Another KIM-1 Expansion

Packaging the KIM-1, adding a TTL serial interface and adding 24K more memory for less than $300 using 2114s are the subjects covered by this KIM-1 expansion article.

John M. Blalock
3054 W. Evans Dr.
Phoenix AZ 85023

Can the MOS Technology KIM-1 be economically expanded to become a versatile high-level-language system? Is the KIM really just a "trainer board," to quote a Phoenix Byte Shop salesman? Why start with a KIM? This article will explain why I started with a KIM and attempt to answer these questions. It will explain the expansion steps I went through and describe in detail the most recent addition, a 24K static RAM memory board that cost less than $300. Even if you don't have a KIM (or a SYM) you might get a smile and perhaps some information from reading about my problems and my thoughts about 2114 static RAM ICs.

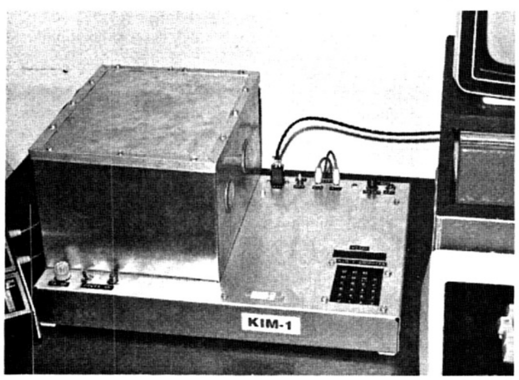


Photo 1. The author's KIM is in there?

Background

My interest in having my own computer dates from 1972, when I learned that my employer was paying just over $2000 for a PDP-8 with 4K of core. "One of these days I'll be able to afford one!" I thought.

Time went by, and I never got the money. Along came microprocessors and Altair, but I still couldn't come up with enough. Yes, I could stretch the budget and buy an Altair, Imsai, etc., but the price of the computer itself was just a start. To do much of anything with it would require adding memory, a terminal, cassette and serial interfaces. That SWTP kit and terminal looked attractive, but it was still over $1000 with the terminal, cassette interface and enough memory to run BASIC.

While I was still saving money, a friend got a MOS Technology KIM-1. I started to hear regularly about all the things it could do. A charter subscriber to both *Byte* and *Kilobaud*, I had already read several articles about the KIM. It had a built-in keypad, display, ROM monitor, TTY interface, cassette interface, etc. It could be expanded without too much difficulty. Only $245 plus a power supply and I could get started in personal computing! I could put off buying the terminal and more memory until later while still learning and having fun with the unexpanded KIM.

Getting Started

One Monday night, another amateur-radio operator listed a KIM-1 on our local two-meter swap net at a price I couldn't refuse. He needed cash badly and had mine the next day. That night I was practicing moon landings on my *own* computer. I played blackjack, Bandit, decoded Morse code, wrote several machine-language routines and more. Every month I looked forward to the next issue of *Kilobaud* to find out what new application or program someone had come up with for the KIM.

One of the first things I decided to do was to package the KIM along with its power supply. I put both in a 13 x 17 x 2 inch chassis. I purposely enclosed the power transformer area on top of the chassis in an 8½ x 11 x 6 inch box that was large enough to also house four or five S-100-size boards. See Photo 1 for a view of my KIM. The KIM board is bolted to the right underside of the chassis using 3/8-inch spacers. The keypad is readily accessible through the large cutout, and the display can be viewed through its cutout.

The next project was to add the S.D. Sales 4K memory board according to the article by Bob Haas in the April 1977 *Kilobaud*, "KIM-1 Memory Expansion," p. 74. That done, I bought a copy of Tom Pittman's Tiny BASIC, borrowed an ASR-33 Teletype and was really computing until I had to return the Teletype. Fortunately, I belonged to a company-sponsored computer club that was putting together a CRT terminal kit that was within the reach of my pocketbook—an ASR-33 wasn't. Work on the KIM slowed until I got the terminal working.

CRT Interface

The terminal's serial interface was designed to work directly with a modem like Ron Lange's in the November 1977 *Kilobaud* ("Build the $35 Modem," p. 94). Not wanting to use the parallel interface and lose the use of the KIM's monitor support of the 20 mA TTY interface, I decided to connect the TTY port to the terminal's TTL serial lines. See Fig. 1 for what worked for me: wire, two resistors and one IC; it couldn't be more simple.

So now I had a working system that not only used the onboard keypad and display to run all the programs in *Kilobaud* and the *First Book of KIM*, but I was also able to write short programs in Tiny BASIC. Soon it seemed that almost every program I wrote used up all available memory before it was completely written. Also, my teenager wanted to put more games on the system, most of which required either string-handling capability, subscripted variables or both. It was time to expand again.

Adding Even More Memory

I wrote Microsoft to ask about a full BASIC for the KIM. They referred me to Micro-Z, Box 2426, Rolling Hills CA 90274. Bob Kurtz at Micro-Z said that they had what I wanted. It required 9K bytes starting at hexadecimal 2000 just for the interpreter. 16K would be enough to handle the interpreter and short programs, but 20 or 24K would be much better.

The January 1978 issue of *Kilobaud* had a good article by John Eaton on interfacing the KIM to the S-100 bus ("Growing with KIM," p. 36). I could use his interface, get two Godbout 8K board kits and two S-100 connectors, and add 16K for just about \$300. That seemed preferable to spending about \$400 for 16K using 4K boards like those from Atwood. On the other hand, why not take advantage of the three-for-\$375 price break from Godbout and add 24K for about \$400? Of course, I could go with only two 8K boards, un-modify my S.D. Sales 4K board and have 20K

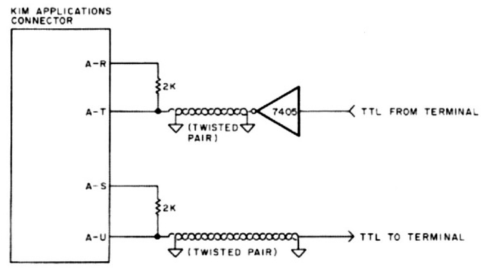
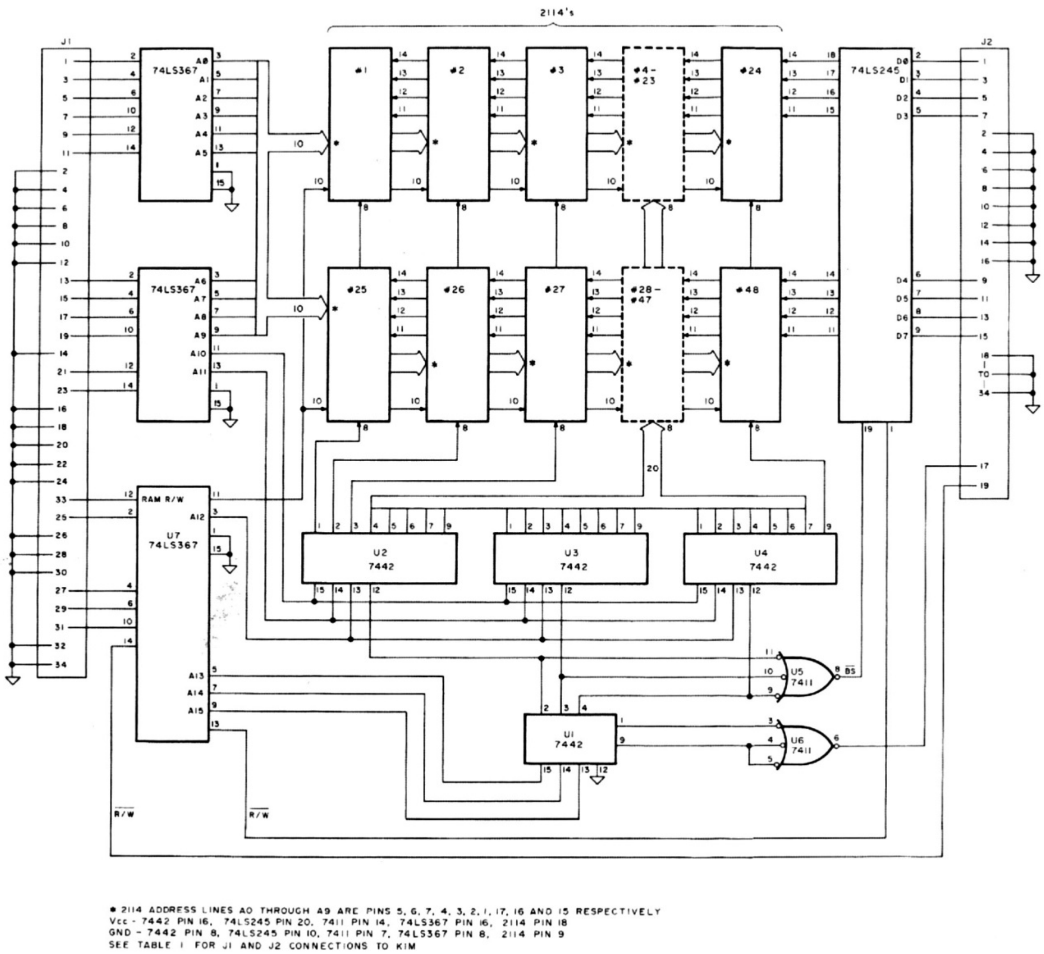


Fig. 1. Circuit to convert the KIM TTY port from 20 mA to TTL.



* 2114 ADDRESS LINES A0 THROUGH A9 ARE PINS 5, 6, 7, 4, 3, 2, 1, 17, 16 AND 15 RESPECTIVELY
Vcc - 7442 PIN 16, 74LS245 PIN 20, 7411 PIN 14, 74LS367 PIN 16, 2114 PIN 18
GND - 7442 PIN 8, 74LS245 PIN 10, 7411 PIN 7, 74LS367 PIN 8, 2114 PIN 9
SEE TABLE 1 FOR J1 AND J2 CONNECTIONS TO KIM

Fig. 2. Schematic of the 24K memory board design.

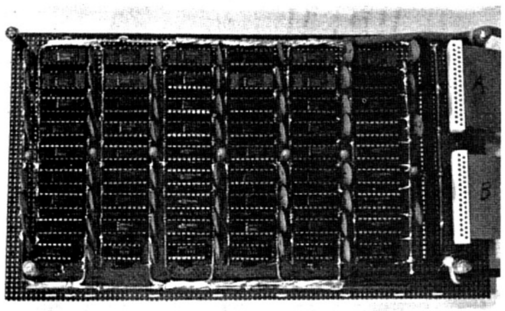


Photo 2. The 24K memory board.

for about $300 additional. This, however, would cost me the use and advantage of the programs I had from 0400 to 13FF.

Whichever way I went, I'd have to mount more S-100 connectors in the cabinet. Ever try to find a four- or five-slot S-100 motherboard? Wouldn't one of those Dynabyte 32K static RAM boards be nice? My income-tax refund wasn't *that* large, but then, with a board like that, I wouldn't need the motherboard. If 2114s didn't cost so much, I could probably fit 24K of them on one S-100 size board....

I ruled out building my own dynamic memory board because of the refresh problem. There are no open slots for refresh in the timing of the 6502 used in the KIM. Apple uses dynamics, but they refresh them as part of their display updating.

I agree with others that the 2114 will be the next industry standard RAM, replacing the 2102 (see Digital Research Corporation's ads). The 2114 is fully static. It is organized as 1K by four bits. Only two 2114s are needed for 1K byte of memory. The main interfacing difference between the 2114 and the 2102 is that the 2114 has four bidirectional data lines. The 2102 has both a data-in and a data-out line. Since it can't be taking in data (a write operation) at the same time data is going out (a read operation), one line for each data bit is all you is need.

The 2114 is an 18-pin IC... compared to only 16 pins for the 2102. You get four times the memory capacity for just over one fourth the size. Prices for 2114s, even though they were going down, were still around $10 each in quantity. It looked as though I'd stick with 2102s.

While still trying to decide which way to go, I stopped to see Steve at Semiconductor Surplus (2822 N. 32 St., Phoenix AZ 85018) one Saturday. I spotted a sign: "2114s—$8.25 each." Steve assured me that the 2114s he had were not too slow. They were 200 ns versions! I told him that I liked his price but would have to wait until they came down to $5 each for 450 ns versions. Then they would be price-competitive with 2102s. Steve asked how many I needed. I said, "Forty eight, at least." He said, "I'll sell you 48 for $5 each." After verifying his guarantee, and the compatibility of the 2114-2 timing with that of the KIM, I went home with $252 worth of ICs (tax included).

Design of the 24K Board

Now to design a circuit for a 24K memory board to interface to the KIM. I decided to skip the S-100 interface compatibility goal as being an extra expense, but did decide to buffer every KIM line to the board. Fig. 2 is the result of my design effort.

The 74LS367s and 74LS245 would provide the buffering, the 7442s would provide address decoding and the 7411 would provide board select signals for the 24K board and for the KIM memory. I used 7442s instead of 74LS138s or other decoders due to their price, availability, available power and speed comparable to the other ICs. The 74LS245 is a bidirectional 8-line Tri-state transceiver. The level on pin 1 controls the data direction. Bringing pin 19 high causes the '245 to act essentially as an open circuit to the data lines from each direction. It is ideal for applications such as this. The 74LS367s provide negligible load to the KIM lines, so they are permanently gated on by grounding pins 1 and 15.

The '367s constantly provide address information (A_0-A_9) to the 2114s. This information is ignored until $\overline{CS}$, pin 8, of the 2114 goes low. 7442 U1 divides the 64K KIM memory space into eight 8K blocks. Either the $8K_0$ or $8K_7$ signal from U1 is used to enable the KIM memory onboard decoder. Enabling the KIM for just the lower 8K will work for its RAM and the S.D. Sales 4K addition, but the reset and interrupt vectors stored in KIM ROM are assumed to be in the upper 8K of memory. Therefore both $8K_0$ and $8K_7$ are used to enable the KIM.

7442s U2, U3 and U4 divide $8K_1$, $8K_2$ and $8K_3$, respectively, into 1K blocks. Since two 2114s comprise a 1K block, each of these 7442s provides the $\overline{CS}$ signal to sixteen of the 2114s. Only one pair of 2114s can be selected at any one time and respond to the address information from '367s.

Data will be written into or read from the selected 2114s depending on the state of 2114 pin 10, $\overline{WE}$. This level is controlled by the KIM RAM R/W signal, just like the KIM onboard 2102s.

The 74LS245 is enabled by the signal from U5. This signal is low if the KIM is addressing the 24K board (2000 to 7FFF hexadecimal). Data direction through the enabled '245 is controlled by the KIM $R/\overline{W}$ signal, which overlaps RAM R/W, thus eliminating propagation delay concerns.

Construction

According to my calculations, I would be able to fit this circuit on one of the S-100-compatible prototype cards. I planned to wire-wrap the connections. Making an etched board was out because the number of runs would require a double-sided board. This was more than I wanted to attempt. I couldn't find an S-100 prototype board that was compatible with the circuit; 74LS245s were unavailable then due to rumored poor manufacturing yield, and 18-pin wire-wrap sockets cost at least 50 cents each! There had to be a better way than spending $24 just for sockets. Remember, I was trying to stay within a limited budget.

I circumvented the lack of a 74LS245 by substituting two 8T28s and a 7402. I got forty eight 18-pin solder tail sockets for $12, and a large piece of .100 inch perfboard for $7. I borrowed a friend's wiring pencil and put it all together. See Fig. 3 for the schematic changes made to use the 8T28s and

J1	KIM	J1	KIM	J2	KIM	J2	KIM
1	E-A	18	GND	1	E-15	18	GND
2	GND	19	E-L	2	GND	19	E-W
3	E-B	20	GND	3	E-14	20	GND
4	GND	21	E-M	4	GND	21	SPARE
5	E-C	22	GND	5	E-13	22	GND
6	GND	23	E-N	6	GND	23	SPARE
7	E-D	24	GND	7	E-12	24	GND
8	GND	25	E-P	8	GND	25	SPARE
9	E-E	26	GND	9	E-11	26	GND
10	GND	27	E-R	10	GND	27	SPARE
11	E-F	28	GND	11	E-10	28	GND
12	GND	29	E-S	12	GND	29	SPARE
13	E-H	30	GND	13	E-9	30	GND
14	GND	31	E-T	14	GND	31	SPARE
15	E-J	32	GND	15	E-8	32	GND
16	GND	33	E-Z	16	GND	33	SPARE
17	E-K	34	GND	17	A-K	34	GND

Table 1. Connections to the KIM. A – 1 = Applications Connector, Pin 1; E – 1 = Expansion Connector, Pin 1.

Photo 2 for a picture of the finished 24K board.

The board measures 5½ by 10 inches. Ribbon cables from the KIM connectors plug onto two connectors mounted on the right side of the board. Next to them, oriented vertically, are the 74LS367s and 8T28s. The rest of the ICs are oriented horizontally in nine rows of six columns each. The top eight rows are the 2114s. The 7442s, 7411 and 7402 are located in the bottom row. Power and ground are routed next to each IC socket with #12 bare copper wire, liberally bypassed with fifty one .01 uF disc ceramic capacitors and seven 10 uF tantalum capacitors.

My junk box provided some #10 stranded wire, which was soldered to the power and ground buses and connected to the 5 V regulated supply. The ribbon cables were made from a 36-inch 34-wire jumper cut in half. Every other wire was connected to ground at the 24K board and at the KIM to minimize the possibility of cross-talk or pickup.

The power and ground buses are held in place by #22 wire loops to the board and by #22 wire connections soldered to the adjacent voltage or ground pin at each socket. The sockets are attached by these same connections plus a drop of epoxy on each side of every socket. There are four holes in the .100 inch perfboard between each column of 18-pin sockets . . . and one hole between rows.

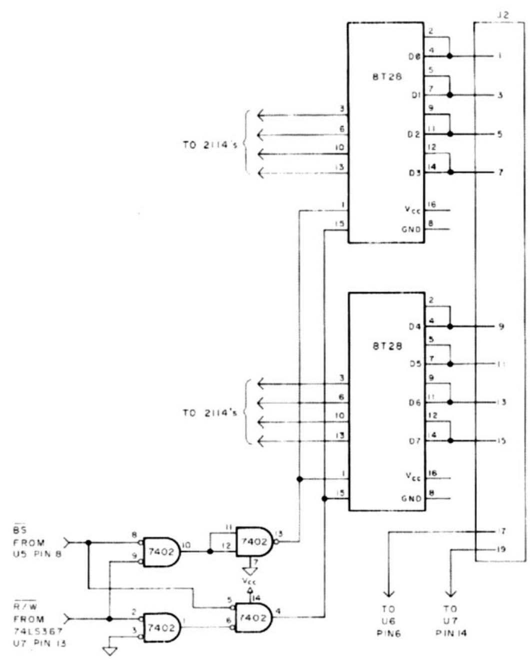


Fig. 3. Modifications to use 8T28s in place of the 74LS245.

Wiring approximately 900 joints and soldering them was tedious, but not as bad as I expected. The wiring pencil worked as advertised, but the wire was a little hard to solder. At least a 37 Watt iron is needed to melt through the insulation and get a good joint. I buzzed out every connection and only found one such bad joint!

Checkout

After I found a wiring error and a few bad 2114s, the added memory worked just as I expected it would. The board draws under 3.5 A from my 5 V supply. Total cost was almost $290, still within budget.

Steve's guarantee on the 2114s was tested and proven good. The Micro-Z version of Microsoft's 6502 BASIC is even better than originally advertised. They've added a Hypertape SAVE routine to speed up the cassette interface and included a data-save/data-load feature. The KIM has 15183 bytes free for BASIC programs, which should be enough to hold almost all programs I care to type in. Speed is almost twice as fast as a Radio Shack TRS-80 with Level II BASIC.

Conclusions

I've spent almost as much for my system as the price of a TRS-80 with 16K RAM and Level II BASIC, but it has some features the TRS-80 doesn't. It's been fun and educational getting here. Yes, the KIM can be expanded into a versatile high-level-language system. No, it's not just a "trainer board." Now for a printer and floppy disk.

When I wrote this article, 450 ns 2114s were being advertised in *Kilobaud* as low as $5.50 each in quantities of 100 and up. 450 ns is fast enough for KIM. By the time you read this, I expect that 2114s will be generally available for $5 or less, so you too can break that $100-per-8K price barrier and add more memory to your KIM. ■