

memCellsF09 Description

Erik Brunvand, Fall 2009

The memCellsF09 library contains fabrication-tested SRAM cells and associated circuitry that can be used to assemble single-port and dual-port SRAMs in the AMI C5N (SCMOS) process. To access the library, add /uusoc/facility/cad_common/local/Cadence/lib/memCellsF09 to your library path. Make sure to click the “show categories” button in your library manager. There are a lot of cells in the memCellsF09 library and they are all grouped into categories. Without seeing these categories it will be hard to find the cells you’re looking for. A view of the library manager showing the memCellsF09 library with the categories enabled is shown in Figure 1.

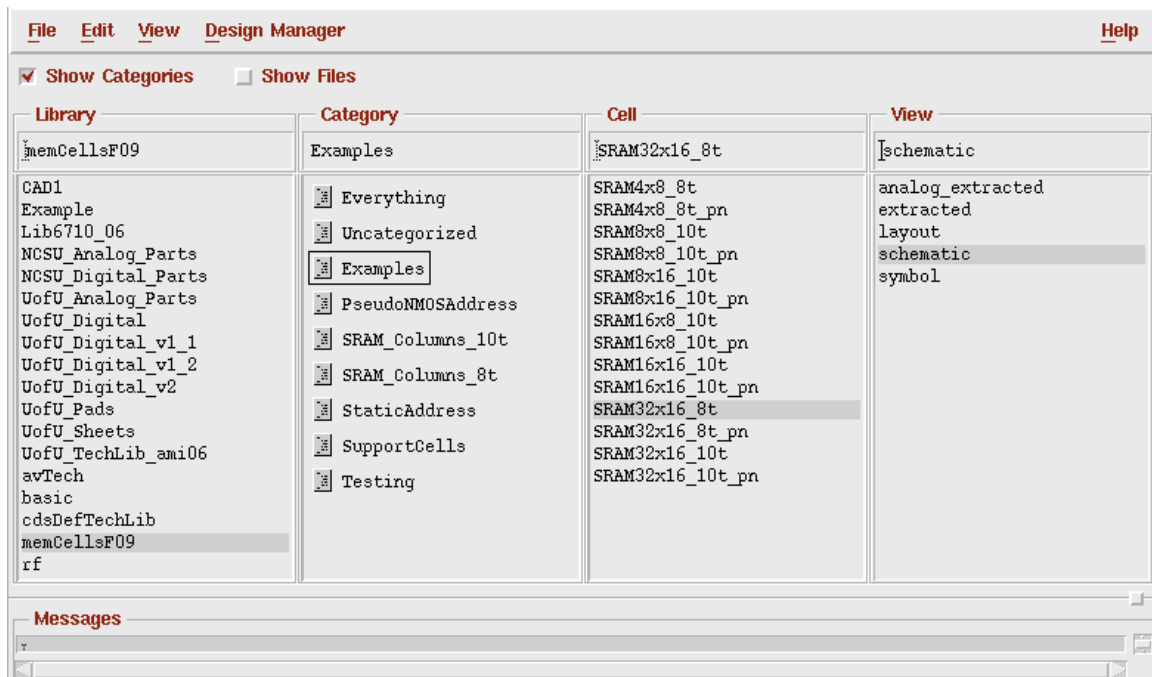


Figure 1: memCellsF09 library (with Categories enabled)

The categories are as follows:

- Everything: This is a category you can use to see every cell in the library.
- Uncategorized: This has cells that are not part of any other category
- Examples: This has example cells that I’ve assembled to show you how the circuits are used to make a complete SRAM. You may use these directly if you like (and if they’re the right size).

- PseudoNMOSAddress: This has address decoders that use a pseudo-NMOS style circuit to make the address decoders. These circuits make a slightly faster address decoder than the static version, but burn a little more power.
- SRAM_Columns_8t: 8-transistor single-ported SRAM columns that include the reading and writing circuitry at the bottom of the column
- SRAM_Columns_10t: 10-transistor dual-ported SRAM columns that include reading and writing circuitry at the bottom of the column
- StaticAddress: Address decoders built from static nand/nor circuits. These are slightly slower than the Pseuo-NMOS circuits, but burn less power.
- SupportCells: This category hides all the small cells that the columns and address decoders use.
- Testing: Examples of config views of some of the Example SRAMS to show how they can be simulated with mixed-mode (spectreVerilog) simulation.

Single-Ported SRAM

The basic single-port SRAM cell is an 8-transistor cell with differential bit/bitbar wires for writing, and a pull-down transistor connected to a separate single-ended read wire. The schematic is seen in Figure 2.

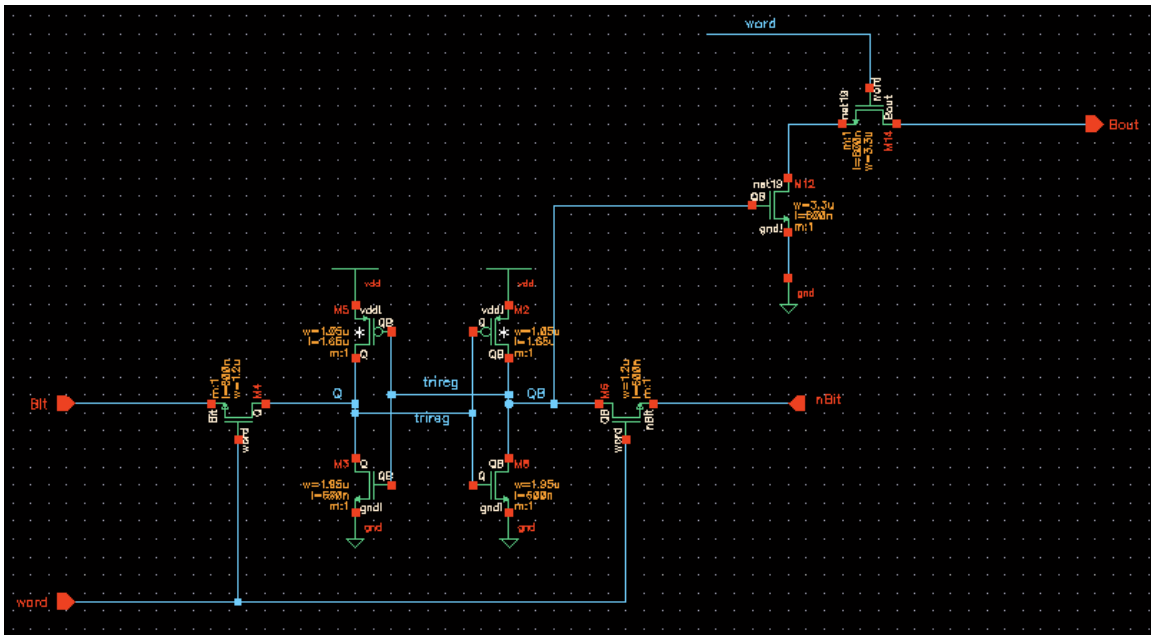


Figure 2: 8-transistor SRAM schematic

Dual-Ported SRAM

The memCellsF09 library also has a 10-transistor SRAM bit that has two ports: one for both reading and writing (depending on the state of WE), and one separate read-only port. The cell has the same cross-coupled storage as the 8-transistor cell, but

adds one more pulldown stack for the separate read interface, and a second Word line for that second read interface. The schematic is shown in Figure 3.

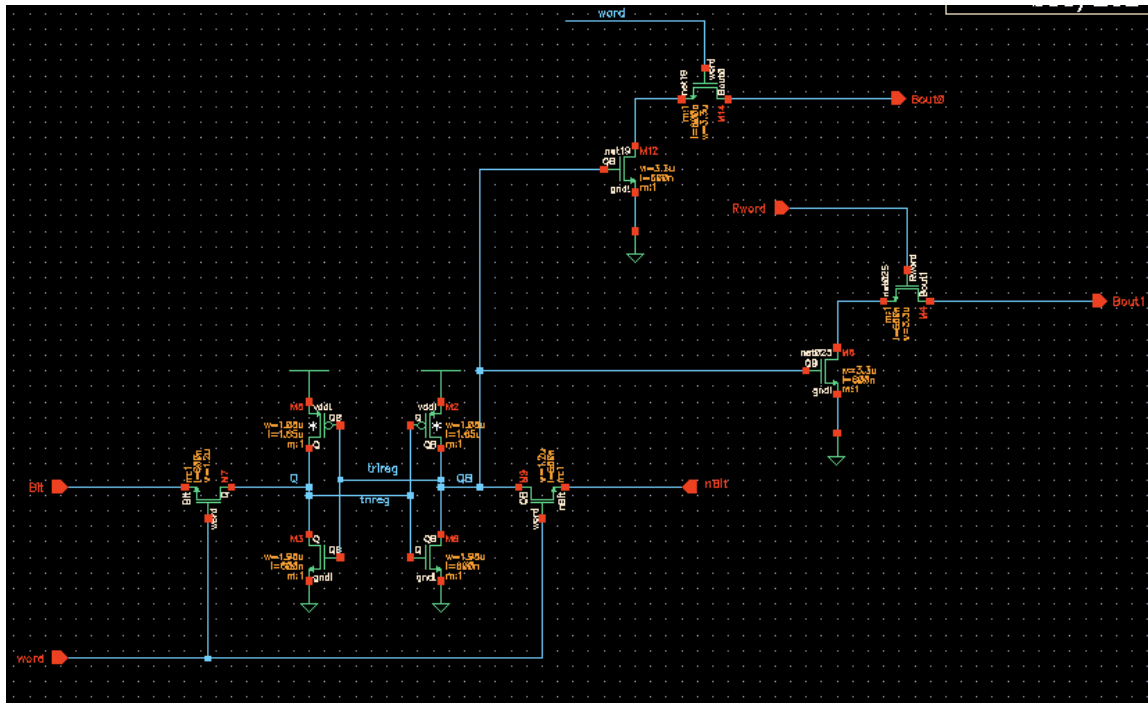


Figure 3: 10-transistor SRAM Schematic

The 10-t dual-port cell is used with two address decoders. One decoder (on the left side of the RAM array in the layout) is both a read and write port. It always reads from that address (to the Dout0 output), and also writes if the WE signal is asserted (high). The second address decoder is located on the right side of the memory array in the layout, and is a read-only port (to the Dout1 port). Reads are asynchronous: as soon as the address changes, the new data is read (after some read delay, of course). Writing is also asynchronous in the sense that there is no clock signal. Writes are, however, synchronized to the WE signal. The write data and address should be set up first. Then the WE signal should be asserted. Shortly after the WE is asserted, the data is written to the memory array.

SRAM Columns

The SRAM cells are assembled into columns that include the SRAM bits and the read/write circuits for the column. The number of bits in the column is the number of words in the memory. The column shares bit and bitbar lines for writing into the SRAM, and the Dout wire for reading the SRAM (Dout0 and Dout1 for the dual-ported SRAM). Each row of the column has a Word line that is activated both for reading and writing. At the bottom of the column are large drivers for writing, and a pullup and inverter for reading the value on the Dout line. The write drivers are only turned on when the WE signal is activated. If the WE is not activated, the drivers are

not turned on and the mild pullups on bit and bitbar are not strong enough to flip the bit. The value stored in the SRAM bit is used to either pull down the Dout wire or not.

Columns in the memCellsF09 library are assembled in pairs where each row of the column has two bits in the row (with a shared Word line). These 2-bit columns may be tiled in an array to make SRAMs of any reasonable word width. Note that this means that SRAMs that use these columns must have an even number of bits in a word.

The pre-assembled SRAM columns come in heights (number of rows, or number of words in the memory) of 4, 8, 16, 32, and 64. These are the only row sizes supported by the address decoder cells.

Address Decoders

Each port in an SRAM needs an address decoder. This converts binary addresses to unary outputs that drive the word lines in the SRAM. Address decoders come in two types: static (built from nand/nor circuits) and pseudo-NMOS (built from a single pmos with the gate tied low acting as a pullup resistor and a pulldown for each bit of the address decoder). The static address decoders are a little slower than the Pseudo-NMOS (mostly because of undersized word-line drivers), but burn less power than the Pseudo-NMOS decoders.

Address decoders (in each style) come in 4, 8, 16, 32, and 64 row sizes (2, 3, 4, 5, and 6 address bits). These address decoders limit the number of rows each SRAM can have.

The address decoders starting with “s_” are static, and with “pn_” are Pseudo_NMOS. The versions with no extra annotations (e.g., s_addr16 or pn_addr8) are designed for the single-port SRAMs. The versions with “10t_l” are designed for the left side of a dual-port SRAM, and with “10t_r” are for the right side of a dual-port SRAM. Note that for the Pseudo-NMOS circuits, the “plain” decoder works for the left hand side of both single- and dual-port SRAMs.

Assembling SRAMs

Using the cells in the memCellsF09 library you can assemble complete single- and dual-ported SRAMs. The number of words can be 4, 8, 16, 32, or 64, and the number of bits in the word can be any even number (up to the width of the chip you’re using the SRAM in). The complete SRAM consists of an address decoder (or two address decoders for the dual-port SRAM) and an array of SRAM columns. The RAMs are assembled as follows:

- Choose an address decoder style. The static address decoders are a little slower, but consume less energy. If you have word widths of 32 or less, you can choose either style. For longer word widths, the larger row drivers in the Pseudo-NMOS address decoders might work better. But, there are no hard

and fast rules. As always, analog simulation is best way to see if the circuit is working the way you'd like it to.

- Choose single- or dual-port SRAM. The pre-assembled single-port columns are located in the SRAM_Columns_8t category, and the dual-port columns are in the SRAM_Columns_10t category. The columns are two-bits wide, and either 4, 8, 16, 32, or 64 bits tall. You can make an array of these columns to make your word any even number (columns each have two bits). When you instantiate the columns, choose the number of columns in the array to match the number of bits in your word. For example, if you want a 32bit word, make an array with 16 columns.
- Assemble the layout for a single-port SRAM with an address decoder on the left, and the SRAM column array on the right. For the dual-port SRAM you need an address decoder on both the right and the left with the SRAM column array in the middle.
 - For the static address decoders, use the `_l` version on the left, and the `_r` version on the right. For the right hand decoder, instantiate the decoder flipped around the vertical axis so that the outputs are on the left of decoder cell.
 - If you're using the Pseudo-NMOS decoders, the left decoder is the same for both SRAM types, and the `_r` version is used on the right (again flipped on the vertical axis). You can look at the cells in the Example category to see how the cells should be overlapped.
- Add shape-pins on the address inputs, the data inputs, and the data outputs. For the static address decoder cells, the address inputs are in the middle of the cells. For the Pseudo-NMOS decoders, the address inputs are at the bottom of each decoder cell.
- Make a schematic that includes cells for the address decoder(s) and the SRAM columns. Again, you can use the Example SRAMs for examples of how to assemble the schematic.
- DRC the layout, then extract and LVS against the schematic.
- You should be able to simulate the schematic with Verilog (Verilog-XL or NC_Verilog), and you can also simulate with Spectre or spectreVerilog for more timing details. There are examples of mixed analog/digital config cells in the Testing category.

The best way to use these cells is to look at the examples in the Examples category. You can copy these cells and then modify them to be whatever word-width and number of words that you like. Once you copy the cell, you can use the "q" properties to modify, for example, the number of columns in the memory array, or the type of address decoder. Remember to change the layout and the schematic, and to re-run DRC, Extract, an LVS. I find that if you're making a new cell by modifying an old one, it's best to delete the symbol and then generate a new symbol once you get things modified the way you want them.

Some examples of assembled SRAMs are seen in the following Figures.

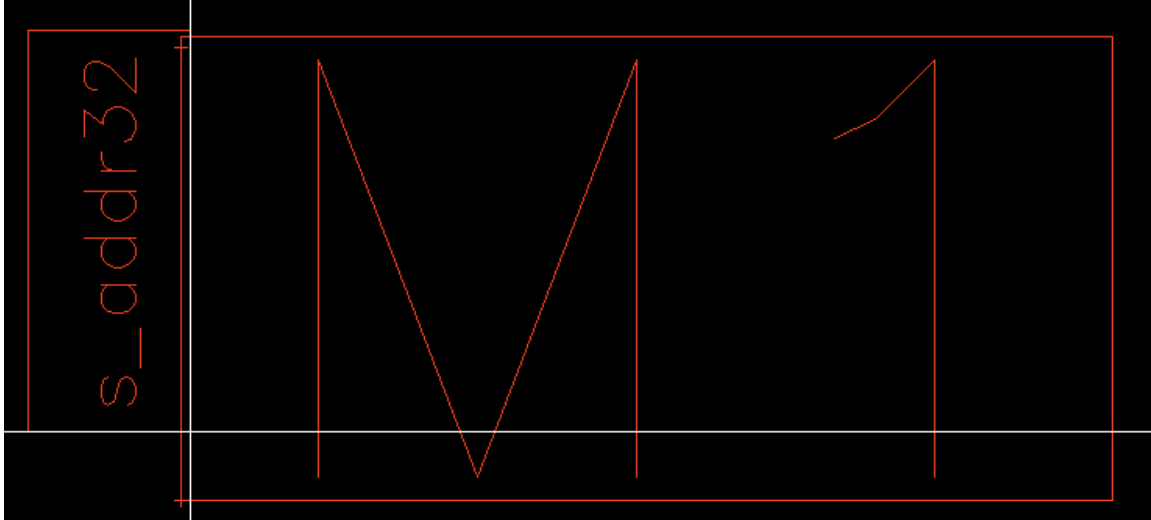


Figure 4: 32x16 (32 words 16 bits per word) single-ported SRAM showing a static address decoder on the left (s_addr32) and an 8t SRAM array (M1) in the center

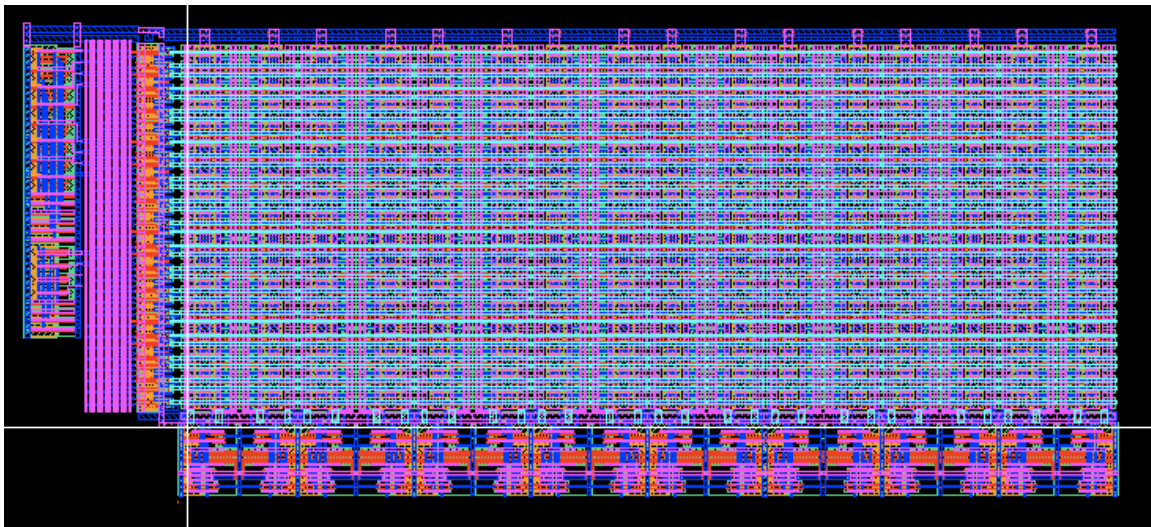


Figure 5: 32x16 single-ported SRAM expanded to see the layout



Figure 6: Schematic of 32x16 single-port SRAM. The interface is in the upper left. The address decoder (s_addr32 in this case) is below that. The SRAM columns are in the lower right. Note that because each column is two bits wide, there are 8 columns here that make up the 16 bits in each word.

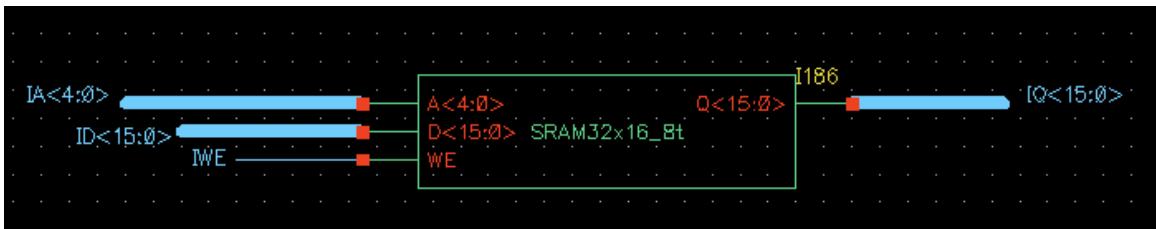


Figure 7: The symbol for the 32x16 single-port SRAM



Figure 8: A 32x16 (32 word, 16 bits per word) dual-ported SRAM. Note that there are address decoders on both sides of the memory array M2.

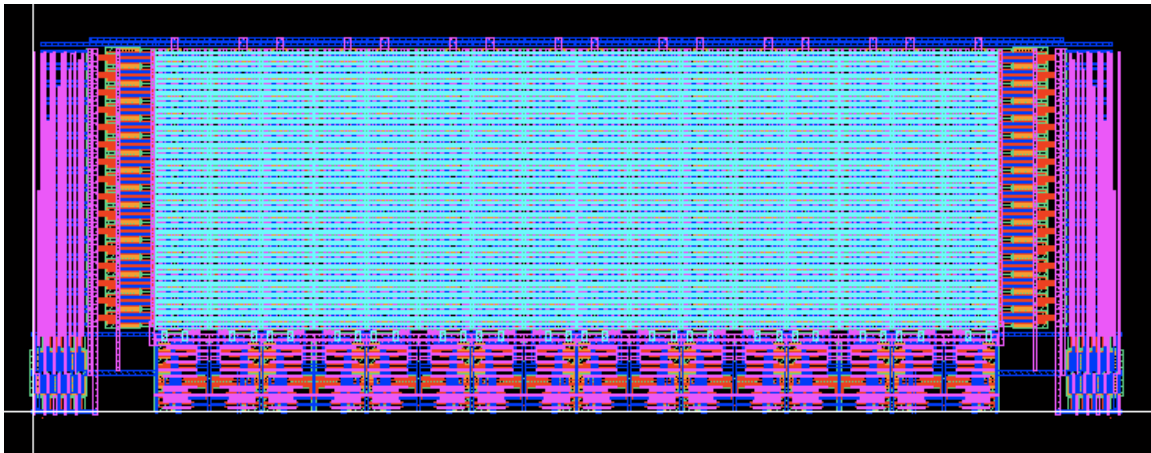


Figure 9: An expanded view of the dual-ported 32x16 SRAM showing the layout

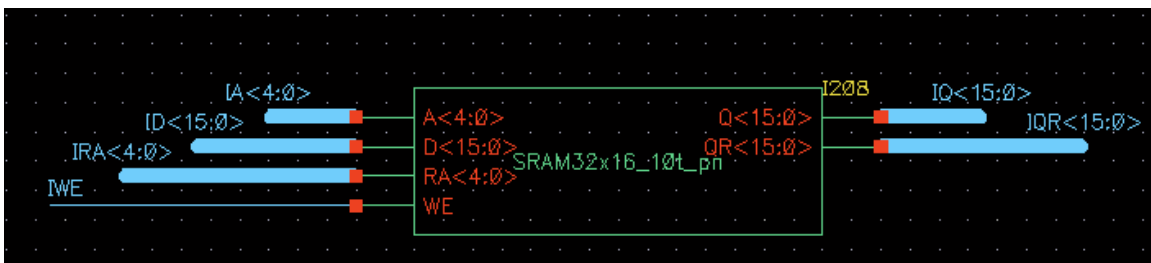


Figure 10: Symbol for the dual-ported 32x16 SRAM. Note that address A is the r/w address and results in data appearing on the Q output. The RA is the Read-Address for the second, read-only port. Its data shows up on port QR.