Two-Chip 32k in the TI-99/4A Console – by Mike Brent
Updated: 27 Jul 2016

The purpose of this modification is to provide a simpler, faster to install, and more reliable 32k mod for the TI-99/4A than the conventional 16-bit mods. While this mod does not provide the performance enhancement of the 16-bit versions, it requires far fewer connections and does not require any cut traces – it can be removed simply by removing the two chips and the wires.

The mod consists of a single 32k x 8 static RAM chip, in my case I used the CY7C199-35PC. This is a 35nS part – slower parts should also work fine. It also uses a 74LS21 Dual 4-Input AND gate. Both of these parts are available at Jameco, as of Nov 7, 2014, with these part numbers:

CY7C199 – Part number 242376 - $1.95 each - https://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_242376_-1

74LS21 – Part number 47108 - $0.59 each - https://www.jameco.com/webapp/wcs/stores/servlet/Product_10001_10001_47108_-1

Also make sure you have some fine wire, soldering iron, and solder. This will take 1-2 hours for most people. You can see that the cost will come in under $3 – that's a far cry from 30 years ago!!

Two "piggyback"s are used to attach the ICs to the console and to reduce the number of wires needed. Unfortunately not as many lines match up as I would like, in part because most of the chips inside the console are on the wrong side of the multiplexer. In that respect the 16-bit mod was almost easier to layout! But never fear, we still don't have too far to go. Now, regarding finding the chips – I have a photo at the end you can reference. Also, when I say 'beside' in this document, I am only talking about next to the long end of an IC, never the short ends!

First we'll hook up the 74LS21. Bend out all the pins except pins 12 and 14 (use the diagram above if you need help mapping them). I find it helpful to trim the bent pins as well, so that only the fat part remains. Don't trim the un-bent pins!

First we need to connect some of the pins on this chip together. Run a short wire connecting pins 2 and 12 (note this pin is not bent out!). Run a second short wire connecting pins 5 and 10. Finally, we need a third run connecting pins 1, 4, and 14 (note this pin is not bent out!)

Next we need to find a 74LS138 on the TI motherboard. If you look near the edge connector, there are two of them side-by-side. We want the one closest to the edge connector, and it's right between a 74LS04 and a 74LS244.

We need to connect some wires to the 74LS138 on the board before we piggy back onto it (it'll be harder if we wait!) Attach short wires to the following pins (leave them floating, don't connect them together): 7, 8, 9, 10 (the four pins at the end farthest from pin 1).

Now take your 74LS21, and line it up with the top of the 74LS138. Pin 1 goes the same

direction on both chips! Pin 14, which is sticking down, should overlap pin 16 of the 74LS138, and there should be one extra row of pins sticking out at the far end of the 74LS138 (where you attached the wires). Solder pins 14 and 12 to the 74LS138 below (they will end up at pins 16 and 14).

Now, working around the 74LS21, we'll hook up the rest of the wires.

Pin 1 should already be connected to pins 4 and 14.

Pin 2 should already be connected to pin 12.

Pin 3 does not get connected.

Pin 4 should already be connected to pins 1 and 14.

Pin 5 should already be connected to pin 10.

Pin 6 needs a long wire attached – long enough to reach halfway across the 9900. Don't hook up the other end right now.

Pin 7 should be attached to the wire coming from the 74LS138's pin 8.

Now up the other side:

Pin 8 needs a long wire attached – long enough to reach halfway across the 9900. Don't hook up the other end right now.

Pin 9 should be attached to the wire coming from the 74LS138's pin 7.

Pin 10 should already be attached to pin 5, **HOWEVER**, you must also attach it to the wire coming from the 74LS138's pin 9.

Pin 11 does not get connected.

Pin 12 should already be connected to pin 2 (and 12 is soldered to the 74LS138 directly below).

Pin 13 is attached to the wire coming from the 74LS138's pin 10.

Pin 14 should already be connected to pins 1 and 4.

That completes the 74LS21.

Technical Explanation
This chip provides the address decoding and selection logic for the RAM chip. The 74LS138 underneath is called a 1-of-8 decoder – it takes the 3 top address lines in the TI's address bus, and outputs a single line signal indicating which area of memory is being accessed,

divided into 8k blocks. 8K x 8 = 64k, so this covers the entire memory space. The address bits are entered on pins 1-3, with the following table showing which pin goes active (low) for the corresponding address:

7	>E000
9	>C000
10	>A000
11	>8000
12	>6000
13	>4000
14	>2000
15	>0000

In the TI console, the lines corresponding to RAM addresses, specifically >2000,>A000,>C000 and >E000, are not connected. The 32k card in the PEB uses it's own decode logic to decide when to respond. This lets us lift the signals for our own use. Of the 32k, the range >2000->3FFF represents the "low" 8k of RAM, and >A000->FFFF is the "high" 24k of RAM. So we need to respond to all of these signals.

The 4-input AND gate makes this quite easy. To select the RAM chip, we need a 'low' signal, and the 74LS138 outputs 'low' on the active range. An AND gate outputs 'high' if all of its inputs are 'high', and 'low' otherwise, so, all we do is feed the inputs for each of the four areas we are interested in. That gives us this truth table:

>2000	>A000	>C000	>E000	Output
1	1	1	1	1
0	1	1	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	0

Since any other memory area will result in the four lines we care about being high, this is enough to select when our 32k chip should respond.

The second problem we have is making sure all 4 areas select a different 8k block in the 32k chip. In the 16-bit console mod, this problem is ignored by using 64k chips and just wasting half the space. I admit it took some head scratching to work it out, but it turns out that the second half of the AND gate was just what we needed!

The address lines A0, A1 and A2 select one of eight blocks of memory. For the four banks we are interested in, A1 gives us the cleanest division. It is 0 for banks >2000 and >A000, and it is 1 for banks >C000 and >E000. In order to generate the other bank divisor, we only need to

differentiate these two areas. By ANDing the selects of >2000 and >C000, we get that result, getting 0 for banks >2000 and >C000, and 1 for banks >A000 and >E000. This is confusing in text, but hopefully this table makes it more clear:

Address	A1	>2000 Select	>C000 Select	2000 & C000	Result (A1/&)
>2000	0	0	1	0	00
>A000	0	1	1	1	01
>C000	1	1	0	0	10
>E000	1	1	1	1	11

We feed the selects for >2000 and >C000 into the inputs of the other side of the dual-AND gate, and tie the two extra inputs high so that they always read as '1'. The two result pins will be used to select the 8k block in the 32k chip to be used for each area. (Note that since the 'select' will not be active for other addresses, it doesn't matter in the above table what their results would be).

Back to work!

Next we need to install the RAM chip, the CY7C199. Due to its size, and the need to go after the multiplexer on the 8-bit bus, there are not many places to piggyback the chip and reduce the number of wires. Ultimately I decided to take the address bus from the 9900, since we got one extra unbent pin versus mounting it on the data bus directly.

Bend out pins 9, 11-14, and all pins on the second side of the chip (15-28). 9 pins will be left down. Clip the thin parts off of the bent legs, but not the unbent ones!

Locate the 9900. It should be pretty easy – it is the most prominent chip on the board! We are going to attach the RAM along the same side as pin one (that is, if you are looking at it so that the part number is readable, it will go along the bottom.

You will have to count carefully, there is no simple alignment here. Pin 1 of the CY7C199 will face the same direction as the 9900, but it will attach at pin 14 (which is A10 of the 9900). Solder the 9 straight pins from the CY7C199 to the 9900 pins below. Double-check your counting before you solder, it's a pain to remove!

Now we can walk around the CY7C199 and do the rest of the pins:

Pin 1 should already be connected to the 9900 below.

Pin 2 should already be connected to the 9900 below.

Pin 3 should already be connected to the 9900 below.

Pin 4 should already be connected to the 9900 below.

Pin 5 should already be connected to the 9900 below.

Pin 6 should already be connected to the 9900 below.

Pin 7 should already be connected to the 9900 below.

Pin 8 should already be connected to the 9900 below.

Pin 9 should be attached to the long wire you ran from the 74LS21 pin 6.

Pin 10 should already be connected to the 9900 below.

For the next part, find the 74LS245. If the 9900 is facing you so that you can read the text on it, it is right above the top left corner of the 9900, with a 74LS373 right next to it. We will be attaching the data bus here, plus one extra signal, for an eventual total of 9 wires. Once you have located it, we can proceed with the CY7C199 pins.

Pin 11 should be connected to the 74LS245 pin 18.

Pin 12 should be connected to the 74LS245 pin 17.

Pin 13 should be connected to the 74LS245 pin 16.

Pin 14 should be connected to the 9900 pin 40 (this is a ground).

Now up the other side:

Pin 15 should be connected to the 74LS245 pin 15.

Pin 16 should be connected to the 74LS245 pin 14.

Pin 17 should be connected to the 74LS245 pin 13.

Pin 18 should be connected to the 74LS245 pin 12.

Pin 19 should be connected to the 74LS245 pin 11.

Pin 20 goes to the long wire connected to the 74LS21 pin 8.

For the next pin, we need to find a 74LS00. Again, if the 9900 is facing you so that you can read the text, it is below the left edge of the 9900, next to a 74LS74. This is a confusing area of the board, if you look to the left of the correct chip, after a small gap there are two more 74LS00... we want the one right beside the 74LS74. Once you have found it, we have one wire to attach to it.

Pin 21 goes to the 74LS00, pin 11.

Pin 22 goes to the 74LS245 pin 10.

Pin 23 goes to the 9900 pin 10.

Pin 24 goes to the 9900 pin 11.

Pin 25 goes to the 9900 pin 12.

Pin 26 goes to the 9900 pin 13.

For the next pin, look back near the edge connector for the 74LS244. It is right beside the edge connector and has no chips immediately beside it.

Pin 27 goes to the 74LS244 pin 11.

Pin 28, finally, goes to the 9900 pin 59.

Technical Explanation

This is mostly a pretty direct wiring of the RAM chip to the appropriate pins. Except for the fact that TI numbers their bits backwards (ie: they consider bit 0 to be the MSB while everyone else considers bit 0 to be the LSB), it's largely a direct mapping.

On the address bus, address lines A1 through A12 map to the TI's A3 through A14 – this represents 4k of direct mapping. The least significant bit, A0/TI-A15, doesn't exist on the 9900. TI-A15 is generated by the multiplexer circuit, so we take it from the 74LS00 as a convenient point to tap it. A14, the most significant bit on the RAM chip, is taken from the TI's A1 (the second most significant bit), and the RAM's A13 is generated from the AND gate we added by ANDing the >2000 and >A000 selects, as described above. Thus, these two bits select an 8k range in the RAM, while the remaining address bits directly address inside that range.

The data bits D0 through D7 are directly connected to the TI's D7 through D0, matching the bit reversal. This is not really important for RAM, but it maintains consistency. We pull them from the 74LS245, which is a buffer chip attached to the multiplexer, making sure we are on the 8-bit side of the data bus.

Chip Enable is generated from the 4-input AND gate as described above, and is low any time any RAM expansion address is addressed. Output Enable is permanently tied to ground so is always active.

Finally, Write Enable is taken from the 74LS244 as a convenient place to tap it. We must use this derived value and not the bit from the 9900, as the multiplexer generates two pulses for a single 9900 memory cycle.

AND THAT'S ALL!

Double and triple check your pin counting, wiring, good solder joints, etc. Take a breath, power up the system, and try SIZE in Extended BASIC to see if the RAM shows up! For best results, run a memory tester like the Corcomp Peripheral Test cartridge, if you have access to it. Have fun with your 32k console!
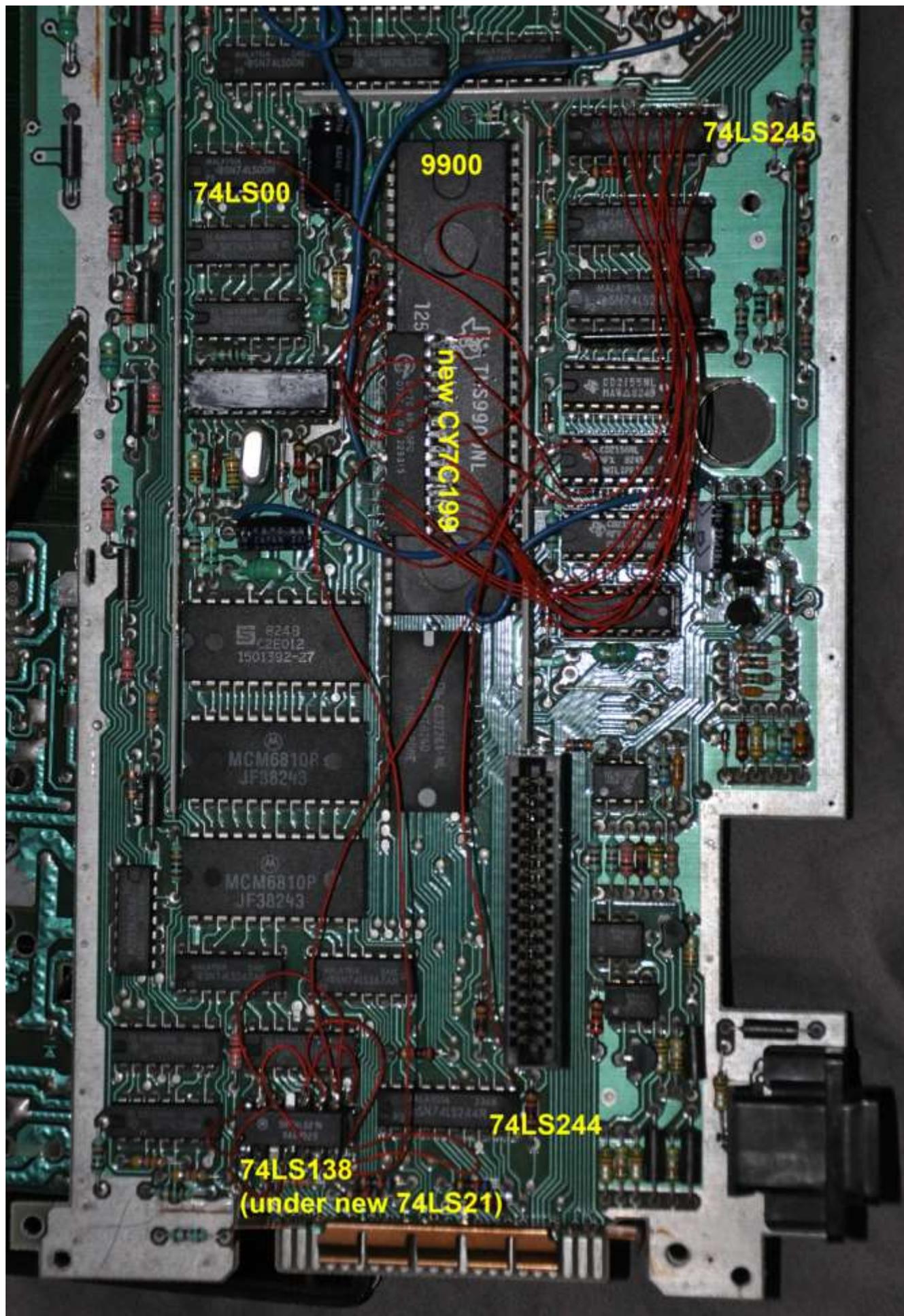
Pinouts of various chips:
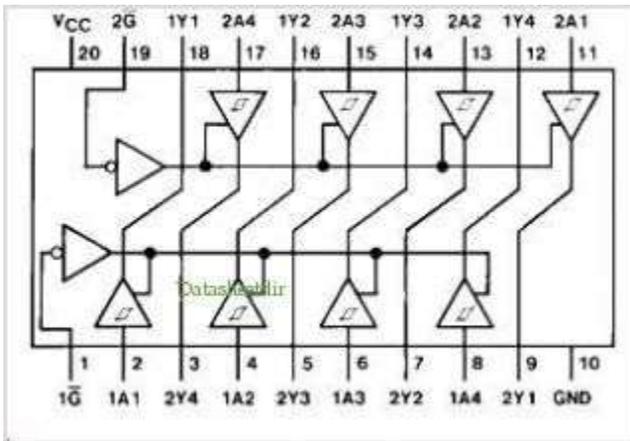
Figure 1 - 74LS244
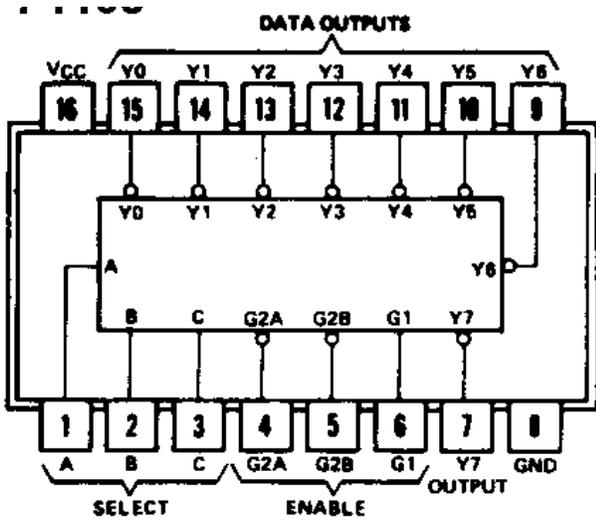
Figure 2 - 74LS138
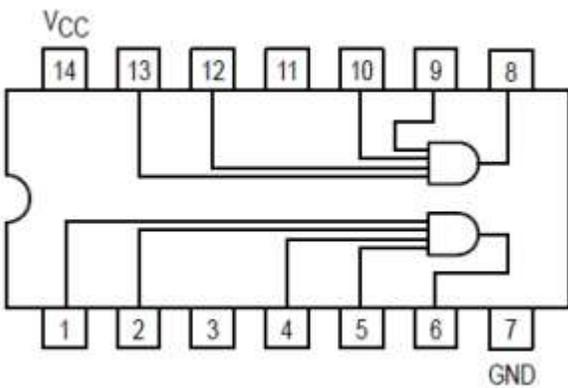
Figure 3 - 74LS21
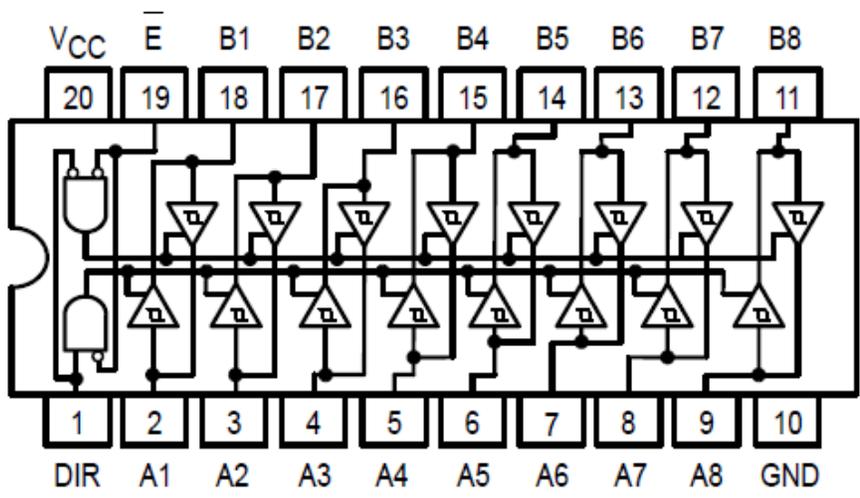
Figure 4 - 74LS245
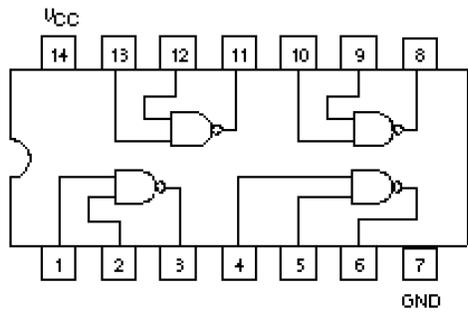
Figure 5- 74LS00
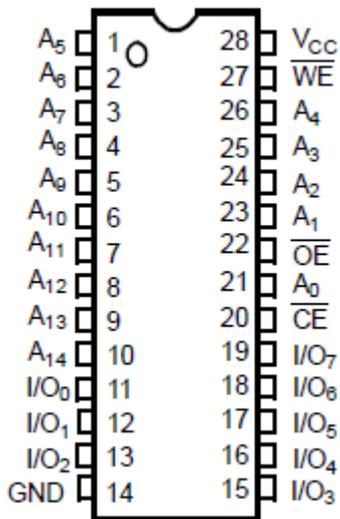
Figure 6 - CY7C199

TMS 9900 PIN ASSIGNMENTS

Pin	Name		Pin	Name
1	V_BB		64	HOLD
2	V_CC		63	MEMEN
3	WAIT		62	READY
4	LOAD		61	WE
5	HOLDA		60	CRUCLK
6	RESET		59	V_CC
7	IAQ		58	NC
8	φ1		57	NC
9	φ2		56	D15
10	A14		55	D14
11	A13		54	D13
12	A12		53	D12
13	A11		52	D11
14	A10		51	D10
15	A9		50	D9
16	A8		49	D8
17	A7		48	D7
18	A6		47	D6
19	A5		46	D5
20	A4		45	D4
21	A3		44	D3
22	A2		43	D2
23	A1		42	D1
24	A0		41	D0
25	φ4		40	V_SS
26	V_SS		39	NC
27	V_DD		38	NC
28	φ3		37	NC
29	DBIN		36	IC0
30	CRUOUT		35	IC1
31	CRUIN		34	IC2
32	INTREQ		33	IC3

Figure 7- TMS9900NL