE555.py). The frequencies this (roughly) generates are shown below.
3. An SC/MP can run at 4Mhz clock speed, rather than 1Mhz. An Arduino can just about keep up at 1Mhz

**Memory Map**

|  |  |  |
| --- | --- | --- |
| Address Range | Read | Write |
| 0000-07FF | ROM monitor (1 or 2 x Motorola 6830 1k x 8 ROM chip) | LEDs , either 8 x single LEDs or 2 x DM9368 chips driving a 7 segment display. |
| 0800-0BFF | RAM Memory (1 to 8 x Motorola 6810 128 x 8 RAM chip) | |
| 0C00-0FFF | Toggle switches (bit 0 = 0)  Keyboard (bit 0 = 1) – key pressed when D7 is logic ‘1’ | 32 x 32 Video write only display via bit 0. |
| 1000-FFFF | *(theoretical expansion)* | |

**Sound Frequencies**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S Value | F2 | F1 | F0 | Frequency (Hz) |
| 0 | 0 | 0 | 0 | Off |
| 1 | 0 | 0 | 1 | 209 |
| 2 | 0 | 1 | 0 | 383 |
| 3 | 0 | 1 | 1 | 592 |
| 4 | 1 | 0 | 0 | 668 |
| 5 | 1 | 0 | 1 | 877 |
| 6 | 1 | 1 | 0 | 1051 |
| 7 | 1 | 1 | 1 | Off |