

Replacing SRAM with a SIMM

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2006-09/msg00427.html>

- *From:* "FyberOptic" <fyberoptic@xxxxxxxxx>
 - *Date:* 2 Sep 2006 11:14:38 -0700
-

Hiya folks. I've got a 6502-based project going which uses sram for system memory and an eeprom for running code. A little more than half of the memory map is available for eeprom use. I have a bank-switching method implemented to take advantage of up to 128k of data from the eeprom, as well.

My eventual goal is to replace most of that eeprom space with more ram, and make the eeprom only store a basic OS, which would be able to load software from a drive or something of that sort into the extra memory, instead of having to reprogram the eeprom every time I want to run something I write.

The thing is though, roughly 32k of system memory obviously isn't very much space to work with. I'd like to be able to implement the same style of bank-switching into this extra system ram, adding maybe up to 512k-1mb of switchable storage capability, to avoid much disk activity. But the problem is sram isn't cheap, especially when you start getting into such "large" capacities.

So my thought was to take one of these countless 30-pin simms I still have laying around and see if it might be possible to use one of those instead. I know that they're dram, which requires refreshing, but I'm curious what this would entail, such as what extra hardware I might need to take care of it, and if it would impact on the normal functionality of the system to allow for the refreshing compared to the transparent workings of sram.

I've done a little bit of looking around, but I still don't know a lot about working with dram in general. Going by a pinout, I don't suppose I really need the parity bit and such, unless you're required to use it. Some simms don't even use parity from what I understand.

Anyhow, this seems like it might be the cost-effective solution, if it can be implemented without affecting how it already works very much. I'd certainly appreciate any input anyone has to offer.

.