Single-chip RAM expansion for the NASCOM 2

I made the idle assertion that it would be simple to build a single-chip 64kbyte extension for the NASCOM-2, so I decided to design and describe it.

The target chip is an ALLIANCE SEMICONDUCTOR AS6C1008. This is a static 128Kbyte x 8 device, of which half will be used and the other half totally wasted!

This part is available as a 32-pin DIL package for about £3 -- from component distributors or on EBAY. It can run on a supply rail of 2.7V - 5.5V and has a pinout that is "sympathetic" to a 2716.

I propose 2 alternative construction methods:

1/ Mounted directly on the NASCOM main board, in and around one of the 24-pin memory sockets. In this scheme there are some flying leads but no additional components are required.

2/ Connected to the NASBUS. For example, if you have a small veroboard backplane it would be easy to mount the memory there. In this scheme an additional component (an inverter) is required.

For construction method 1, a convenient location is the A1 socket. If necessary, replace the socket with a 24-pin turned-pin socket, and use 2 more sockets to build up the height then finish with a 32-pin socket with its pin 16 aligned to the underlying pin 12 (GND in both cases).

In the wiring diagram below:

- pins marked A1/<number> connect directly down to the corresponding pin of the A1 socket.

- pins 30 and 32 can be wired to VCC (pin 24) of the A1 socket.

- Pins A10, A11, A12, A14, A15, A16 need short flying leads to NASBUS address lines A10, A11, A12, A13, A14, A15. These can be picked up from a combination of the BASIC ROM and IC2 or IC47.

- /WE can be wired to NASBUS /WRB eg on LKB1

- /OE wiring is described in "theory of operation", below.

- /CS can be wired to NASBUS /MREQB eg on IC47/15

- The LINK BLOCK for A1 (LKB1) should be configured with all the links "UP", like a 4118 SRAM.

- The LKS1 jumper block should be configured with NO CONNECTIONS to pins 3 or 5 or 7 (/RAM_G1, /RAM_G2, /XROM).

```
                           AS6C1008
                     +-----------------+
                    1|        O        |32
         NC        --+ NC          VCC +--     **** Wire to A1/24
                    2|                 |31
         ****      --+ A16         A15 +--     ****
                    3|                 |30
         ****      --+ A14         CE2 +--     **** Wire to A1/24
                    4|                 |29
         ****      --+ A12         /WE +--     ****
                    5|                 |28
         A1/1      --+ A7          A13 +--     A1/24 (unused address line)
                    6|                 |27
         A1/2      --+ A6           A8 +--     A1/23
                    7|                 |26
         A1/3      --+ A5           A9 +--     A1/22
                    8|                 |25
         A1/4      --+ A4          A11 +--     ****
                    9|                 |24
         A1/5      --+ A3          /OE +--     ****
                   10|                 |23
         A1/6      --+ A2          A10 +--     ****
                   11|                 |22
         A1/7      --+ A1          /CE +--     ****
                   12|                 |21
         A1/8      --+ A0           D7 +--     A1/17
                   13|                 |20
         A1/9      --+ D0           D6 +--     A1/16
                   14|                 |19
         A1/10     --+ D1           D5 +--     A1/15
                   15|                 |18
         A1/11     --+ D2           D4 +--     A1/14
                   16|                 |17
         A1/12     --+ GND          D3 +--     A1/13
                     |                 |
                     +-----------------+
```

Theory of operation (piggy-back construction)

===

The NASCOM has ROM (NAS-SYS) in the low 2Kbytes, then 2Kbytes of RAM (workspace and video), then a hole, then 8Kbytes ROM at the top of the address space (BASIC). The hole is 52Kbytes and the idea is to fill the whole of that region with RAM.

Connecting MREQ to the expansion RAM's chip-select means that it will be selected for any memory access (but not I/O accesses). Connecting WE to the expansion RAM's write-enable means that it will be written to for any write. For example, it shadows the workspace RAM and any write to the workspace RAM will also write to the expansion RAM.

The two remaining areas of difficulty are:

- How to control the expansion RAM's output enable. This needs to be asserted for any read access to the 52Kbyte "hole" but must not be asserted for any other addresses becasue that would cause a bus clash (the fact that both devices might drive the same data value will not help us!!).

- How to control the enable on buffer IC45 so that the RAM's data is driven (via the NASBUS) to the CPU.

-> the expansion RAM's /OE can be connected to IC44/12 (schematic sheet 3). That signal is high when another device (NAS-SYS, BASIC) is enabled on the same segment of the data bus. Using this as /OE means that the expansion RAM is the default driver of this part of the data bus.

-> the /RAM_G1 signal is connected to gnd. This means that the /RAM_DISABLE signal is always asserted, disabling any NASBUS RAM resources, because all RAM is being provided via buffer IC45. IC45 is enabled for all reads except video RAM reads and during refresh cycles.

Theory of operation (connected to the NASBUS)

The only difference (apart from more wires to connect) is that now the expansion RAM's data bus will be connected directly to the NASBUS -- the connections will be to the B side of buffer IC45 rather than to the A side. As a result, the enable on buffer IC45 must be controlled differently.

-> the expansion RAM's /OE can be connected to NASBUS signal /RAM_DISABLE through an inverter. This will enable the RAM when no main-board resource is driving the data bus.

-> the /RAM_G1 signal is disconnected (the on-board pullup will take it high). This means that buffer IC45 is enabled for NAS-SYS and BASIC ROM reads and disabled for video RAM reads, for refresh and for any other reads, leaving the 52Kbyte "hole" for the expansion RAM.

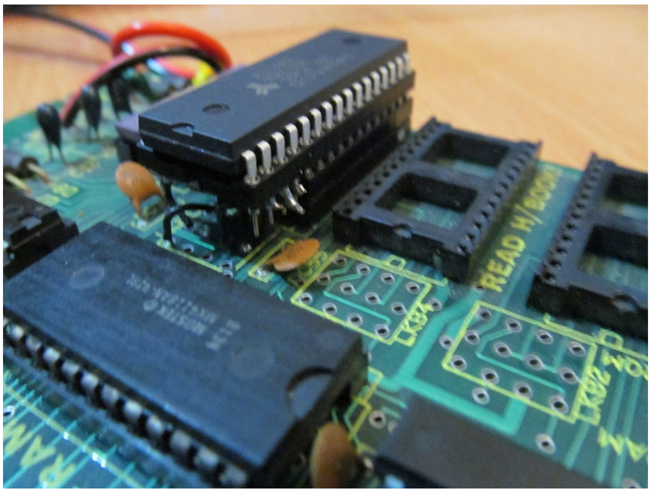
Construction notes and Photos (piggy-back construction)

mick@mixtel.co.uk wrote: It's not a particularly easy mod to do, even if you have a very fine soldering iron. I was fortunate in that the link blocks hadn't been fitted so I was able to thread my wires through those holes to the back. If you can't do this then getting access to most signals may involve soldering onto IC pins. I'd advise removing the link block anyway, just to make some space.

Don't attempt to unsolder the A1 socket without removing adjacent ICs, including the BASIC ROM. It's surprising how much heat there is around. I used a heat gun and ended up scorching the PCB very slightly. It would probably be kinder to rip the top off the socket then unsolder the individual pins, with hindsight.

I think that, if I were to do it again, I would start by soldering very fine flexible wires onto the 32-pin socket and feeding them through holes etc before pressing the socket into place - and 2x 24-pin sockets is just nice for height. Take the +5 from the link block pad, not from pin 28 of the RAM chip. It's a lot easier as you are trying to solder onto a pin that's going to be plugged into a socket otherwise.

To pick up a GND signal for LKS1 I scraped a bit on the leg of the adjacent capacitor. Trying to solder onto the copper track with an 8W USB soldering iron was futile.



Questions or comments to the author:

Neal Crook, foofoobedoo@gmail.com

The definitive source of this document is my github vault:
https://github.com/nealcrook/nascom/blob/master/doc/nascom2_64k_expansion.pdf

mick@mixtel.co.uk produced a beautiful construction diagram which you can find here: https://groups.io/g/Nascom-Computers/files/Hardware%20-%20DIY/N2mods.pdf

```
Revision History
================
22Jun2019 V1.0
25Jun2019 Correction to /OE connection for piggy-back construction.
03Jun2020 Add construction photos courtesy of mick@mixtel.co.uk; the piggy-back
          version of the design is now “proven”.
```