3-chip 60Kbyte RAM expansion for the NASCOM 1

Questions or comments to the author

Neal Crook, foofoobedoo@gmail.com

13Oct2019 V1.0
05Apr2020 V1.1 Correct databus ordering on NASCOM Connector

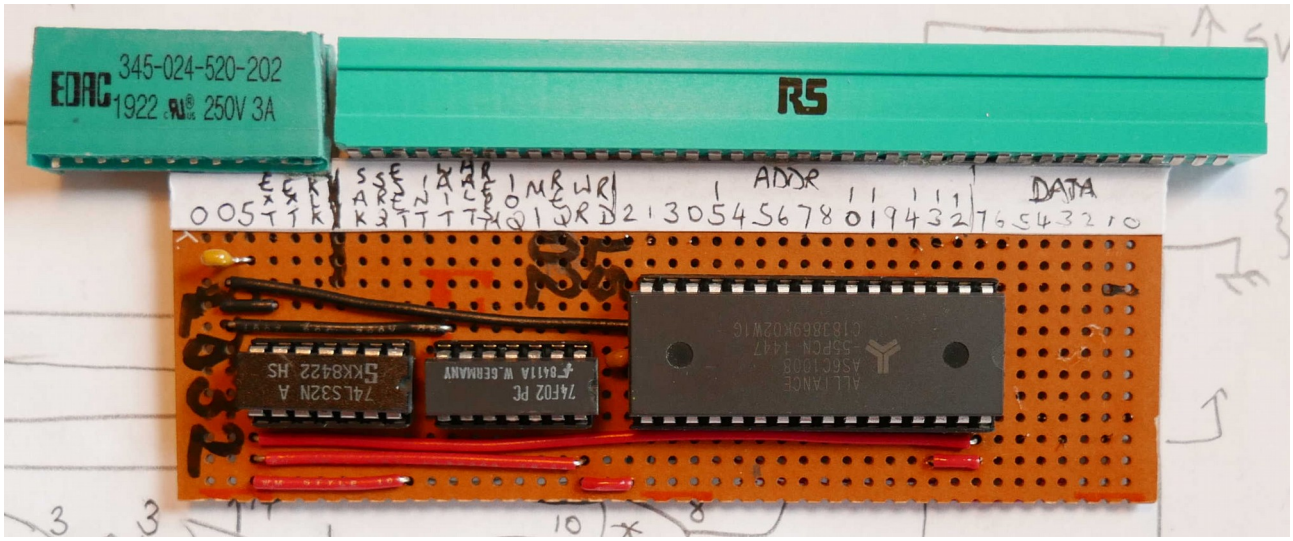

Illustration 1: RAM expansion, top-side

This document describes the design and construction of a 60Kbyte RAM expansion board for a NASCOM 1. It attaches to the 43-pin edge connector.

The memory is implemented using an Alliance Semiconductor AS6C1008. This is a static 128Kbyte x 8 device which runs on a supply rail of 2.7V – 5.5V. In this design, only 60Kbytes are used; the rest of the RAM is totally wasted!

This part is available as a 32-pin DIP for about £3 -- from component distributors or on EBAY.

The other two active components are a 74LS32 and a 74LS02. In addition, you will need an edge connector, some veroboard, decoupling capacitors and hook-up wire.

By locating the RAM chip carefully, relative to the edge connector pick-up, 11 of the address line connections can be made without adding any wires.

Theory of operation

Refer to Illustration 2: Circuit Schematic. The RAM is connected to D[7:0] and A[15:0] from the NASCOM 1 expansion connector. The unused RAM address line is tied off (to 0V or +5V). The unused chip-enable (CE2) is tied HIGH.

The RAM chip-enable (/CE) is wired to /MREQ, so that the RAM is enabled for all memory accesses.

The RAM write-enable (/WE) is wired to /WR, so that writes to any address go to the RAM.

The RAM output-enable (/OE) needs some control logic; if it was simply connected

to /RD, it would drive at the same time as accesses to the NASCOM 1 memory (EPROM, video RAM and workspace RAM), causing bus contention.

An unexpanded NASCOM 1 uses the low 4Kbytes of address space. Its LK5 jumper is set to “INTERNAL” and those 4Kbytes are repeated (aliased) through the 64KByte address space.

With the RAM expansion board fitted, The LK5 jumper must be set to “EXTERNAL”. The RAM expansion board is responsible for asserting /MEMEXT when NASCOM 1 memory is being addressed. The expansion RAM be will be enabled for all other addresses.

The memory decoding is achieved using three OR gates. These gates assert /MEMEXT when A[15:12]=0000. A NOR gate generates a signal that is the inverse of /MEMEXT – ie, it is logic 1 when the NASCOM 1 is being addressed. This is OR-ed with /RD to generate the RAM output-enable (/OE). As a result, the RAM is enabled during any read that is outside the NASCOM 1 address range.

The NOR gate can be replaced with an inverter driven from /MEMEXT – use whatever you have in your parts box.

3 sections of the 74LS02 NOR are unused; the unused inputs can be tied to +5V (or left to float high if a genuine TTL device is in use).

All three devices are connected to +5V and to 0V, both available on the NASCOM 1 expansion connector. I suggest 1-3 0.1uF decoupling capacitors from +5V to 0V.

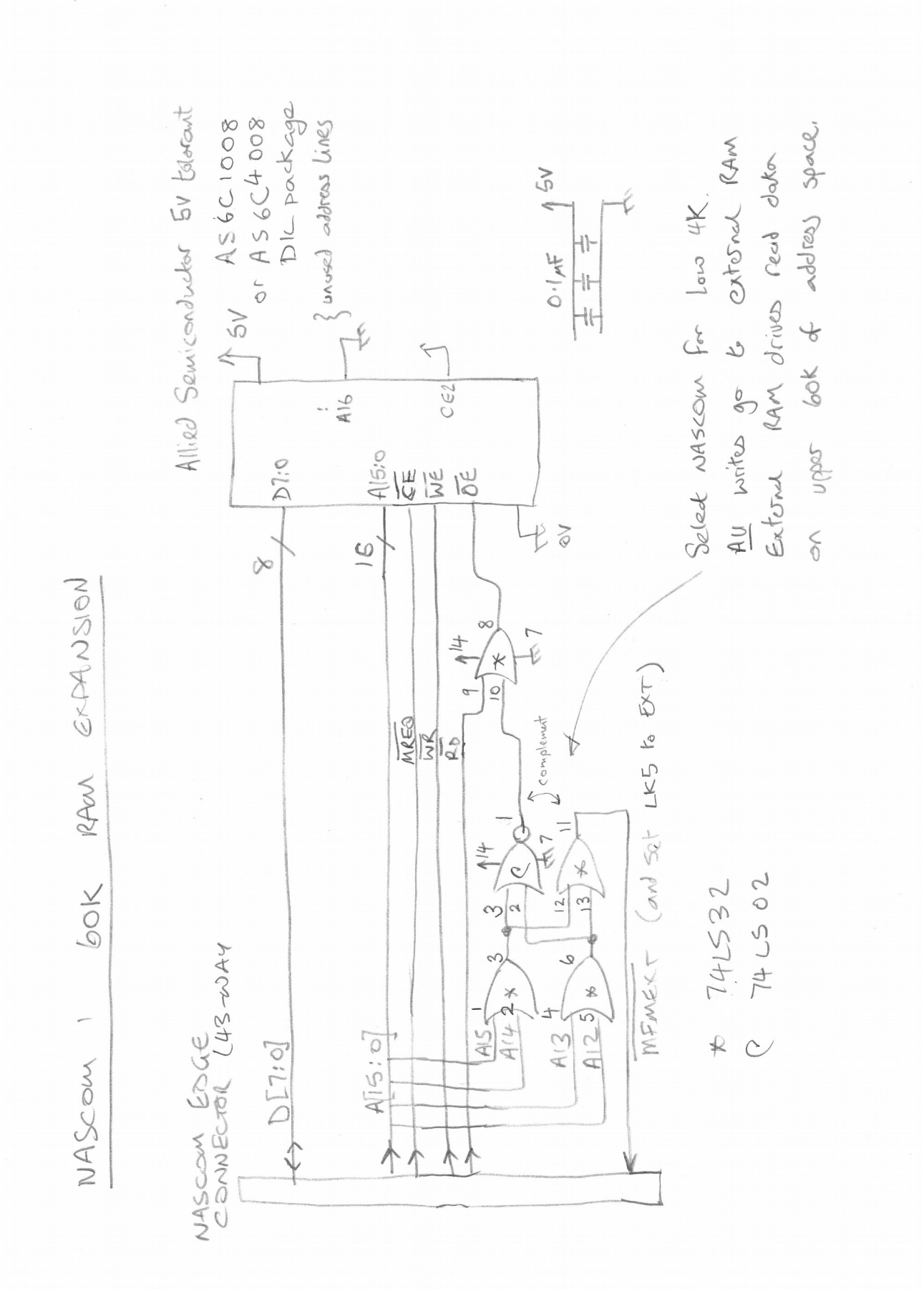

Illustration 2: Circuit Schematic

Construction

I constructed the prototype on a piece of 43-way 0.1″ Veroboard. I found an edge-connector on EBAY but did not read the description properly and bought a connector that was too small. I ended up using 2 connectors, of different depths, each cut to length. Line the connectors up carefully, either on the NASCOM 1 expansion connector itself or on another piece of Veroboard, then solder the edge connector tails onto the piece of Veroboard that will hold the RAM and other components. The end result should be mechanically stable. If it is not, you might need some epoxy to make it rigid.

Glue a narrow strip of paper in place and mark up the NASCOM 1 signals. This will help you in construction and in debug.

Place a 32-pin DIL socket for the RAM such that pin 2 of the RAM aligns with A12 on the NASCOM 1 expansion connector (I cut down a 40-pin DIL socket). Place 2 14-pin DIL sockets for the 74LS32 and 74LS02. Refer to the pictures. Orient all the sockets to have pin1 at the same end.

YOU CAN CONNECT THE 16 ADDRESS LINES TO THE RAM IN ANY ORDER. LIKEWISE, YOU CAN CONNECT THE 8 DATA LINES TO THE RAM IN ANY ORDER. DO WHATEVER MAKES THE WIRING EASIEST. HOWEVER, A[15:12] (IN ANY ORDER) MUST BE USED FOR THE CONNECTIONS TO THE 74LS32.

Start by using thick-ish wires to connect +5V and 0V to each of the 3 chips. In each case, wire directly from the chip to a power rail from the edge connector (don’t daisy-chain the power connections). Fit the decoupling capacitors.

Use thinner wire (for example, PTFE-insulated wire-wrap wire) for the other connections; use the RAM chip pinout to guide wiring of the address lines, data lines and control lines to the RAM. Use the circuit schematic to guide wiring of the 74LS32 and 74LS02.

Cut the appropriate traces on the Veroboard.

A rear view of my board is shown below.

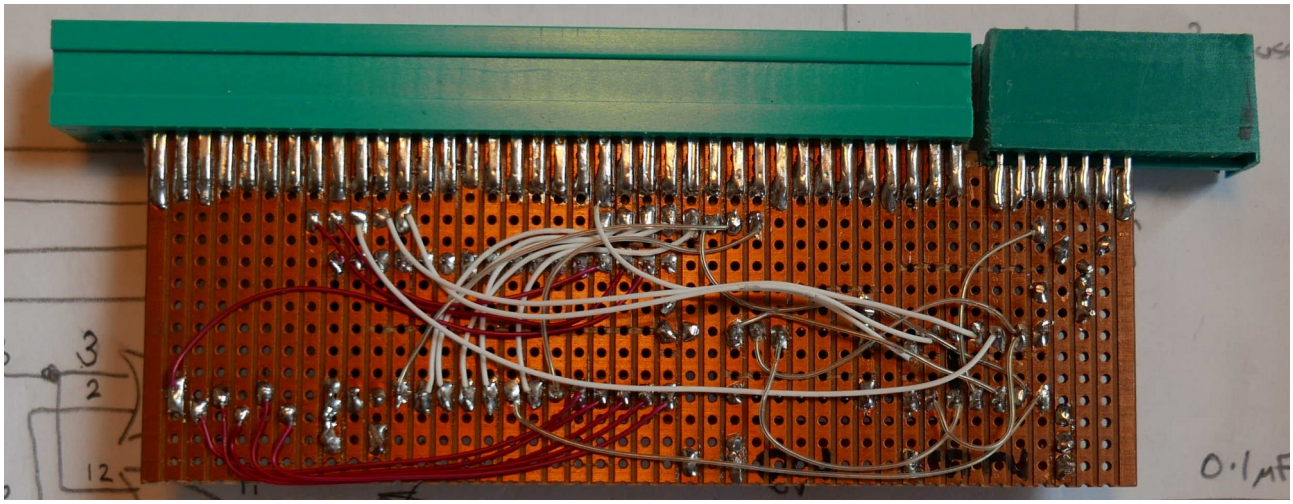

Illustration 3: RAM expansion, bottom-side

DEBUG

Check that your NASCOM 1 is working. From NAS-SYS type “T 1000 1020” and observe that the memory dump is an alias of the NAS-SYS ROM at address 0. Power down.

Fit the expansion board. Check that it is aligned correctly. Populate the 74LS32 and 74LS02 but NOT the RAM chip. Power up the NASCOM 1 and confirm that it still works correctly. Power down.

Move the NASCOM 1 LK5 jumper from “INTERNAL” to “EXTERNAL”. Power up the NASCOM 1 and confirm that it still works correctly. From NAS-SYS type “T 1000 1020”. You should *not* see an alias of the NAS-SYS ROM. Instead you should see junk – probably a mixture of 7F and FF values. Power down.

Populate the RAM. Power up the NASCOM 1 and use the NAS-SYS M command to attempt to modify memory at address 1000. You should have memory all the way up to FFFE. You can load ROM images and execute them from RAM (for example, you can load NASCOM ROM basic to memory at E000).

If any of the steps do not work as described, check your construction for shorts (traces not cut on the veroboard) and wiring errors. If the NASCOM 1 boots successfully and you have an oscilloscope you can use a simple test program to exercise the expansion memory:

```
0C80  21 00 10   LD HL, £1000
0C83  34         INC (HL)
0C84  C3 83 0c   JP £0C83
```

Use the M command to enter the code then use the E command to execute it. Trigger the ‘scope on /WE and look at /MEMEXT, /CE, /OE. They should all be toggling in a tight loop.

A picture of the finished board attached to a NASCOM 1 is shown below.

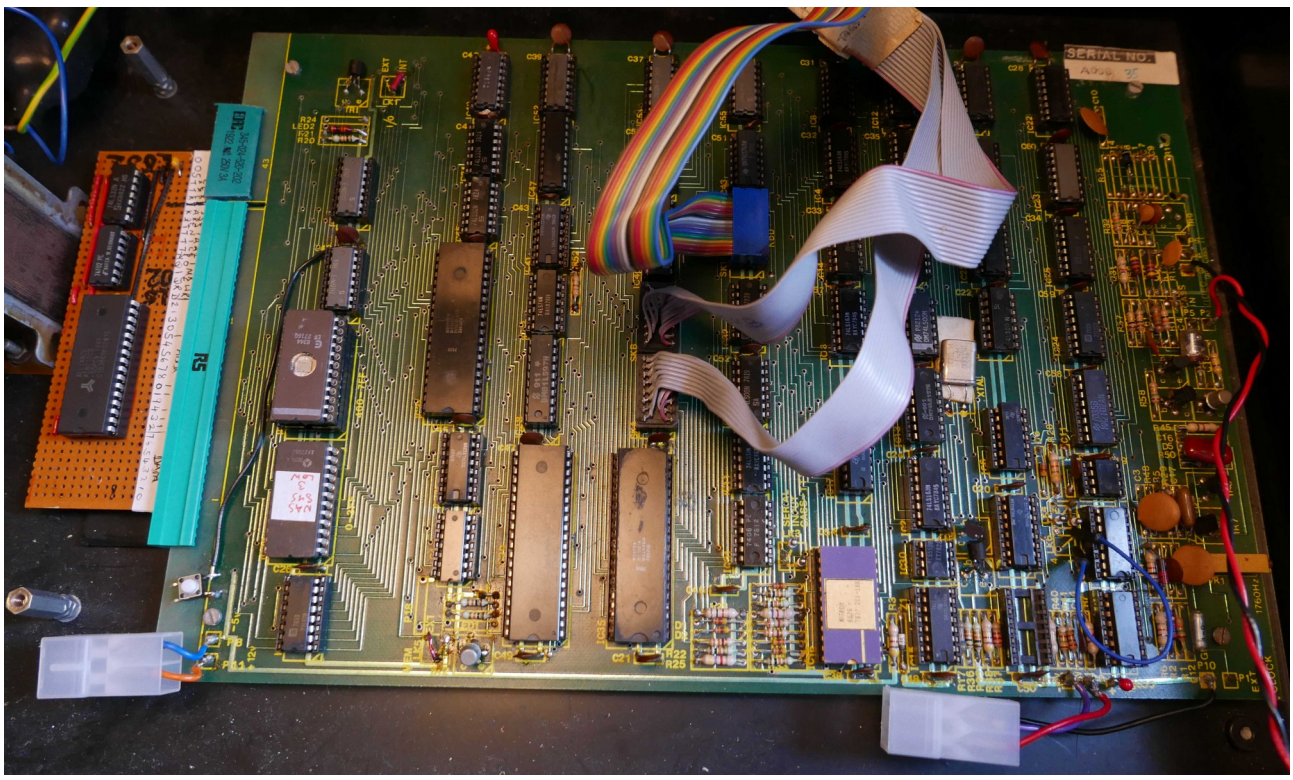

Illustration 4: The finished board, installed

APPENDIX – NASCOM 1 EXPANSION CONNECTOR PINOUT

This was reverse-engineered from the NASCOM 1 “Issue 3” schematic, then corrected when the schematic was shown to be in error. The pinout of the RAM chip is shown lined up to minimise the wiring (it connects 11 of the address lines). A ‘x’ indicates (some of the) cut tracks on the Veroboard.

1	D0					
2	D1					
3	D5 [NOTE1: D2]					
4	D2 [NOTE1: D3]					
5	D4					
6	D3 [NOTE1: D5]					
7	D6					
8	D7					
9	A12					
10	A13					
11	A14					
12	A9					
13	A11					
14	A10					
15	A8					
16	A7					
17	A6					
18	A5					
19	A4					
20	A15	x				
21	A0	x				
22	A3	x				
23	A1	x				
24	A2					
25	/RD					
26	/WR					
27	/MREQ					
28	/M1					
29	/IORQ					
30	/RFSH					
31	/HALT					
32	/WAIT					
33	/INT					
34	/RESET					
35	/BUSRQ					
36	/BUSAK					
37	== POLARISING SLOT ==					
38	CLK					
39	/IOEXT					
40	/MEMEXT					
41	5V					
42	0V					
43	0V					

NOTE1: indicates the INCORRECT assignment shown on the NASCOM 1 “Issue 3” schematic.

AS6C1008

```
          +-------U------+
1  --+ NC       VCC +-- 32
2  --+ A16      A15 +-- 31
3  --+ A14      CE2 +-- 30
4  --+ A12      WE# +-- 29
5  --+ A7       A13 +-- 28
6  --+ A6        A8 +-- 27
7  --+ A5        A9 +-- 26
8  --+ A4       A11 +-- 25
9  --+ A3       OE# +-- 24
10 --+ A2       A10 +-- 23
11 --+ A1       CE# +-- 22
12 --+ A0       DQ7 +-- 21
13 --+ DQ0      DQ6 +-- 20
14 --+ DQ1      DQ5 +-- 19
15 --+ DQ2      DQ4 +-- 18
16 --+ VSS      DQ3 +-- 17
          +--------------+
```