**SCRUMPI 3 Information**

Sources

1. “BRO” Brochure picture published on the Register
2. “ETI” Justin A.T. Halls review, Electronics Today International October 1978
3. “JMK” Scrumpi 3 – A Microprocessor with low cost I/O, by JMK.

Memory Map

Scrumpi 3 is designed to operate in page 7xxx, this is known from the disassembly of the code shown in the brochure.

|  |  |
| --- | --- |
| 000-1FF | 512 x 8 “PROM B” |
| 200-5FF | 2 x 512 x 8 PROM (Expansion) |
| 600-7FF | 512 x 8 “PROM A” |
| 800-BFF | 1k x 8 RAM (Expansion) pairs of 256x4 2112 RAM chips |
| C00-C0F | Keyboard Port, design conjecture |
| D00 | UART, no idea whatsoever ☺ |
| E00-EFF | 256 x 8 VDU Memory, pair of 256x4 2112 RAM chips |
| F00-F7F | 8154 I/O |
| F80-FFF | 8154 RAM |

General Information

1. The CPU is clocked at 3.5Mhz and is an SC/MP II (BRO)
2. The basic system has 1k PROM and 128 byte RAM made up of MM5204 PROMs (BRO)
3. This is expandable to 2k PROM and an extra 1k RAM made up of 2112 RAM Chips (BRO)
4. The RAM chip is provided by an NS 8154 RAM I/O Chip (ETI)
5. The RAM chip is arranged so that the RAM is F80-FFF (ETI)
6. The 1st expansion PROM is at 0200 (ETI) and is checked for.

Video

1. The VDU is 32 x 8 characters using a 7 x 9 matrix, 6 bit ASCII and is mapped to RAM memory at E00-EFF (BRO)
2. Bit 7 of the character set inverts the font to black on white. (BRO)
3. Flag F0 further inverts this for the whole screen (BRO/JMK)
4. The inversions cancel each other out (JMK)
5. The ASCII mapping is done in software in the VDU, bits 0-5 drive the character generator, bit 7 the inversion, bit 6 unused (JMK) “The lower 6 bits go into a DM8678 type of character generator”

Keypad

1. 16 key keypad which maps directly to 4 bit data (BRO)
2. INT key is connected to SENSE A and is used for CR/LF (BRO)
3. There are three further shift keys which represent different chunks of the ASCII character set (BRO)
4. These keys are “PUNC” “ALPH1” and “ALPH2” (ETI)
5. Bit 7 of the keyboard port is the additional key (BRO)
6. The decoding is done by the MPU rather than an external encoder (JMK)

Serial Interface

1. SB and F1 drive a software TTY (BRO)
2. The UART is at address D00 with a x16 clock derived from the VDU line frequency (15.625khz) (BRO)
3. The UART is an AY-5-1013 (ETI)
4. SIN/SOUT are not used (BRO)

Parallel Interface

1. There are two 8 bit I/O ports. (BRO)
2. F2 is an external connection (BRO)

Software Routines

1. The character set routine outside ASCII is 01 (Clear) 08 (Backspace) 0D (CR/LF) (BRO)
2. The software routines PUTVDU,GETKBD,PUTTTY,GETTTY,PUTART,GETART are present in “ROM A”. From the decoding we know this to be the ROM at $7600-$7700 (BRO) this is confirmed by the article (ETI)
3. 64 bytes of the RAM is free (ETI)
4. 32 bytes of the RAM is the stack (ETI)
5. 8 bytes are used for storing labels (ETI)
6. There are eight Subroutines – Read Keyboard (GETKBD), Write VDU (PUTVDU), Write a string (terminated in $04, from the listing) (?), write hex with or without a trailing space (?,?) read hex from keyboard (?), display six digit number with sign (?) (ETI)
7. According to the ETI article there is no detail on how to use the UART and I/O ports ?

Monitor

1. PROMB has instructions G,H and I (BRO)
2. The other intructions are described in the ETI article (ETI)

Conjecture

1. The 1k expansion RAM is from 800-BFF because there is no other continuous 1k block. It could be at 400-5FF but this is bonkers.
2. The second expansion PROM is at 400-5FF for the same reason.

Things we don’t know

1. Precisely what is in the monitor. The brochure says one, the review another. Difficult to comprehend that the reviewer would have made the routines up and sounds plausible except the last one (decimal ?). They could both be right ☺
2. How the keyboard works. We know there are 8 bits available. Bit 7 is the keyboard extra key (INT). The other three function keys (PUNC,ALPH1,ALPH2) have to be used in conjunction with other keys. It follows that it is simpler design to assign these to Bits 4,5 and 6 (see decoding)
3. The lower 3 bits are presumably therefore 4 input lines, and the keyboard is read using A0-A3 in a similar fashion – these bits are encoded in a 4->2 fashion to make 16 key keypad.
4. How the UART is connected. The ETI author, who isn’t a fool, couldn’t figure this out either and resorted to tracing the PCB. It is not obvious from the data sheet how you might do this as it is with some others. Common sense suggests putting RDE, SWE and DS on address bits and connecting the In and Out together and relying on software to keep the bits valid (all are active low), and using a fourth for RDAV (e.g. active on address select to reset DAV).
5. The UART configuration may be hard wired, or it could be driven via a latch (NB2,NB1,TSP,NP,EPS are required, one could lose 5/6 bit data so tie NB2 to logic ‘1’ and that means one quad latch – probably worth it)

Keyboard ASCII Conversion

One possible way in which the bits 4,5 and 6 work is shown below, which minimises the code needed to scan the keyboard.

Seems as good as any.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Unshifted | Alpha 1 (4) | Alpha 2 (5) | Punc (6) |
| Read Bits 4-6 : | $0x | $1x | $2x | $4x |
| Xor $40 | $4x | $5x | $6x | $0x |
| Set to $30 if Zero. | $4x | $5x | $6x | $3x |
| Add $F0 | $3x (0-9 etc.) | $4x (@A-O) | $50 (P-Z etc,) | $2x (Space etc) |