Title

SRAM Design with OpenRAM in SkyWater 130nm

Authors

Cirimelli-Low, Jesse
Khan, Muhammad Hadir
Crow, Samuel
et al.

Publication Date

2023-05-25

DOI

10.1109/iscas46773.2023.10181379

Peer reviewed

SRAM Design with OpenRAM in SkyWater 130nm

Jesse Cirimelli-Low*, Muhammad Hadir Khan*, Samuel Crow*,
Amogh Lonkar*, Bugra Onal*, Andrew D. Zonenberg†, Matthew R. Guthaus*
*Computer Science and Engineering, University of California Santa Cruz, Santa Cruz, CA 95064
†IOActive, Seattle, WA 98119
{jcirimel, mkhan33, sacrow, alonkar, bonal, mrg}@ucsc.edu andrew.zonenberg@ioactive.com

***Abstract*—OpenRAM is an open-source framework for the development of memories with an initial focus on SRAMs. OpenRAM provides an application interface for netlist, layout, and characterization to create designs using either open-source or commercial verification and simulation tools. The first silicon in SkyWater 130nm has been successfully verified which includes a 32-bit 1-kilobyte dual-port SRAM macro. This paper presents the first macro design, test setup, test results, and enhancements on a subsequent tape-out.**

I. Introduction

OpenRAM is an open-source memory compilation framework for on-chip memories [1]–[5]. It is currently focused on Static Random-Access Memories (SRAMs), but extensions are under way for other types of memories (register files, non-volatile memories, etc.). While OpenRAM has existed for quite a few years using non-fabricable processes (FreePDK45 [6], older MOSIS processes (SCMOS), etc.), we partnered with Google and SkyWater to release OpenRAM for their 130nm open-source Process Design Kit (PDK) [7]. To date, OpenRAM has been used on the initial Strive SoC [8], on all seven Multi-Project Wafers (MPWs) as part of the Caravel management core [9], and on many of the user contributed projects.

This paper presents the first OpenRAM test design in SkyWater 130nm, called OpenRAM Test-Chip One (OR1), as well as its successful bring-up and measurement. In addition, we discuss improvements that were incorporated on a second OpenRAM test chip (MPW2) with silicon delivery happening very soon. The rest of this paper is organized as follows: Section II provides an overview of the SkyWater 130nm process, Section III explains the implementation of the OpenRAM SkyWater 130nm SRAM, Section IV discusses the test-chip tape outs, Section V explains our bring-up and measurement setup, Section VI presents the measurement results, and Section VII concludes the paper.

II. Overview

SkyWater 130nm is the first open-source commercial process supported by OpenRAM. SkyWater 130nm is well into Deep Sub-Micron (DSM) scaling and introduces multiple layers with lithographic challenges. While technically a 150nm drawn length process, the effective channel length is 130nm. It has a nominal 1.8V supply voltage and includes a nominal and low V_{th} and NMOS and PMOS devices as well as a high-V_{th} PMOS. The process has a Titanium Nitride (TiN) Local Interconnect (LI) layer specifically for memories to improve density as well as a deep nwell for substrate isolation. In addition, multiple layers have Optical Proximity Correction (OPC) to improve lithography. This is especially used in the SRAM primitives.

SkyWater is the first technology in OpenRAM to have foundry-provided bit cells rather than ones created using user design rules. The foundry bitcells use the high-V_{th} PMOS devices for low leakage. These bitcells use "core memory" rules which have pre-made, rule-based OPC and place certain restrictions on array layout. The OPC allows transistor sizes and area less than what would be achievable by user design rules. The respective bitcell layouts are shown in Fig. 1.

III. Implementation

The OpenRAM implementation proceeded in two phases for the OR1 and MPW2 tape-outs. On OR1, OpenRAM including the technology files and bitcells was entirely open-source, but it leveraged proprietary tools for DRC, LVS, and simulation because the complete open-source PDK was in development. The goal of OR1 was to achieve functionality in the new technology on first silicon. The goal of MPW2 was to transition entirely to open-source tools for DRC, LVS, and simulation. We also expanded from a single working macro to multiple configurations including both dual- and single-port memories.

A. Technology

OR1 faced several challenges: support for non-standard layers (LI, deep nwell, and nitride poly-cut), non-standard

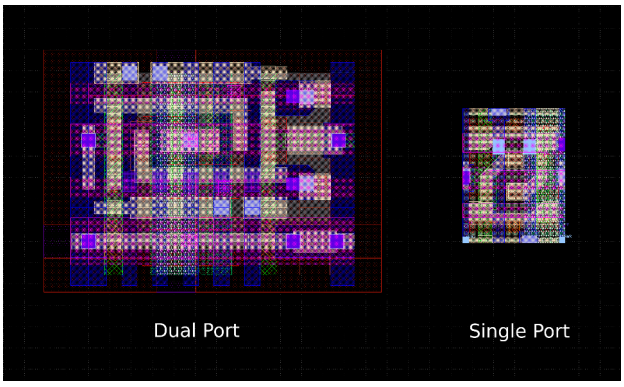

Fig. 1. 1RW1R bitcell (left, area 6.162μm^2) and 1RW bitcell (right, area 1.896μm^2)

usage of layer purposes, custom library cell names instead of fixed cell names, core memory design rules, and OPC. While we used proprietary tools for verification on OR1, MPW2 added support for abstract cell views to enable design rule waivers. OpenRAM's support for Sky130 also included support for the OpenLane flow as discussed in Section IV-C.

B. Bitcell and Bitcell Array

OpenRAM generates dual-port memories with any combination of read, write, and read-write port types. The OR1 test chip specifically uses a custom 8 transistor (8T) bitcell to implement a combined 1 Read-Write and 1 Read (1RW1R) structure. The 1RW1R bitcell is a combination of the primary 8T dual-port bitcell along with "tap" cells which bias the array's nwells, and "strap" cells which connect the polysilicon bitcell wordlines to the lower resistance M2 layer. The 1RW1R bitcell has OPC on both the polysilicon and diffusion layers.

On MPW2, we taped out various dual- as well as single-port memories. The single-port memories also use a custom foundry bitcell and have a variety of internal strap and tap cells depending on placement context. This more complex, regular pattern required tiling the cells to match a foundry-approved reference bitcell array. The single-port 6T bitcell has OPC on the polysilicon, diffusion, and LI layers. Both single- and dual-port memories are shown in Fig. 1.

OpenRAM uses a Replica BitLine (RBL) for self-timing [10]. We constructed two additional bitcells (replica and dummy) from the foundry bitcell while trying to avoid any changes to OPC. An RBL is used to time the sense enable signal of each read port. The RBL uses a column of replica bitcells which model the capacitance and leakage of a regular bitline to emulate the worst case timing for the SRAM. The replica bitcell always pulls down on the positive bitline. The bitcell array has a replica bitcell for every row as well as an extra replica bitcell in a dummy row below or above the array. The combination of the regular row and the dummy row ensure that the replica column has twice the pull down and is faster than a normal bitline [11]. In the extra dummy row, OpenRAM also places dummy bitcells which are identical to the regular bitcell except that the bitlines are not connected. The dummy rows add capacitance to the wordline capacitance for more accurate timing of replica wordline enable signal.

C. Address Decoder

OpenRAM memories use hierarchical address decoders for higher speed at the cost of additional area which is amortized over an entire bank. While OpenRAM can generate simple parameterized "standard cell" style gates that are used in control logic, these gates are not optimized for area or height. Decoder cells must be pitch-matched to the height of the bitcell and OpenRAM generated gates are often too large to be compatible with custom foundry bitcells. We create custom thin NAND gate cells which are used in conjunction with OpenRAM generated cells to implement hierarchical decoding in Sky130. In Sky130, the dual-port bitcell is $1.975\mu m$ tall and the single-port bitcell is $1.58\mu m$ tall, so our custom decoders are the same height. We created NAND2, NAND3, and NAND4 layouts for both single- and dual-port memories which match the pitch of the bitcells and are shown in Fig. 2. The decoder inverter is generated automatically as a parameterized gate by OpenRAM so that it may be sized appropriately for the array. Using our NAND4 gate, we supported hierarchical decoding with up to 4096 rows per bank.

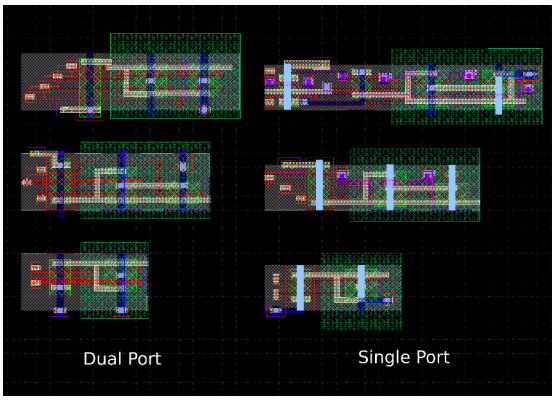

Fig. 2. Custom NAND gates for dual and single port banks. Top to bottom: NAND4, NAND3, NAND2

D. Data Port Logic

We utilized earlier SCMOS technology cells to quickly create custom sense amplifiers and write driver cells. These were able to match the width of the bitcells to enable a no column mux option. The precharge and column mux are automatically generated using parameterized transistors.

IV. Tapeouts

A. OpenRAM Test-Chip One (OR1)

OR1 included a single, byte-writable 32-bit 1-kilobyte dual-port SRAM with 1RW port and 1R port as shown in Fig. 3. The chip was IO-pad limited so a subset of data inputs and outputs were selected for verification purposes. Due to quick turn-around time and utilizing a simple test infrastructure, multiplexing of inputs and outputs was not explored on OR1. Specifically, OR1 bonded out control signals for both ports (clk0, csb0, web0, clk1, csb1), the byte mask for both ports (wmask0[3:0], wmask1[3:0]), and the eight bits for each port (addr0[7:0], addr1[7:0]). For the data pins, the most and least significant bit of every byte (bits 0, 7, 8, 15, 16, 23, 24, 31) for both data ports (din0, dout0, dout1) were bonded out. This enabled testing of the masking features as well as measuring timing and functionality of a range of bits. The chip core itself was manually routed. The power grid over the memory macro was manually extended to included a power ring and connected to the supply pins.

B. MPW2 Test Chip

The MPW2 test chip, shown in Fig. 4, has an assortment of both single- and dual-port memories in a range of sizes as shown in Table I. Additional test logic interfaces between the Caravel host through the Logic Analyzer (LA) parallel

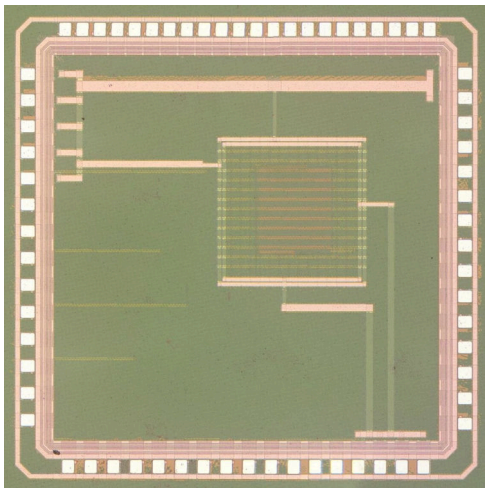

Fig. 3. The OpenRAM (OR1) test chip contains a single dual-port, 1-kbyte, 32-bit SRAM macro. 2.4mm x 2.4mm. Photo © 2021 John McMaster, CC-BY

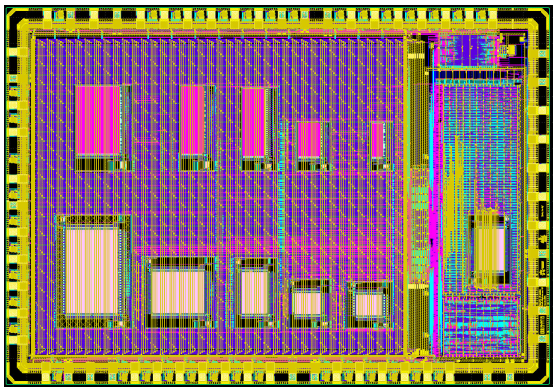

Fig. 4. The MPW2 test chip contains 5 dual- and 5 single-port SRAM macros along with the Caravel test harness. 5.2mm x 3.6mm.

interface or off-chip through a General Purpose I/O (GPIO) serial interface. Both of these use a 112-bit packet with fields for selecting and controlling operations on each memory as well as address and write data. During a read operation, the output data is read through the LA or GPIO interface.

C. OpenLANE Flow

We used OpenLANE [12] to design our test chips, although we had to make adjustments to the flow for our designs. We lowered the placement and routing density to reduce congestion. We increased the placement legalization distance to allow us to move cells across memories. Our test design used two possible sources for the clock: an off-chip scan clock through a GPIO pin or an on-chip clock through the LA interface. A custom design constraint file using the output of the clock select mux was used for static timing analysis.

We added several features to OpenRAM to support the OpenLANE flow. The Power Distribution Network (PDN) router required us to add metal rings around our memories for each supply. TritonRoute, the gridded detailed router, required us to create a signal escape router to put the control and data signals to the perimeter of each macro. We also generated an abstract LEF file which blocks all metal layers while allowing power and pin access to the macros.

We performed both behavioral and gate-level netlist simulations using both the LA and GPIO interface. The tests used C routines to read and write data from the Caravel host during the LA tests and a Verilog serial port for the GPIO tests. The tests automatically verified the results and also dumped the waveform view which could later be used to visualize the results.

V. Measurement and Test Setup

The OR1 verification platform consisted of two custom PCBs. The individual OR1 test chips were soldered to a carrier board containing bypass capacitors and a connector to the controller board. The controller board contained a STM32 microcontroller for top level test program control and a Xilinx Spartan-7 FPGA for test pattern generation and data capture.

The I/O pad ring for the OR1 test chip used standard LVCMOS33 signaling. The core power rail for the OR1 was controlled by a DAC and could be varied above or below the nominal 1.8V to test performance across voltage corners.

Output data waveforms and clocks were registered in FPGA output cells to minimize skew, and all traces from the FPGA to the test chip were delay matched. Read data was registered in FPGA input cells with a separate PLL phase, which could be adjusted with respect to the output clock in order to compensate for propagation delays and capture in the middle of the data eye. Very large (over $10ns$ in some cases) read capture delays were needed, due to excessive unbuffered routing delay between the OpenRAM macro's outputs and the I/O pad cells.

The top side of the QFN packaged test chip was attached to a two-stage Peltier heat pump with Arctic Silver 5 thermal interface material, allowing the device to be tested across thermal corners. Three conditions were tested, aiming at the upper, lower, and nominal points of typical commercial temperature range components. Temperatures were measured at the surface of the Peltier plate touching the chip (FLIR Lepton 3 thermal camera) and the back side of the PCB opposite the chip (PT100 RTD). No on-die temperature sensor was available so actual T_j could not be measured, however given the low thermal mass/resistance of the QFN package and negligible self-heating we expect it is likely close to the Peltier plate temperature.

- Cold (target T_j 0C): +2.6A through Peltier, -0.2C on plate surface, +12C on bottom of PCB
- Ambient (target T_j +23C): heat pump inactive, +23.9C air temperature
- Hot (target T_j +85C): -2.7A through Peltier, +86.6C on plate surface, +55.7C on bottom of PCB

VI. Results

The OR1 memory has been tested and is operational. We have run tests to measure operating frequency, voltage, and retention. In each case, the initial conditions are fully operational and exhibit no errors. We ran experiments to quantify bit

TABLE I
MEMORY CONFIGURATIONS TAPED-OUT ON THE MPW2 TEST CHIP

Macro name	Instance name	Size (bytes)	Ports	Data word (bits)	Depth	Width (μm)	Height (μm)
sky130_sram_1kbyte_1rw1r_8x1024_8	SRAM0	1024	1RW1R	8	1024	455.3	446.5
sky130_sram_1kbyte_1rw1r_32x256_8	SRAM1	1024	1RW1R	32	256	479.8	397.5
sky130_sram_2kbyte_1rw1r_32x512_8	SRAM2	2048	1RW1R	32	512	683.1	416.5
sky130_sram_4kbyte_1rw1r_32x1024_8	SRAM3	4096	1RW1R	32	1024	693.9	668.8
sky130_sram_8kbyte_1rw1r_32x2048_8	SRAM4	8192	1RW1R	32	2048	1093.8	720.5
sky130_sram_1kbyte_1rw_32x256_8	SRAM8	1024	1RW	32	256	478.4	223.4
sky130_sram_2kbyte_1rw_32x512_8	SRAM9	2048	1RW	32	512	481.1	322.7
sky130_sram_4kbyte_1rw_32x1024_8	SRAM10	4096	1RW	32	1024	806.9	351.2
sky130_sram_4kbyte_1rw_64x512_8	SRAM11	4096	1RW	64	512	828.6	339.0
sky130_sram_8kbyte_1rw_64x1024_8	SRAM12	8192	1RW	64	1024	830.7	541.7

error rates in each case using the bonded-out bits as discussed in Section IV-A and include some of the analysis here.

To optimize read timing, we swept capture delay from 6.6 to $11ns$ at frequencies from 10 to $45MHz$ on five sample dies. A representative result is in Fig. 5. At each point we tested single and dual port operation across the three thermal corners for a total of six tests.

We then repeated the same frequency sweep across the test dies while sweeping voltage from 1.4 to $1.8V$. There were no errors above $1.7V$ at frequencies below $34MHz$ with both ports simultaneously reading the same addresses. With only a single read port active (less loading on bit cells) and the clock reduced to $25MHz$, voltage could be reduced to $1.54V$. A representative result is shown in Fig. 6.

We performed a minimum retention voltage test by writing all bits with random data at $1.8V$, dipping to the test voltage for $10s$, raising the voltage back to $1.8V$, and reading the results back after allowing $10ms$ for power to stabilize. The first errors were seen at $440mV$ when cold and $410mV$ when hot.

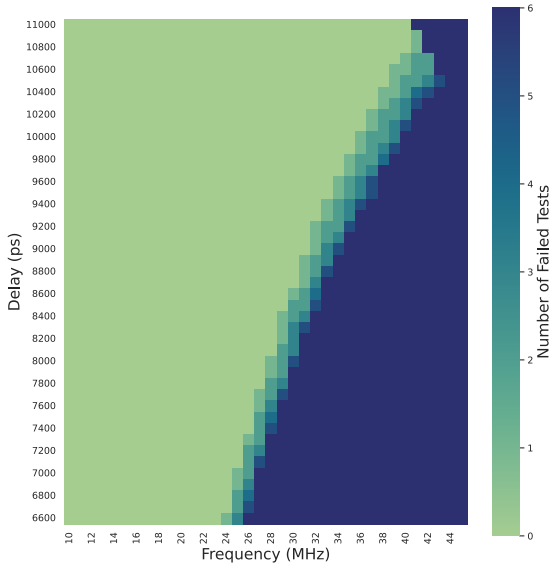

Fig. 5. Frequency and delay analysis for representative die 4 at 1.8V.

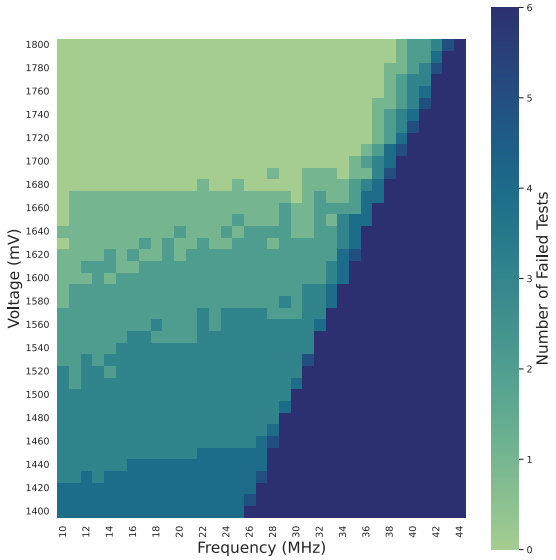

Fig. 6. Frequency and voltage analysis for representative die 4.

VII. CONCLUSION

Overall, we are pleased with the initial functionality of OR1 and look forward to verifying the MPW2 test chip. Beyond MPW2, we have also added enhancements for two additional test chips. These include a third Wishbone memory test interface and additional memory configurations.

REFERENCES

[1] M. R. Guthaus, J. E. Stine, S. Ataei, B. Chen, B. Wu, and M. Sarwar, "OpenRAM: An open-source memory compiler," in *ICCAD*, 2016, pp. 1–6.

[2] M. Guthaus, H. Nichols, J. Cirimelli-Low, J. Kunzler, and B. Wu, "Enabling design technology co-optimization of srams through open-source software," in *2020 IEEE International Electron Devices Meeting (IEDM)*, 2020, pp. 41.7.1–41.7.4.

[3] B. Wu, J. E. Stine, and M. R. Guthaus, "Fast and area-efficient sram word-line optimization," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

[4] B. Wu and M. R. Guthaus, "Bottom-up approach for high speed sram word-line buffer insertion optimization," in *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)*, 2019, pp. 305–310.

[5] H. Nichols, M. Grimes, J. Sowash, J. Cirimelli-Low, and M. R. Guthaus, "Automated synthesis of multi-port memories and control," in *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)*, 2019, pp. 59–64.

[6] "Freepdk45," https://www.eda.ncsu.edu/wiki/FreePDK45:Contents.

[7] "Skywater open source PDK," https://github.com/google/skywater-pdk.

[8] R. T. Edwards, M. Shalan, and M. Kassem, "Real silicon using open-source EDA," *IEEE Design & Test*, vol. 38, no. 2, pp. 38–44, 2021.

[9] "Efabless caravel harness SoC," https://caravel-harness.readthedocs.io/en/latest/.

[10] B. Amrutur and M. Horowitz, "A replica technique for wordline and sense control in low-power SRAM's," *JSSC*, vol. 33, no. 8, pp. 1208–1219, 1998.

[11] S. Ataei, J. E. Stine, and M. R. Guthaus, "A 64 kb differential single-port 12t sram design with a bit-interleaving scheme for low-voltage operation in 32 nm SOI CMOS," in *ICCD*, 2016, pp. 499–506.

[12] A. Ghazy and M. Shalan, "OpenLANE: The open-source digital ASIC implementation flow," in *Proc. Workshop on Open-Source EDA Technologies (WOSET)*, 2020.