UNIVERSITY OF PENNSYLVANIA ESE 5700

Project 1, Part 2: the 16×4 SRAM macro

Tyrone Marhguy • Victor Wanjohi

Cadence GPDK 18 nm $V_{DD} = 0.8$ V, nominal corner, 27°C Part 1 cell 0.39 μm^2, density factor $D = 28$
$T_{CLK,\min} = 0.69$ ns

Rubric map (handout, Part 2: "Report" list and 5-point grading)

Required item	Pts	Section	Status
Array layout with DRC/LVS summaries; extracted C_{BL}, C_{WL} vs. Part 1 estimate	1	§2	**Missing**: not drawn; estimates only
Decoder: logical-effort analysis; alternatives compared (predicted vs. measured); chosen implementation with sizes	1.5	§3	**Partial**: one design analysed and simulated; second design missing
Timing: derived WL pulse widths; annotated timing diagram; controller schematic; margin analysis	1.5	§4	**Partial**: pulse widths, timing diagram, schematic, margins done; +20% C_{BL} case missing
Column circuits with sizes and unit tests	(with above)	§5	**Done** at the stand-in load (3.5 fF)
Verification waveforms and pass/fail table	1	§6	**Done** (tests i–iii pass)
Metrics table with FOM, D and $T_{CLK,\min}$	(with above)	§7	**Partial**: all metrics measured; area is an estimate (array not laid out); write power uses a checkerboard pattern
Short paragraph on trade-offs		§8	**Done**
Team members and contributions		§1	Victor done; **Tyrone's line missing**

Red text marks work not done. All simulations are schematic-level with stand-in loads ($C_{BL} = 3.5$ fF in the column tests, device parasitics +0.62 fF in the macro, $C_{WL} = 0.5$ fF); they must be rerun with the extracted loads of §2.

1. Team members and contributions

Member	Contribution
Victor Wanjohi	Column circuits and unit tests (write driver, precharge, read path, output latch, read column); row decoder schematics and its simulation; FO4 characterization; macro and controller test benches and the clock-period, power and access measurements.
Tyrone Marhguy	**[TODO: fill in: array layout, extraction, controller, other]**

Summary. A 16-row by 4-column array of the Part 1 cell, a 4-to-16 decoder (2-2 predecode, NAND2 rows), and per column a precharge, write driver, 2:2 inverter read path and output latch, driven by a delay-chain controller. At $T_{CLK,\min} = 0.69$ ns the macro passes all three tests (checkerboard, complement, read-after-write), with read access 488 ps, write time 339 ps, write/read/standby power 124/115/112 μW, area 54.6 μm^2 *(est.)*, FOM $\approx 4.7 \times 10^3$ *(est.)*.

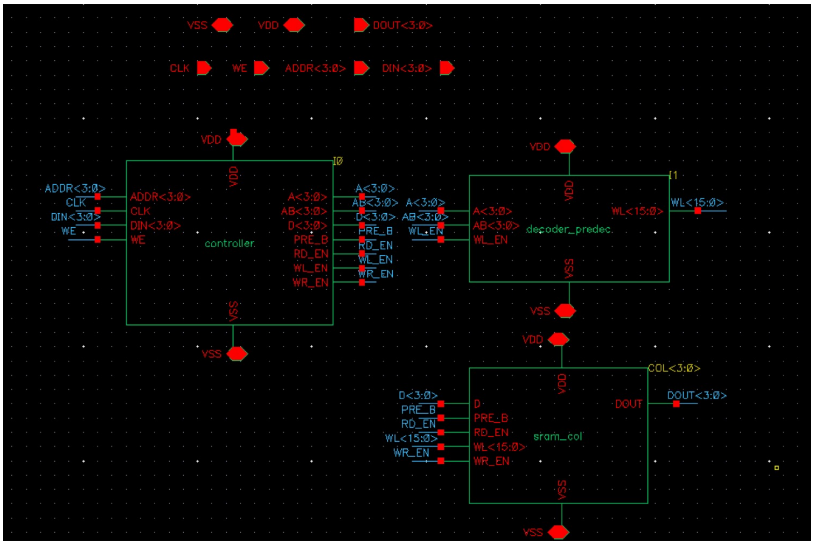


Figure 1: `sram_macro`: controller, decoder (`decoder_predec`) and four `sram_col` instances; ports match the handout interface (CLK, WE, ADDR, DIN, DOUT).

2. Array layout and extraction (2.1)

Rubric (1 pt): array layout, extraction, C_{BL}/C_{WL} vs. Part 1 estimate. **Status: missing.**

[TODO: Tiled 16×4 layout (mirrored rows and columns), DRC and LVS summaries, extracted C_{BL} per column and C_{WL} per row. Not done.]

Expected from the Part 1 per-cell extraction (180 aF per bitline node, 90 aF per wordline node), before wire: $C_{BL} \approx 16 \times 180$ aF $= 2.9$ fF per column and $C_{WL} \approx 4 \times 90$ aF $= 0.36$ fF per row. The simulations use 3.5 fF (column tests) and 0.5 fF (decoder), chosen to cover these plus wire.

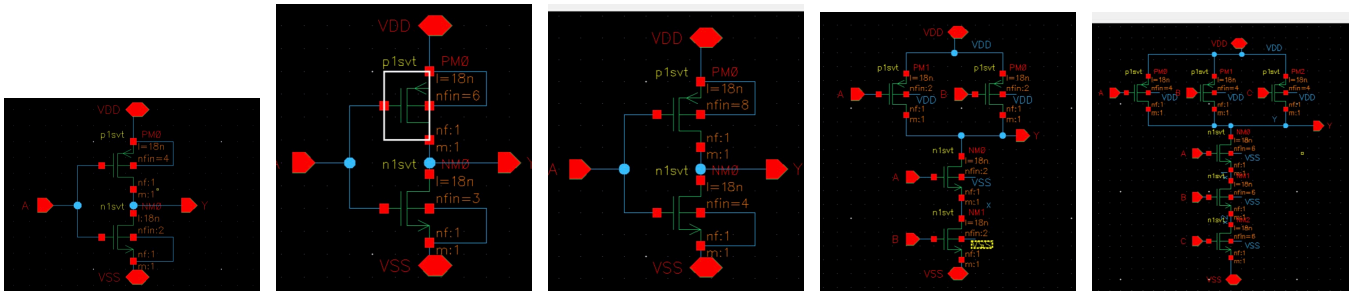
Area (Part 1 convention: $D = 28$, one fin $= 0.048$ μm, $L = 18$ nm, so 0.0242 μm^2 per fin; fin counts from the netlist, `netlists/tb_sram_macro.scs`):

Block	Instances	Fins	Area, μm^2
Array, 64×0.389 μm^2 (tiled, not laid out) *(est.)*	64	–	24.9
Row decoder	1	520	12.6
Controller (23 `inv_u`, 9 latches, 4 `inv_2x`, 3 `nand2_u`)	1	426	10.3
Write drivers / precharge / read paths / latches	4 each	144 / 24 / 16 / 96	3.5 / 0.58 / 0.39 / 2.3
Periphery $= D \sum W_i L_i$		1226	29.7
Macro total			**54.6**

3. Row decoder (2.2a, 2.2b)

Rubric (1.5 pt): logical-effort analysis; ≥ 2 alternatives, predicted vs. measured delay; chosen implementation with sizes. **Status: partial** (chosen design complete, second design missing).

Chosen implementation: 2-2 predecode. $\mathrm{PA}\langle j\rangle = \mathrm{WL_EN} \cdot (A_{1:0} = j)$ from `nand3_u`+`inv_2x`; $\mathrm{PB}\langle m\rangle = (A_{3:2} = m)$ from `nand2_u`+`inv_2x`; $\mathrm{WL}\langle 4m + j\rangle = \mathrm{PA}\langle j\rangle \cdot \mathrm{PB}\langle m\rangle$ from a per-row `nand2_u`+`inv_drv`. Sizes (fins per device, $L = 18$ nm): `inv_u` PMOS 4/NMOS 2 (unit); `inv_drv` 6/3; `inv_2x` 8/4; `nand2_u` 2/2 (half-strength); `nand3_u` PMOS 4, NMOS 6 (pull-down matches `inv_u`).



(a) `inv_u` (b) `inv_drv` (c) `inv_2x` (d) `nand2_u` (e) `nand3_u`

Figure 2: Decoder gates with sizes.

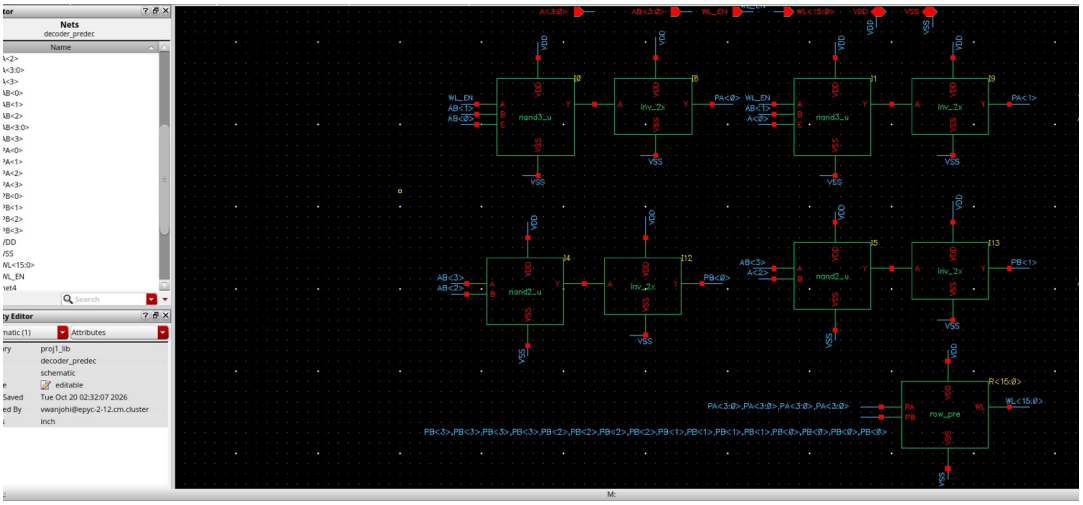


Figure 3: `decoder_predec`: predecode gates (top and middle) and the 16 `row_pre` instances.

Logical effort (address→WL, 4 gates: `nand3_u`, `inv_2x`, `nand2_u`, `inv_drv`). Assumptions: $\tau = 10.5$ ps (FO4 $= 5\tau$, below), ≈ 20 aF of gate capacitance per fin (Part 1: 90 aF per cell over 4 pass-gate fins), $p_{\mathrm{INV}} = 1$, $p_{\mathrm{NAND2}} = 2$, $p_{\mathrm{NAND3}} = 3$, $C_{WL} = 0.5$ fF. $G = \frac{5}{3} \cdot 1 \cdot \frac{4}{3} \cdot 1 = \frac{20}{9} = 2.2$; each literal reaches 2 predecode gates and each predecode output drives 4 rows, $B = 2 \times 4 = 8$; $C_{in} = 10$ fins $= 200$ aF (one `nand3_u` input), so $H = 500/200 = 2.5$ and $F = GBH = 44$; $\hat{N} = \log_4 F = 2.7$; at the 4 stages used the stage effort is $F^{1/4} = 2.6$ (below the optimum of ≈ 4, so the 4 stages are not buffer-limited); $P = 3 + 1 + 2 + 1 = 7$; $D = 4(2.6) + 7 = 17.3\,\tau \approx 182$ ps. This predicts the address-to-wordline path ($B = 8$ is the fan-out of an address literal). Measured below, with WL_EN held high and $A\langle 0\rangle$ switching (PA path): 106 ps for the new row rising and 112 ps for the old row falling, so the model overestimates by about 1.6× (182 vs. 112 ps). The 97 ps WL_EN-to-wordline delay is a different path and is not compared with the prediction.

FO4 delay. Chain of `inv_u` (`tb_fo4`), 0.4 V crossings: $t_{pHL} = 54.7$ ps, $t_{pLH} = 50.6$ ps, FO4 $= 52.6 \approx 53$ ps; $\tau = 10.5$ ps *(est.)*.

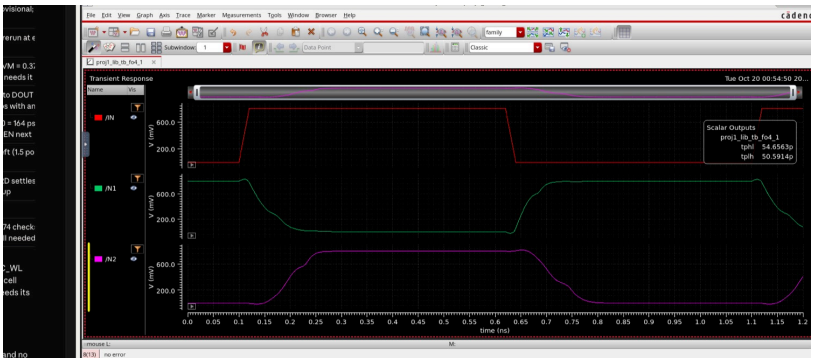


Figure 4: `tb_fo4`: IN, N1, N2 for a chain of `inv_u`.

Measured decoder delay (`tb_dec_pre`, 0.5 fF per wordline, WL_EN 300 ps with 20 ps edges, static address): WL_EN→WL rising 97.4 ps, falling 101 ps, identical at all 16 addresses (the enable always enters the same `nand3_u` pin). The peak of the summed wordlines is 824 mV at every address, i.e. exactly one wordline high (< 1 V) and it reaches high (> 0.72 V). `wl_idx` (row index weighted by wordline peak, over 0.824 V) returns the selected row:

Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
`wl_idx`	0.059	1.061	2.063	3.064	4.034	5.035	6.037	7.039	8.009	9.01	10.01	11.01	11.98	12.98	13.99	14.99

Each value is distinct and within 0.07 of its row (offsets are coupling glitches of about 2 mV on rows sharing PA$\langle 0\rangle$, weighted by row number: $(4+8+12)\times 2$ mV$/0.824$ V $= 0.058$ vs. 0.059 at address 0). So each address selects its own row.

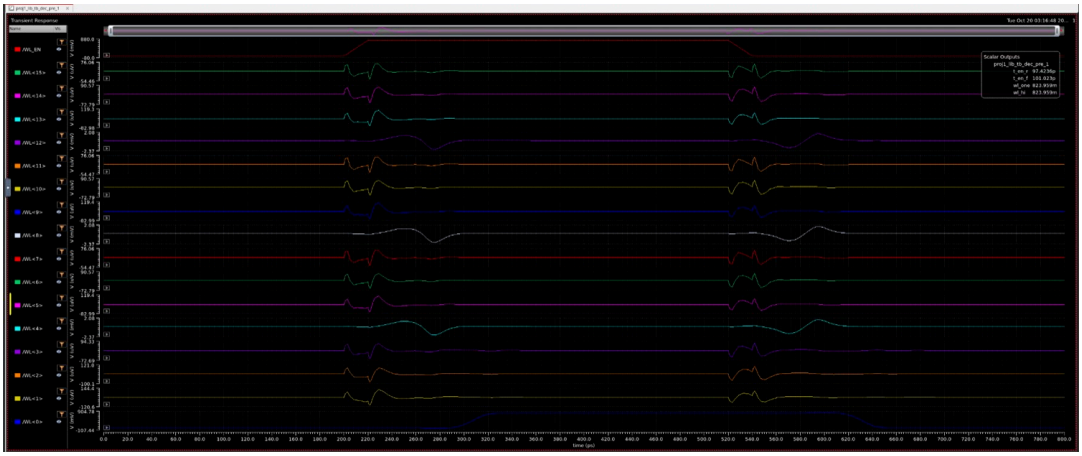


Figure 5: Address 0: only WL$\langle 0\rangle$ rises; other rows show ≤ 2 mV coupling.

Address-to-wordline delay (2.2a) and alternatives (2.2b). With WL_EN held high and $A\langle 0\rangle$ switching with its complement (the PA path, 0.5 fF per wordline), the address-to-wordline delay is 106 ps for the new row rising and 112 ps for the old row falling, against 182 ps predicted with $B = 8$: the logical-effort model overestimates by about 1.6×. The old and new rows overlap briefly (summed peak 1.02 V, above the 1 V limit of the static test), which is why the wordlines are gated by WL_EN; the falling WL_EN delay is not defined here because WL_EN does not switch.

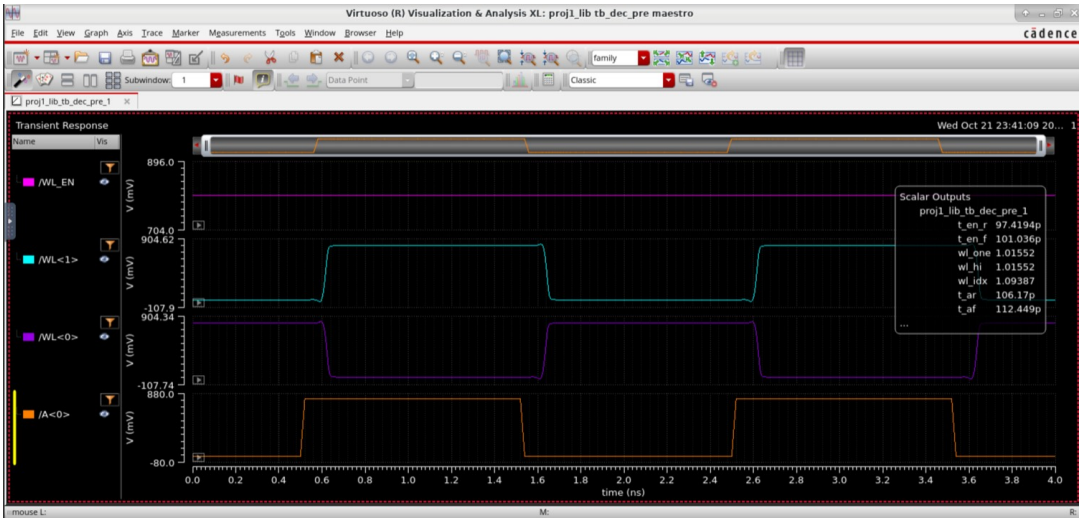


Figure 6: Address-to-wordline delay: $A\langle 0\rangle$ switches with WL_EN held high; WL$\langle 1\rangle$ rises 106 ps after the new address and WL$\langle 0\rangle$ falls 112 ps after it.

[TODO: Repeat with $A\langle 2\rangle$ (the PB path) and quote the slower; then the flat design (4-input NAND per row; WL_EN enters at the row NAND2, two stages from the wordline, so compare address-to-wordline for both decoders): logical effort, simulated delay, and the comparison.]

4. Timing and controller (2.2c, 2.4)

Rubric (1.5 pt): derived WL pulse widths; annotated timing diagram; controller schematic; margin analysis, incl. recovery if C_{BL} is 20% larger. **Status: partial.**

Derived pulse widths (2.2c).

Quantity	Value	Source
Bitline reaches the read inverter's trip point	83 ps	read-column test (§5)
Read: minimum WL pulse (50% width)	≈ 110 ps	83 ps plus margin *(est.)*
Write: minimum WL pulse (50% width)	100 to 120 ps	write-driver sweep (§5)
Design WL pulse (50% width)	140 ps	set by the write

At +20% C_{BL} the bitline time scales to ≈ 100 ps *(est.)*, inside 140 ps; the write sets the pulse.

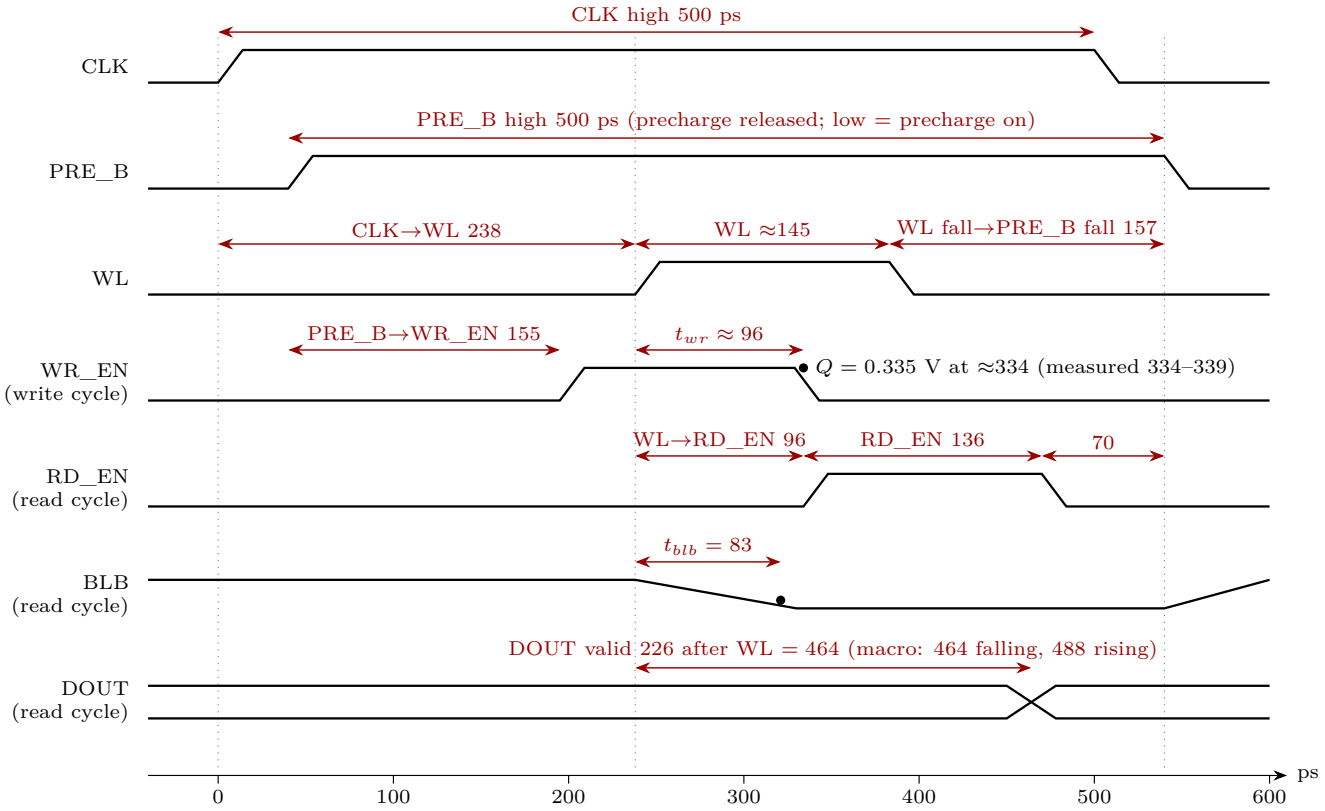


Figure 7: Annotated timing diagram (2.2c), $t_{clk} = 1$ ns, times in ps from the rising CLK edge; the WL pulse of about 145 ps exceeds the minimum read (110) and write (100 to 120) pulses. Measured: CLK→WL 238, WL→RD_EN 96, RD_EN width 136, WL and RD_EN fall 157 and 70 ps before PRE_B falls, PRE_B rises 155 before WR_EN, $t_{blb} = 83$, write time ≈ 96 after WL, access 226 after WL. Derived from these: WL fall at 540 − 157 = 383 ps; RD_EN from 334 to 470 ps; PRE_B edges ≈ 40 ps after CLK; the WR_EN width is taken equal to WL_EN's (not measured). BLB is a sketch (not to voltage scale). PRE_B low turns the precharge on, so the bitlines are released at the rising edge and recharge from 540 ps.

Controller (2.4). Inputs CLK, WE, ADDR, DIN; outputs A, AB, D, PRE_B, WL_EN, WR_EN, RD_EN. Input latches (`out_latch`, enable CLKBL from its own `inv_2x`) hold ADDR, DIN, WE while CLK is high. PRE_B = `inv_2x`(CLKB) follows CLK; the precharge cell is three PMOS gated by PRE_B, so precharge is on while CLK is low. WL_EN = CLKBUF AND (CLKBUF through 7 `inv_u`): a one-shot per rising edge. WR_EN = WL_EN AND WE; RD_EN = (WL_EN through 8 `inv_u`) AND $\overline{\text{WE}}$.

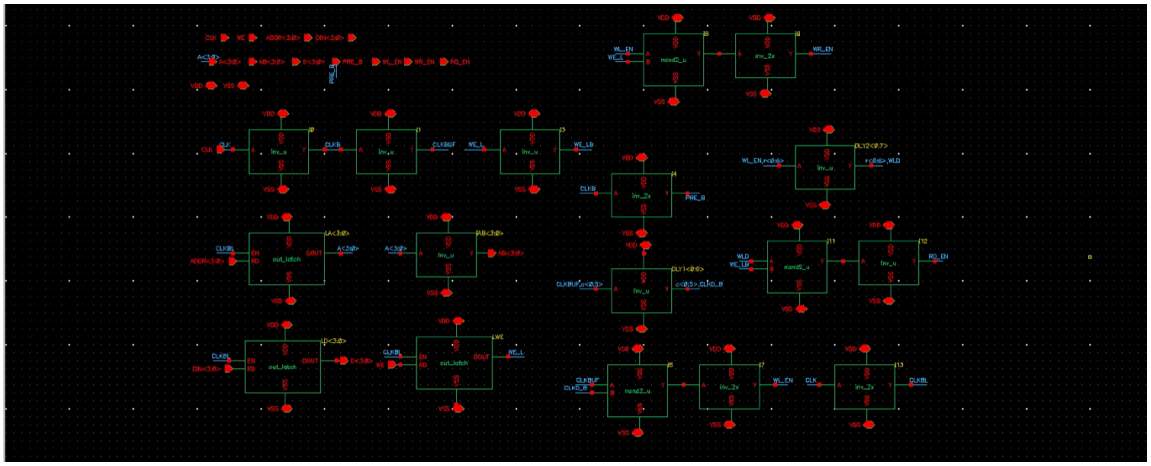


Figure 8: `controller`: input latches, WL_EN one-shot (DLY1), WR_EN and RD_EN (DLY2).

`tb_controller` (0.4 V crossings)	Before	After	Note
CLK → WL (`t_clk_wl`)	356 ps	238 ps	latch enable moved to own inverter
RD_EN fall before PRE_B fall (`m_rd_pre`)	30 ps	70 ps	smallest margin
WL fall before PRE_B fall (`m_wl_pre`)	118 ps	157 ps	
PRE_B rise before WR_EN rise (`gap_pre_wr`)	178 ps	155 ps	
WL → RD_EN (`t_wl_rden`)	96 ps	96 ps	column test needs ≥ 80
RD_EN width (`rden_w`)	136 ps	136 ps	column test needs ≥ 100
WL_EN width (`wlen_w`)	134 ps	134 ps	write needs 120 to 140

Margin on each edge (after): WL_EN width +14 ps over 120 ps; RD_EN lead +16 ps over 80 ps; RD_EN width +36 ps over 100 ps; RD_EN before precharge +70 ps; WL before precharge +157 ps; PRE_B before WR_EN +155 ps. No spurious pulses: RD_EN peaks at 3.3 mV in a write cycle and WR_EN at 0.41 mV in a read cycle (< 80 mV).

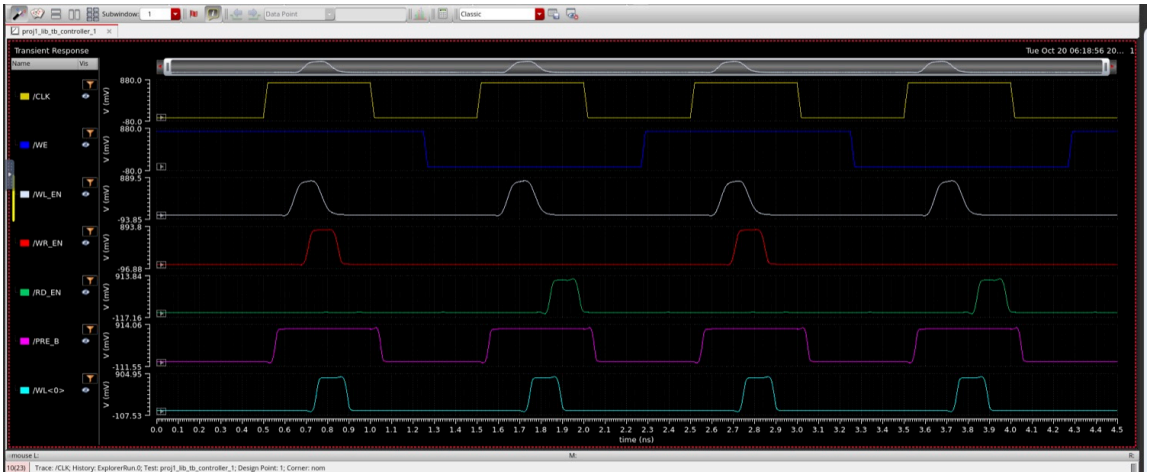


Figure 9: Controller transient: WR_EN pulses only in write cycles, RD_EN only in read cycles; WL ends before PRE_B falls.

Limit and recovery. The 70 ps RD_EN margin does not set $T_{CLK,\min}$: it was estimated to vanish near 0.86 ns, yet the macro passes down to 0.69 ns (§6, §7). The edge that fails below 0.69 ns is not identified. If C_{BL} were 20% larger the read delays grow, and the recovery is a shorter DLY2 chain or a longer clock half period. *The controller bench has no bitlines, so its loads do not model a larger C_{BL}; the pulses stay fixed and the column timing moves relative to them.* **[TODO: Test +20% C_{BL} in a bench with real bitlines: the macro run with `cbw` = 1.32 fF, or `tb_read_col` with C_{BL} = 4.2 fF, and report the margin on each edge.]**

5. Column circuits and unit tests (2.3)

Rubric ((within the 1.5 pt controller/column item)): precharge, write driver, read path (state the differential and margin), output latch, each unit-tested. **Status: done** at $C_{BL} = 3.5$ fF stand-in; **[TODO: rerun with the extracted C_{BL}]**.

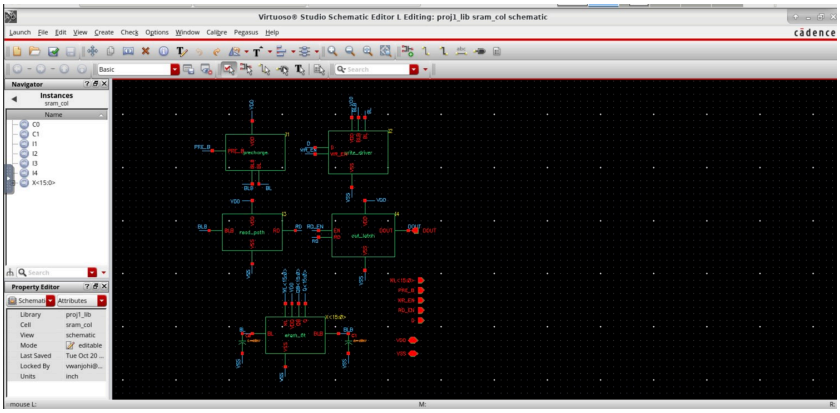


Figure 10: `sram_col` (one per bit): precharge (I1), write driver (I2), read path (I3), output latch (I4), the 16 cells `X<15:0>`, and the wire capacitance `cbw` on BL and BLB.

`sram_col` has no standalone bench: `tb_read_col` (below) tests the same four circuits on one cell with C_{BL} on both bitlines, and `sram_col` itself is verified in `tb_sram_macro` (four instances, all 16 rows, tests (i) and (ii)) and `tb_sram_macro_t3` (test (iii)), §6.

Precharge: three PMOS (2 fins each: one per bitline to VDD, one equalizer), gate PRE_B. Recovery at $C_{BL} = 3.5$ fF: BL reaches 99% of V_{DD} in 142 ps, with a peak of 845 mV and BLB near 690 mV during equalization.

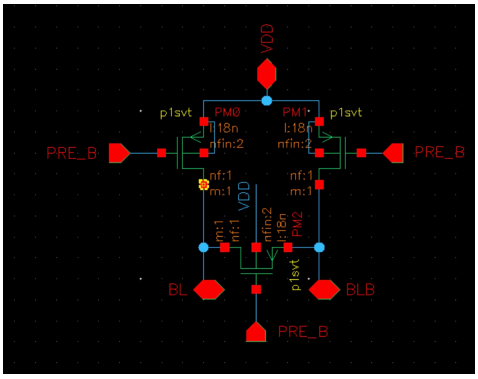


Figure 11: `precharge`.

Write driver (36 fins: four 8-fin NMOS pull-downs plus a 2+2-fin inverter). A small driver fails because the write races an undriven bitline: the cell discharges BLB through its pass gate, pinning QB low and blocking the cell's feedback. Fin sweep, pulse 0.3 ns, $C_{BL} = 3.5$ fF:

nfin	Write 0	t_{wr0}	t_{wr1}	lowest BLB (mV, from waveform)
2	fail, Q stalls at 0.66 V	–	–	∼ 0
4	fail, Q stalls at 0.38 V	–	–	∼ 60
5	pass	153 ps	154 ps	∼ 190
6	pass	121 ps	121 ps	∼ 260
8	pass	96 ps	96 ps	∼ 345

`pw_w1` (WL at 50% = +20 ps)	C_{BL} = 3.5 fF	4.2 fF (+20%)
40p, 60p, 80p	fail (Q stalls 0.62, 0.46, 0.27 V)	fail (0.66, 0.51, 0.34 V)
100p, 120p, 150p	pass, 96 ps	pass, 102 ps

nfin 5 is the smallest that writes; nfin 8 is provisional (6 vs. 8 trades 0.77 μm^2, 3% of the array, for 25 ps; decide with $T_{CLK,\min}$). Write time is 96 ps, 102 ps at +20% C_{BL}; minimum pulse 100 to 120 ps at 50%.

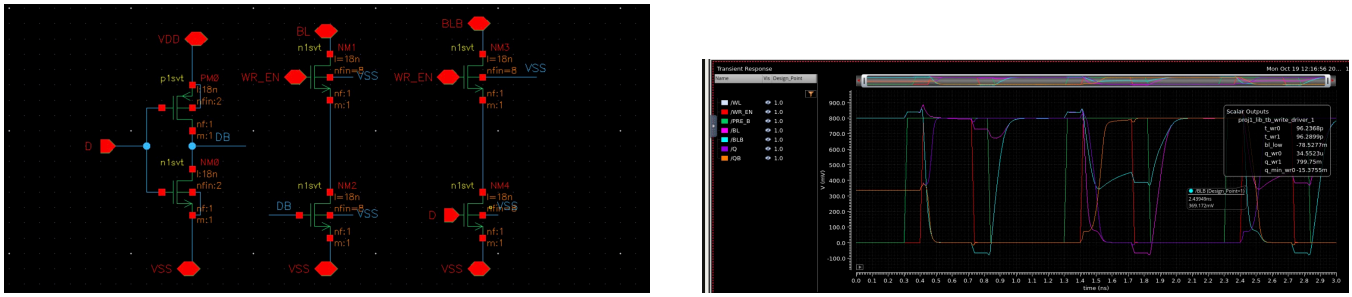


Figure 12: Write driver at nfin 8 and its write-0/write-1 transient ($q_{wr0} = 35$ μV, $q_{wr1} = 799.8$ mV, $t_{wr0} = 96$ ps).

Read path: inverter on BLB, full swing (no sense amplifier), sized 2:2 (PMOS 2, NMOS 2 fins), switching point 0.376 V, i.e. BLB must fall $0.8 - 0.376 = 0.42$ V to flip RD (margin: the read pulse of 140 ps gives 57 ps of slack over the 83 ps bitline time). 4:2 would switch at 0.415 V, save ≈ 7 ps *(est.)* and add 8 fins (0.19 μm^2); it is not on the critical path since the write sets the pulse.

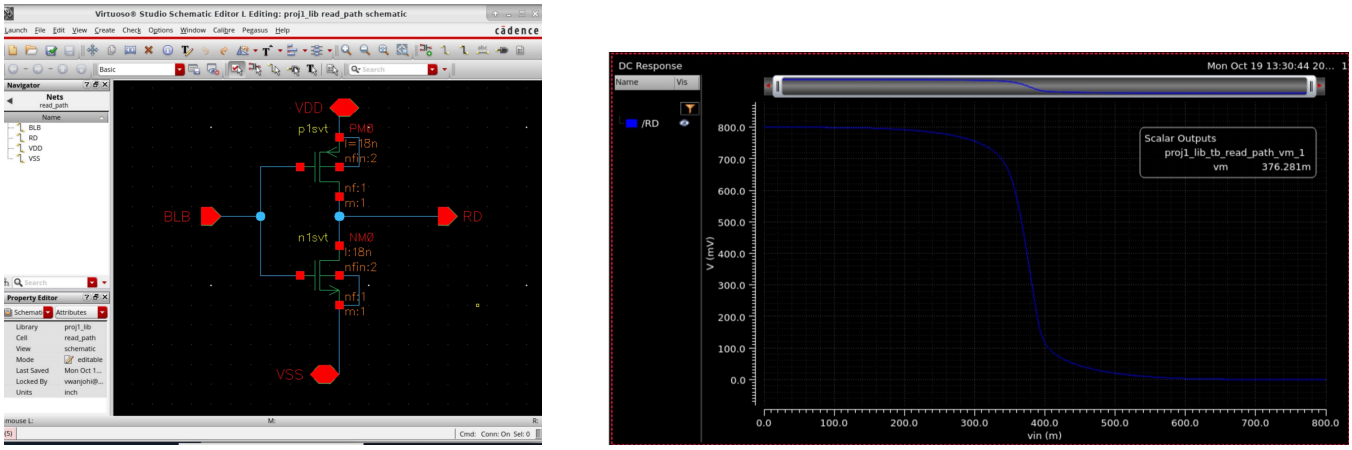


Figure 13: `read_path` (2:2) and its transfer curve, $V_M = 376$ mV.

Output latch (all devices 2 fins, 24 fins). Enable-pulse sweep ($c_{dout} = 1$ fF; specs hold-1 > 0.72 V, hold-0 < 0.08 V):

`pw_en`	$t_{en,r}$	$t_{en,f}$	hold-1	hold-0	
10p	–	–	-58 μV	77 μV	fail
20p	99.5 ps	76.0 ps	799.8 mV	88 μV	pass
40p	88.7 ps	74.8 ps	799.8 mV	89 μV	pass
80p	87.2 ps	74.5 ps	799.8 mV	125 μV	pass
150p	87.3 ps	74.5 ps	799.2 mV	1.25 mV	pass

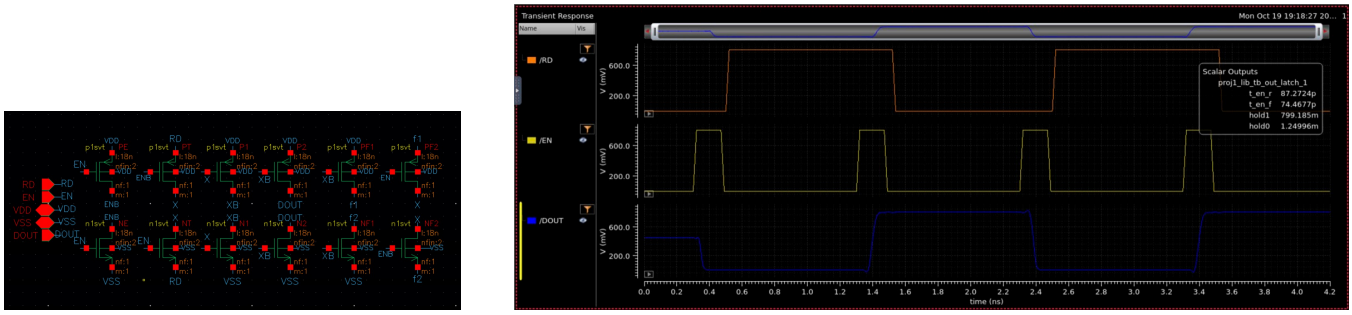


Figure 14: `out_latch` and its unit test at 150 ps (DOUT written only while EN is high).

Read column (cell, write driver, precharge, read path, latch; write 1, read, write 0, read, write 1, read). Read-1 access t_{acc1} (WL→DOUT, ps) over latch-enable delay `rd_dly` and width `pw_rden` (24 points); all checks pass except the third read at `pw_rden` = 40 p (latch never captures the 1):

`rd_dly`	40p	80p	120p	160p	t_{acc0}
1.48n	fail	247	226	226	165
1.52n	fail	235	235	235	205
1.56n	fail	271	271	271	245
1.60n	fail	311	311	311	285
1.64n	fail	351	351	351	325
1.68n	fail	391	391	391	365

Opening the latch later only adds delay (40 ps per 40 ps of `rd_dly`). **Chosen: `rd_dly` = 1.48 n, `pw_rden` = 120 p** (shortest width that loses nothing): BLB reaches the read inverter's switching point 83 ps after WL, RD crosses 0.4 V at 113 ps (140 ps with the latch loading it), DOUT valid 165 ps (read 0) and 226 ps (read 1) after WL; the cell is undisturbed (q_{rd1} = 799.7 mV, q_{rd0} = 39 μV).



Figure 15: Cycle 6 read of a 1, C_{BL} = 3.5 fF, `rd_dly` = 1.48 n, `pw_rden` = 120 p: BLB crosses the read inverter's 0.376 V switching point 83 ps after WL rises; DOUT is valid 226 ps after WL rises.

6. Verification (2.5)

Rubric (1 pt): full-macro checkerboard write/read, complement overwrite/read, worst-case-row read-after-write; waveforms for (iii), pass/fail table for (i)–(ii); inputs through realistic buffers. **Status: done.**

Bench (`tb_sram_macro`, t_{clk} swept, DOUT loaded by 1 fF): CLK, WE and ADDR pass through two `inv_u` stages each. The address is a binary counter (one step per cycle); WE is high for 16 cycles (write all words), then low for

16 (read all), repeating. DIN $= \{X, \bar{X}, X, \bar{X}\}$ with $X = a_0 \oplus ph$: each word is 0101 or 1010 alternating by row, and ph flips at $32.25\,t_{clk}$ to complement every word. Checks: `err1` (checkerboard read pass) and `err2` (complement read pass), each the maximum DOUT error over the read window against 0.1 V; `err3` is test (iii).

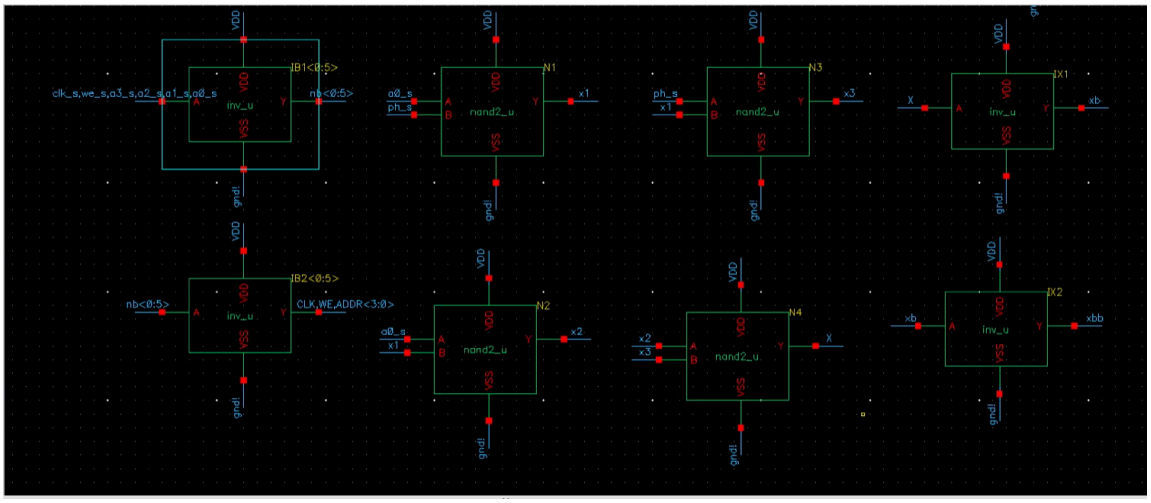


Figure 16: Input buffers (IB1, IB2) and the NAND XOR that builds the DIN pattern.

Test	Result at $t_{clk} = 1$ ns	
(i) checkerboard write, read all 16	`err1` $= 173.8\ \mu$V (< 0.1 V)	**pass**
(ii) complement overwrite, read all 16	`err2` $= 173.8\ \mu$V	**pass**
(iii) far-end row (row 15), read right after a write to the opposite value	`err3` $= 173.8\ \mu$V	**pass**

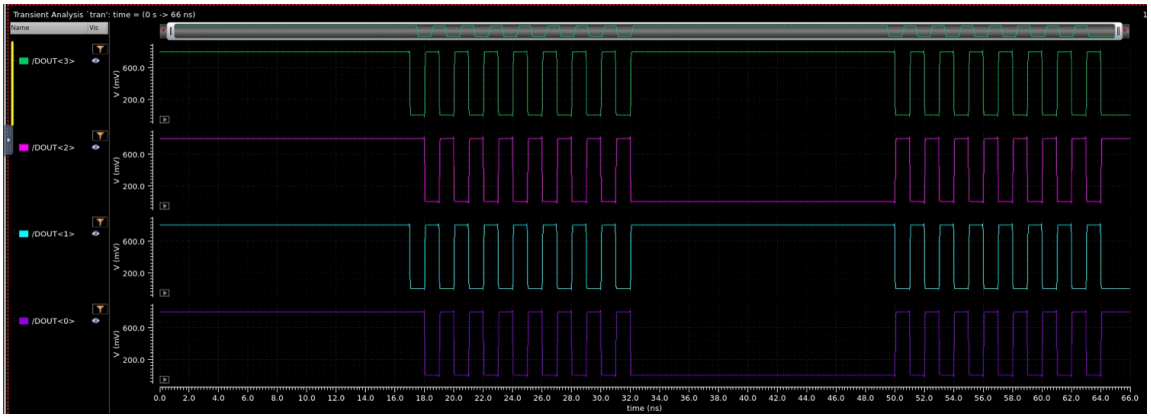


Figure 17: DOUT⟨3:0⟩: bits 3,1 opposite to bits 2,0 (0101/1010 alternating each cycle); DOUT holds while WE is high (32 to 49 ns).

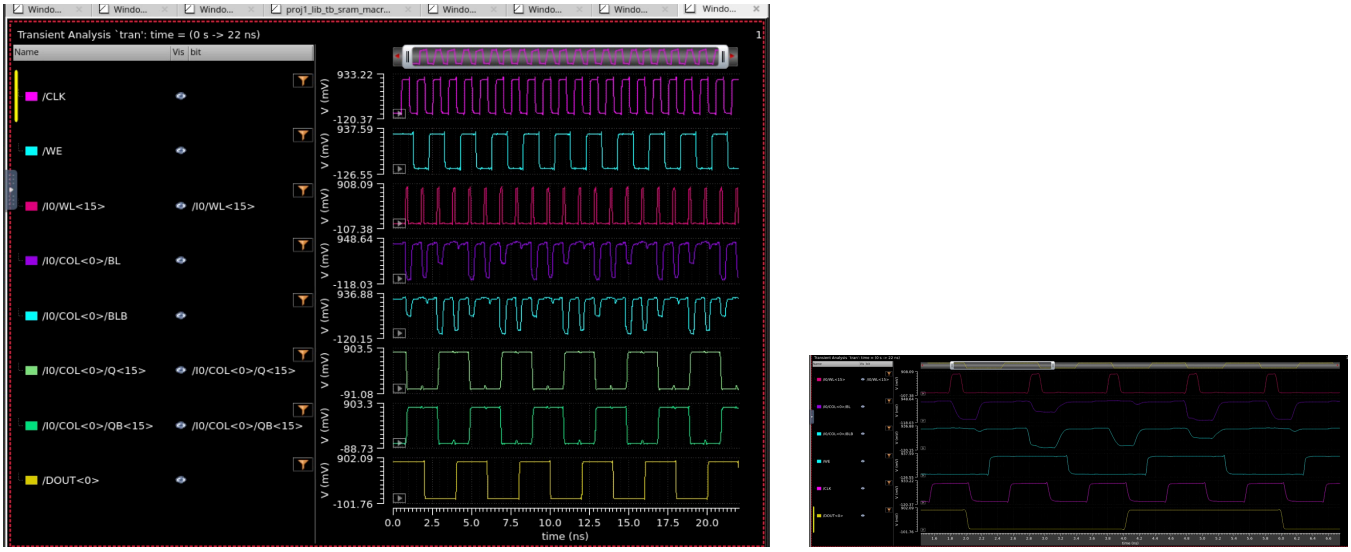


Figure 18: Test (iii), row 15 (the far-end row, a stand-in: at schematic level all rows are equivalent): WE alternates every cycle; each read returns the value just written, which is the opposite of the previous one. DOUT changes only after the reads (about 0.5 ns after the edge).

Write time (later of the Q and $\overline{Q}$ crossings of 0.335 V, from the CLK edge): 338.6 ps (write 1), 333.7 ps (write 0); the cursors give 333.3 ps (CLK 448 mV to Q 306 mV). **Read access** (CLK to DOUT⟨1⟩, 0.4 V): 463.7 ps falling, 484.2 ps rising at 1 ns.

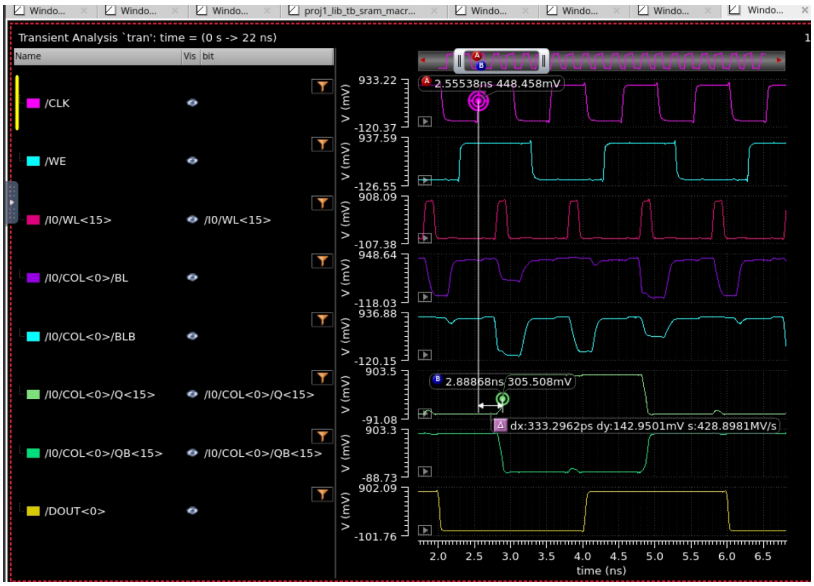


Figure 19: Write time on row 15: cursor A at the CLK edge, cursor B on $Q\langle 15\rangle$, 333.3 ps.

Clock-period sweep (t_{clk} from 1 ns down to 0.6 ns; power is the macro's VDD-pin current ×0.8 V over the write pass, read pass, and a clock-stopped window):

t_{clk}	err1	err2	Result	$t_{acc,f}$	$t_{acc,r}$	write	read
700 ps	5.10 mV	5.12 mV	pass	463.7	485.5	122.6 μW	113.4 μW
690 ps	23.3 mV	22.4 mV	**pass**	463.8	488.3	124.3 μW	115.2 μW
680 ps	452 mV	441 mV	fail	463.8	496.9	126.0 μW	118.0 μW
670–600 ps	1.28 V	1.28 V	fail	463.9–465.3	–	127.7–142.8 μW	114.9–122.8 μW

(1000 to 750 ps also pass: `err1`/`err2` grow from 174 μV to 0.50 mV; `err3` from 174 μV to 0.50 mV, 5.6 mV at 700 ps, about 30 mV at 690 ps, about 0.6 V at 680 ps, 1.28 V at 670 ps and below; write times stay at 338.6/333.7 ps.)

Below 680 ps the read returns the wrong value (1.28 V error). $T_{CLK,\min} = 0.69$ ns (10 ps resolution): 690 ps passes all three tests with about a factor of 3 of margin on `err3`, 680 ps fails.

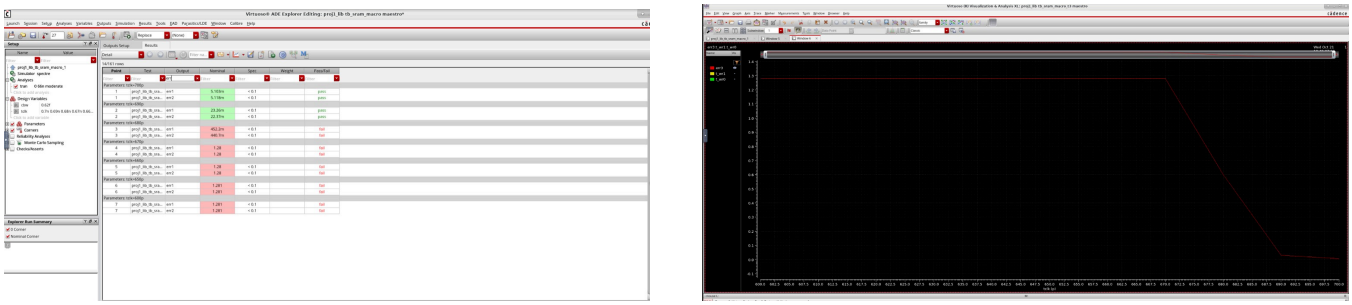


Figure 20: Error against clock period: `err1`/`err2` (left) and `err3` (right); the tests fail below 0.69 ns.

Test-bench notes. The a_3 source has width $8.25\,t_{clk} - 20$ ps instead of $8\,t_{clk} - 20$ ps, so a_3 falls at the capture edge of cycle 16; the width has been corrected to $8\,t_{clk} - 20$ ps in the saved bench. The results above come from the bench before the correction and are unaffected: rows 0 and 8 hold the same data, and test (iii) uses a fixed address and fails at the same period. The checks compare DOUT to a pattern that depends on a_0 only; correct row selection is shown by the `wl_idx` test of §3.

7. Metrics and figure of merit

Rubric ((within the 1 pt verification/metrics item)): metrics table with FOM, D and $T_{CLK,\min}$. **Status: partial** (area and write power are estimates).

At $V_{DD} = 0.8$ V, nominal, 27°C, $T_{CLK,\min} = 0.69$ ns, $D = 28$ ($0.39\ \mu\text{m}^2/(16 \times 0.048 \times 0.018\ \mu\text{m}^2)$):

Metric	Value
$T_{CLK,\min}$	**0.69 ns** (10 ps resolution)
Worst-case read access (CLK to DOUT)	488 ps at 690 ps (488.3 ps on DOUT⟨1⟩, 488.9 ps on DOUT⟨0⟩): the columns are identical at schematic level and the decoder delay is the same for every row, so the worst case depends on the data (a read of 1, rising output), not on the bit or the address
Worst-case write time (later of $Q/\overline{Q}$)	339 ps (row 15)
Power, array write at $T_{CLK,\min}$	124.3 μW, measured with a checkerboard write pattern (second write pass), not the handout's all-0 then all-1 sequence
Power, checkerboard read sweep	115.2 μW
Power, standby	112.2 μW clock running (no chip-select: every edge triggers an access); 3.99 μW clock stopped (leakage)
Area: array (not laid out) + periphery ($D\sum W_i L_i$)	$24.9 + 29.7 = 54.6\ \mu\text{m}^2$ *(est.)*
FOM = **Area**·**Power**·$T_{CLK,\min}$	$54.6 \times 124.3 \times 0.69 \approx \mathbf{4.7 \times 10^3}\ \mu\text{m}^2 \cdot \mu\text{W}\cdot\text{ns}$ *(est.)*

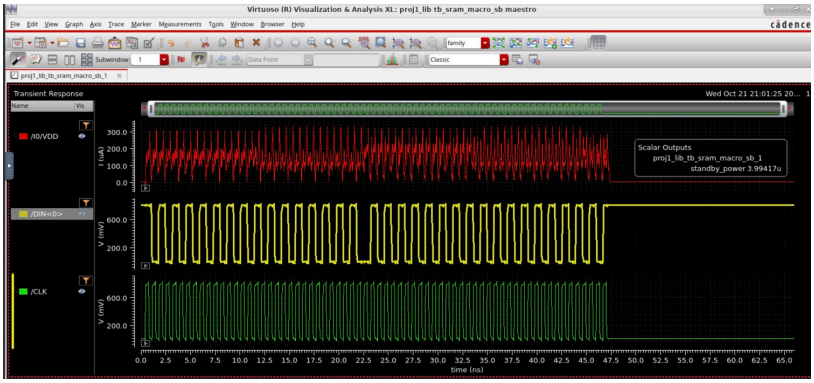


Figure 21: Standby: after the clock stops (47 ns) the macro's VDD current falls to about zero (3.99 μW).

8. Trade-offs

- **Write-driver size:** nfin 5 is the smallest that writes (153 ps); 8 fins gives 96 ps for 0.77 μm^2 (3% of the array) over 6 fins; provisional.
- **Read inverter 2:2 vs. 4:2:** 2:2 saves 8 fins and costs $\approx$ 7 ps *(est.)*, off the critical path since the write sets the wordline pulse.
- **Latch timing:** opening RD_EN early (1.48n) gives the shortest access (226 ps) with no glitch; 120 ps is the shortest pulse that loses nothing.
- **Controller:** moving the latch enables to their own inverter cut CLK-to-WL from 356 to 238 ps and doubled the thinnest margin (30 to 70 ps) for 12 fins; the periphery is 54% of the area, mostly the 9 controller latches.
- **No clock gating:** a running clock always accesses the array, so "standby" with the clock running is 112 μW against 4 μW stopped; a chip-select would remove it.