TOY-SRAM

Robert Montoye

June 2021

Microprocessors use multiple high-speed multiport register files to improve their performance. In BOOM, the high-performance RISC V, the custom register file took as much design effort as the rest of the design. The TOY-SRAM, open-source, multi-port memory system is designed to replace custom circuit design with a simple set of choices for the fab and the system designer. For the chip fabrication facility, it offers a canonical specification that can be optimized for use in a wide variety of applications. The system designer can then select the desired fast, low voltage friendly multiport memory from a menu of choices.

<u>System components:</u>

1) **The custom core consists of a 10T SRAM dense library cell and its associated, well-tuned, hand-placed, standard cell decoders, and I/O circuitry in a hard macro.** The array is 64 registers of 24 bits so that 3 copies produce 72-bit SEC-DED error correction for 64 bits. Its size is approximately 360 PC x 450 M3 tracks. This array + wrapper is 2/3 of the area of *many* standard 2R1W register files, because it uses the 10T SRAM cell, optimized decoders, and I/O circuitry, which are carefully placed and separated from the "wrapper."

   Bit cell:

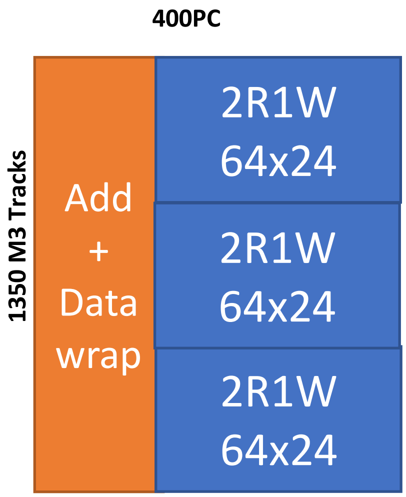
   The bit-cell,] starts with a 6T SRAM, which has a row of 2 PFETS with 2 gaps, and a row of 4 NFETs with no gaps. It expands to 10T cell with 2 Reads and 1 Write. This requires 3 word lines (M2) and 4 bit lines (M1)

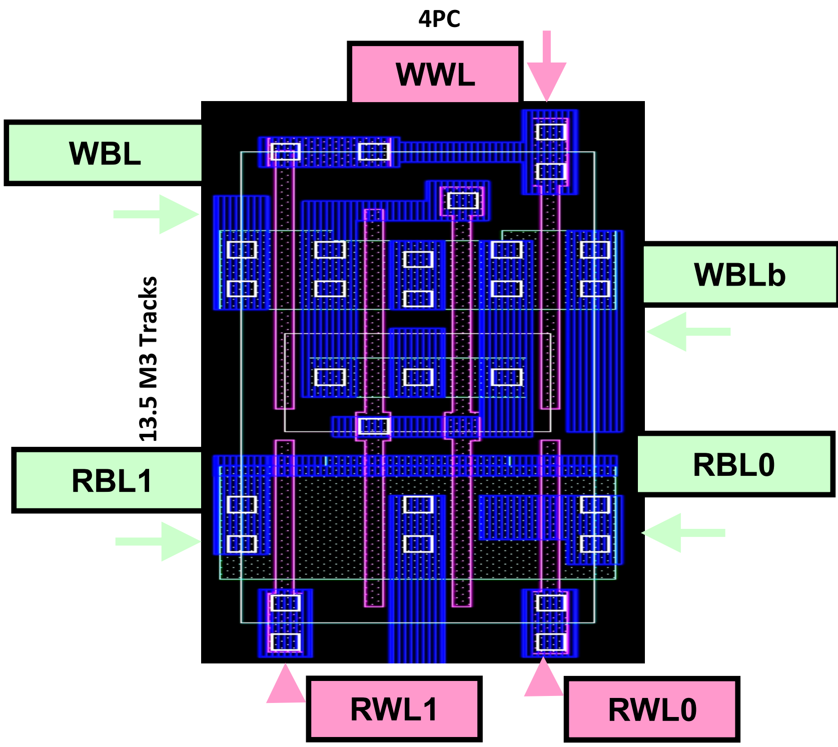
Separation of the Read and Write ports, eliminates "Read disturb" by reading off a high impedance path. This allows a < 10 FO4 Read and Write sub-cycle. It also allows TOY-SRAM more digital and less "mixed signal" and supports much lower VDD operation.

   Modular peripherals:

   TOY-SRAM exploits similarities of cell libraries to produce efficient periphery at the custom core closest to the 10T SRAM cell. Aligned with the 10T SRAM cell is a decoder composed of hand-placed 4PC wide NAND2 and inverter standard cells allowing > 50% bit-cell occupancy with a 3B subarray, further speeding up the basic array.

2) **A CAD-friendly "wrapper" with latches, clocking system, and output muxes supporting SDR and DDR clocking per port.** When used correctly the SDR or early read is < 10 FO4 (CLK to Q) and the late read is < 20F O4. The wrapper enables sharing many custom cores and can multiplex BIST input/output and column redundancy if necessary





SRAM Clocking:

TOY-SRAM allows either single or double data rate for the Read or Write, ranging from 2R1W, to 2R2W, to 4R2W, depending on which port is run at double data rate. Reading twice a cycle costs latency for the second access, but its modest cost justifies a late read for many users. DDR Read and Write require a clock generator that uses a single edge to form all the necessary clocks/multiplexing to launch two < 10 FO4 events in a cycle.

Multiplexing for DDR Read and Write:

1) DDR Write multiplexes the address/enable as well as the input data. The alignment of the input data and the WWL pulse requires care to fit two < 10 FO4 wave-pipelined stages of: decode-WWL- Cell Write

2) DDR Read multiplexes only the address/enable and requires an extra XNAND latch to hold the value of the 2nd read, creating two < 10 FO4 wave-pipelined stages of: decode-RWL-LBL-GBL-Latch.

3) **An algorithm to expand read or write ports without redesigning the hard macro.**
At ISSCC 2011, an early version of SRAM used bank duplication to produce 4 Read ports with 2 banks of 2R2W custom cores by writing duplicate data to both banks and reading from each bank. Similarly, one can write separate data to copies, maintain a MostRecentlyWritten per-register bit, read both copies, and select the most recently written one to increase the number of Write ports. This allows growth both Read and Write ports from single custom core, using additional copies in addition to faster sub-clocking.

<u>Summary</u>

Building TOY-SRAM in the Skywader 130nm fab will test the efficacy of this modular portable efficient system, and potentially ease migration to smaller nodes down to 2nm. silicon by suggesting a simpler metallization that scales from Skywater 130nm., simplifying the first design that is more than a transistor parameterization site TOY-SRAM will make the most voltage scalable and highest bandwidth per unit area multiport memory system a standard and available to help improve competitiveness in silicon design and implementation.

The Case for Making TOY-SRAM Open Source

Accomplish three key goals by making TOY-SRAM available as the open-source:

1) Having access to an open-source TOY-SRAM will ease multiport memory development for processors and accelerators, eliminating much of the time-consuming circuit/layout of multiport SRAMs, enabling IBM to focus more on product value.
2) An open-source, *multi-port voltage scalable* SRAM would standardize the bit cell component, spurring competition among vendors for the performance critical 10T SRAM, which has been ignored while focusing on higher density 6T SRAM.
3) Having an open-source, multi-port SRAM would further advance open source hardware at the lowest design level, closest-to-the-fab.