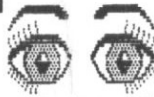


**SUPERAMIS**

Date: 8 Apr 1975  
 Drawn By: [Name]  
 Checked By: [Name]  
 Approved By: [Name]  
 Part No: [Number]  
 Rev: REV. 2

# A Look at the Super AMS Card

by Shawn Baron



Although most of this article is geared toward programming and the technical side of the AMS card, it does start out basic(read to see), and might just help you out a bit. Have fun with the AMS card.(If you have one of course.)

To start: When Richard Baron, my dad, and I, got an AMS card, the first thing I did was try to plug it in. I had already removed my 32K card, and then I tried to put it in. The first "problem" I had was which side is which. I wasn't sure how to plug it in. My dad came and somehow got it in there. The card is in there pretty tight, but it does work 100% fine. I put the side with the 4x2 grid of small chips(the side opposite the round blue/black thing), toward the front of the PE Box. This put the "top" side (the side with all of the chips) toward the disk drives. Now it was installed. To test this I went into Basic(right after plugging it in), and did the size command. It came up perfect.

NOTE: For those of you wishing to use the latest version of XBPacker(v?.?), it has come to my attention that you must FIRST load ABOOT, then load XBPacker. Thanks to Jack for that one.

The next day, I tried writing a program to display what pages are dumped into memory where. I had many problems. I could get paging to work out of Mini Memory by doing it manually, but I needed a program to do it. I finally figured it out.

Table 1 AMS Card Setup CRU Base >1E00

--Label--	Memory Address--	PagePos--	"DUMB" Page #--
--M2	>2000->2FFF	>4004	>02
--M3	>3000->3FFF	>4006	>03
--MA	>A000->AFFF	>4014	>0A
--MB	>B000->BFFF	>4016	>0B
--MC	>C000->CFFF	>4018	>0C
--MD	>D000->DFFF	>401A	>0D
--ME	>E000->EFFF	>401C	>0E
--MF	>F000->FFFF	>401E	>0F

The program below shows a basic Assembly program to use the AMS card:

```

M2 EQU >4004 *PagePos
M3 EQU >4006 */from Table 1
*--Setup AMS Card--*
LI R12,>1E00 *CRU Base of AMS Card
SBO 0 *Turn AMS on.(This IS always done)
*Now setup "DUMB" mode for memory mapped.
LI R0,>4000 *Start of memory(See Table 1)
CLR R1 *Start at page >00.
LP1 MOV R1,*R0+ *Move 2 bytes(page in each)to
* MemMapper, and INCT
AI R1,>0101 *Add 1(page)to both bytes.
CI R0,>4020 *All Done?
JLT LP1 *No, init more.
*Now turn memory mapper on.
SBO 1 *Turn MM(Memory Mapper)on.
*--Now we Page--*The AMS is now setup to swap pages.
LI R0,>2020 *Load page >20(32)into both bytes.
MOV R0,QM2 *Page #32 into Memory >2000->2FFF
LI R0,>2121 *Load page >21(33)into both bytes.
MOV R0,QM3 *Page #33 into Memory >3000->3FFF

```

NOTE: Paging is not allowed into other memory areas, although there is area in the >4000 area of MM to allow it. As far as I can tell, it just ignores it.

If you have questions about the AMS card, feel free to write to SW99ers, we will do our best to help you.

Please remember, we are also learning how to program the Super AMS. We will publish anything we can find out about it. Please let us know what you find out!

Thanks to Jack for asking me the right questions to make me think.

## BBS Liability

by Tom Wills



I had a number of comments about the disclaimer I have about "private" messages not really being private on Cactus Patch. Most comments, maybe even all of them, are negative. "Why do I have to read it each and every time I log in?" I've been asked time and time again. Partially to let everyone know I can, and WILL read private mail. Why? To make sure no one is harrasing another user. And to make certain everyone lives up to the standards required of Cactus Patch callers.

Okay, so why have such strict standards for callers. First of all, we are a family oriented user group, and the Cactus Patch is the official BBS of the SouthWest Ninety Niners. Secondly, and more importantly, it's the way I want it.

Recently David Ormand emailed me the following article he got from the Internet. Basically it says I can be held liable for the stuff on the Cactus Patch, as every BBS Sysop can be. No way I plan on getting into deep trouble with the justice system. So read the following and see if you don't agree.

### JUDGE RULES PRODIGY LIABLE FOR LIBELLOUS CONTENT

(June 5th 1995) In a landmark judgement that will make life extremely hard for operators of on-line services, Justice Stuart Ayn of State Supreme Court in Mincola, New York last week ruled that Prodigy Services Co was a publisher of information rather than a simple distributor, and could therefore be sued for libel. The IBM Corp-Sears Roebuck & Co joint venture now faces a \$200m libel lawsuit. The decision came in response to a motion filed by Stratton Oakmont Inc, a Long Island securities investment firm that sued Prodigy in January for libel and defamation in response to a series of messages posted to Prodigy's Money Talk electronic bulletin board in October by an unknown Prodigy user. The messages tagged Stratton Oakmont, which specializes in penny-stock issues, and its president, Daniel Porush, as criminals and characterised the company's dealings as fraud. Stratton Oakmont has been investigated by the Securities & Exchange Commission for reported irregularities and securities violations, the New York Times notes.

Judge Ayn drew a distinction between Prodigy - which uses human monitors and automated systems to remove objectionable material posted to its network - with other information services that merely act as conduits for unfiltered information, much as a book store or a library distributes information but does not publish it. Prodigy says it was pre-screening messages more than a year before the material objected to went up in October. The judge also held that Charles Epstein of Chicago, who managed the Money Talk forum under contract to Prodigy, was acting as an agent of Prodigy. Stratton Oakmont contends that Prodigy is responsible for Epstein's failure to remove the postings promptly.

(C) Computergram International

## 512Kx8 bit Low Power full CMOS Static RAM

### FEATURES

- Process Technology: Full CMOS
- Organization: 512Kx8
- Power Supply Voltage: 4.5~5.5V
- Low Data Retention Voltage: 2V(Min)
- Three state output and TTL compatible
- Package Type: 32-DIP-600, 32-SOP-525, 32-TSOP2-400F/R

### GENERAL DESCRIPTION

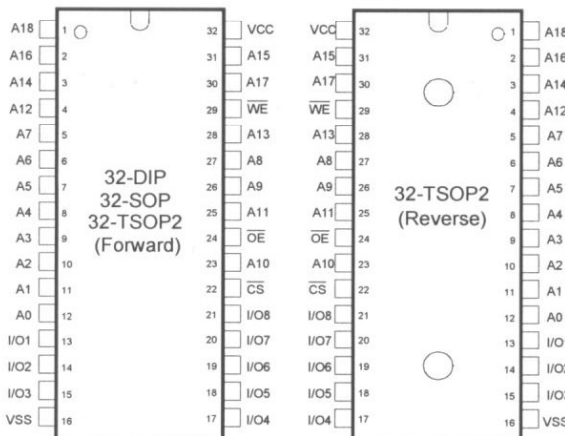
The K6X4008C1F families are fabricated by SAMSUNG's advanced full CMOS process technology. The families supports various operating temperature range and various package types for user flexibility of system design. The families also support low data retention voltage for battery back-up operation with low data retention current.

### PRODUCT FAMILY

Product Family	Operating Temperature	Vcc Range	Speed	Power Dissipation		PKG Type
				Standby (I <sub>SB1</sub> , Max)	Operating (I <sub>CC2</sub> , Max)	
K6X4008C1F-B	Commercial (0~70°C)	4.5~5.5V	55 <sup>1)</sup> /70ns	20μA	30mA	32-DIP-600, 32-SOP-525, 32-TSOP2-400F/R
K6X4008C1F-F	Industrial (-40~85°C)			30μA		
K6X4008C1F-Q	Automotive (-40~125°C)					32-SOP-525, 32-TSOP2-400F

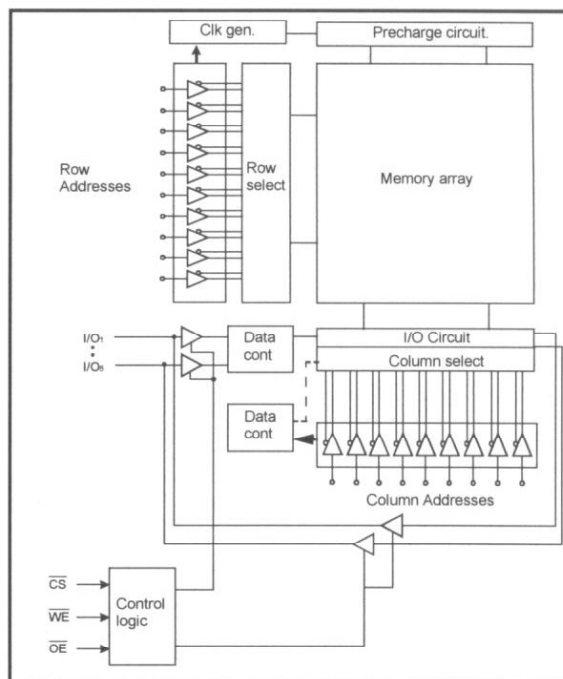
1. The parameter is measured with 50pF test load.

### PIN DESCRIPTION



Pin Name	Function
$\overline{WE}$	Write Enable Input
$\overline{CS}$	Chip Select Input
$\overline{OE}$	Output Enable Input
A0~A18	Address Inputs
I/O1~I/O8	Data Inputs/Outputs
Vcc	Power
Vss	Ground

### FUNCTIONAL BLOCK DIAGRAM



SAMSUNG ELECTRONICS CO., LTD. reserves the right to change products and specifications without notice.