Open-source Toy-SRAM Test chip
High Specific Bandwidth

We make high specific bandwidth multiport memories child's play

Suggesting a first-class citizen 10T-SRAM, which is pumped & replicated for additional ports

To pare much of the custom circuit design from processors

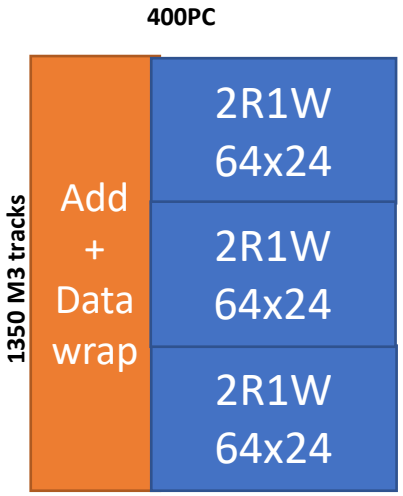
Draft Version 0.2

What is specific bandwidth? Why does Toy-SRAM do so well?

- It measures the Read & Write bandwidth per unit area for a 64Reg x 8B
  - It is an analog to specific gravity which is mass per unit volume – Bandwidth per unit area
  - Its more encompassing than bit density, which drives complexity to improve bandwidth
- It is enhanced by having a 10T SRAM separate 2 read + 1 write ports and
  - Supports low-cost super-pipelining.(2x + the system frequency without latch overhead)
  - Enables energy efficient ultra-low voltage operation by avoiding read disturb
- Its Read & Write specific bandwidth can be expressed with two metrics:
  - Technology dependent "X TB/(sec * mm^2)"
  - Technology independent "Y 1/(FO4 delay * PC PITCH * min horizontal metal pitch)"
- We would demonstrate specific bandwidth results from 130nm. to 2nm.
  - And use it to grow as many ports as necessary through replication
  - To produce as more efficient processors/accelerators with less circuit effort.

Test Structures for inclusion in a Toy-SRAM Test Site

Experiment →



=2R1W 64x72

Blue – custom cell

Orange: synthesized hard core

- 72 bits x 64 regs 2 clocking modes regfile. ALL Signals are LATCH BOUNDED
  - 2R 1W SingleDataRate,
    - Its synthesized wrapper contains:
      - 250 latches and for the data in/out
      - LCB's, clock generation, and pre-pre-decode
  - 4R2W DoubleDataRate R&W
    - Its synthesized wrapper contains:
      - 500 latches and for the data in/out
      - LCB's, clock generation, and pre-pre-decode
      - Double pump mux stage at border minimizing double pump path depth.
- Updated array sizing

  64 registers ix 8B with SEC-DED using 3 3B sub-arrays

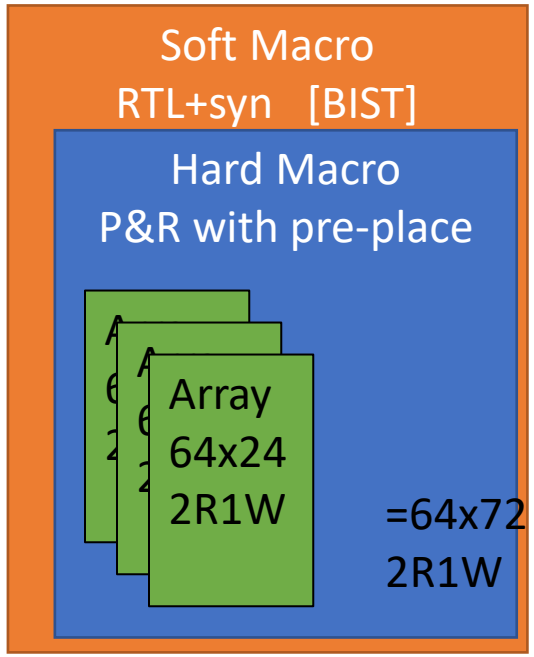
  ~500-1000 nets from the ADD/CLK wrap to the cores 95% in M3, 5% D5

->Pipe clean With SDR and Validate DDR methodology/characterization

->Measure performance from silicon, and ease technology migration through templates

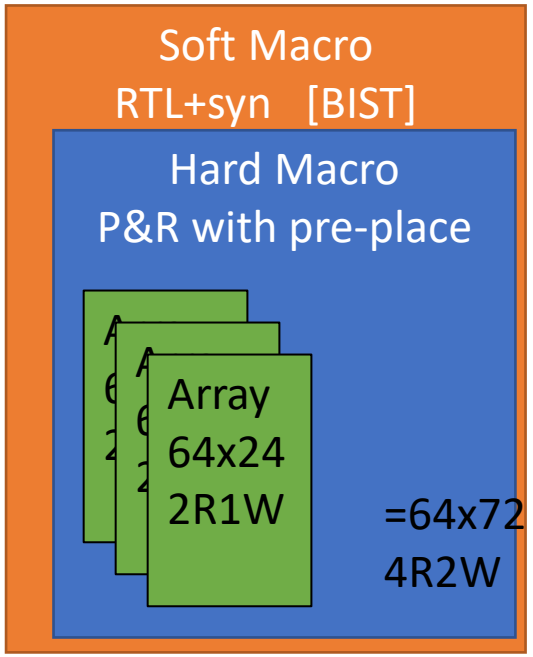
Testsite definition. Proposed 2 experiments

Experiment and testsite top levels



Single Pump Read & Write

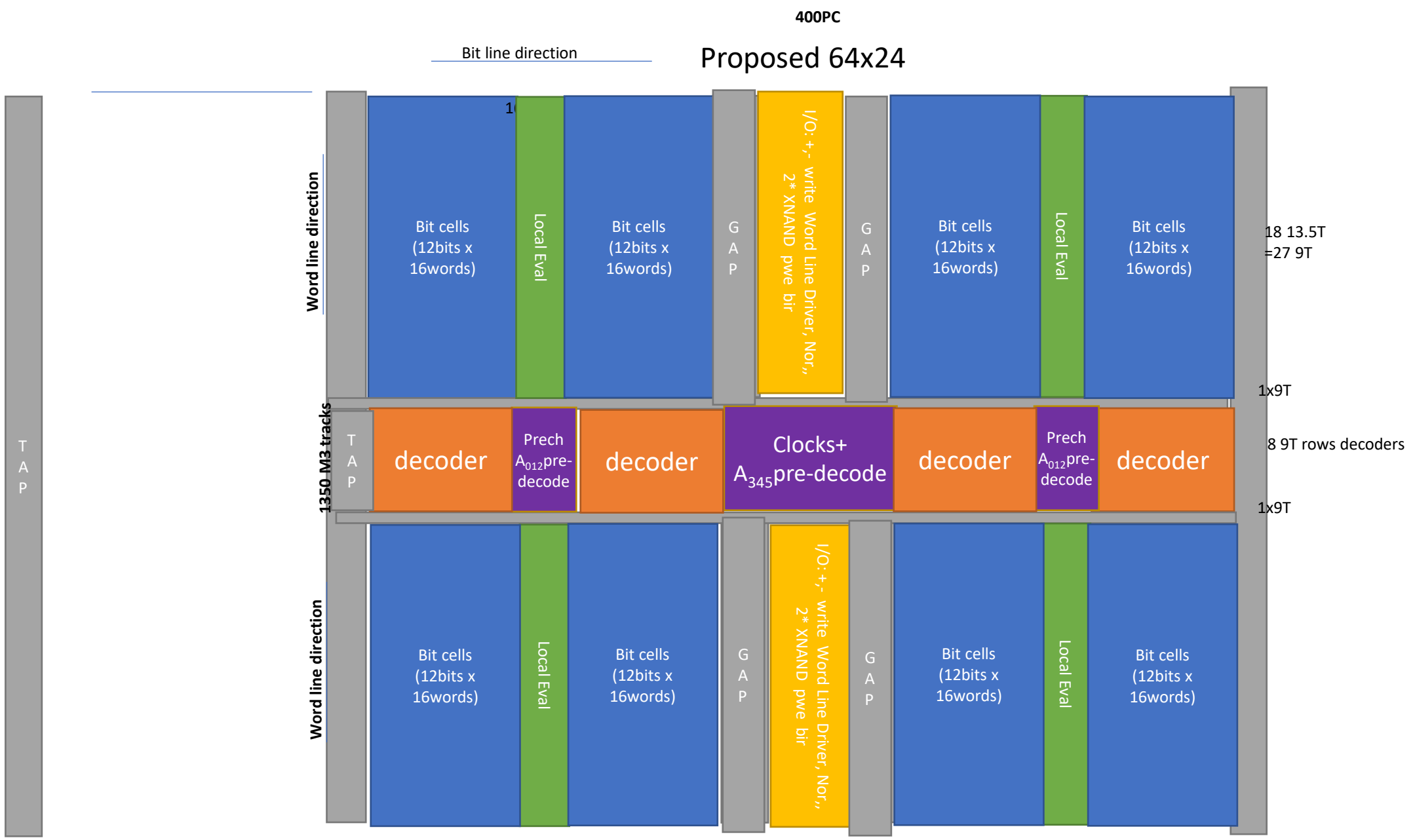
THEN



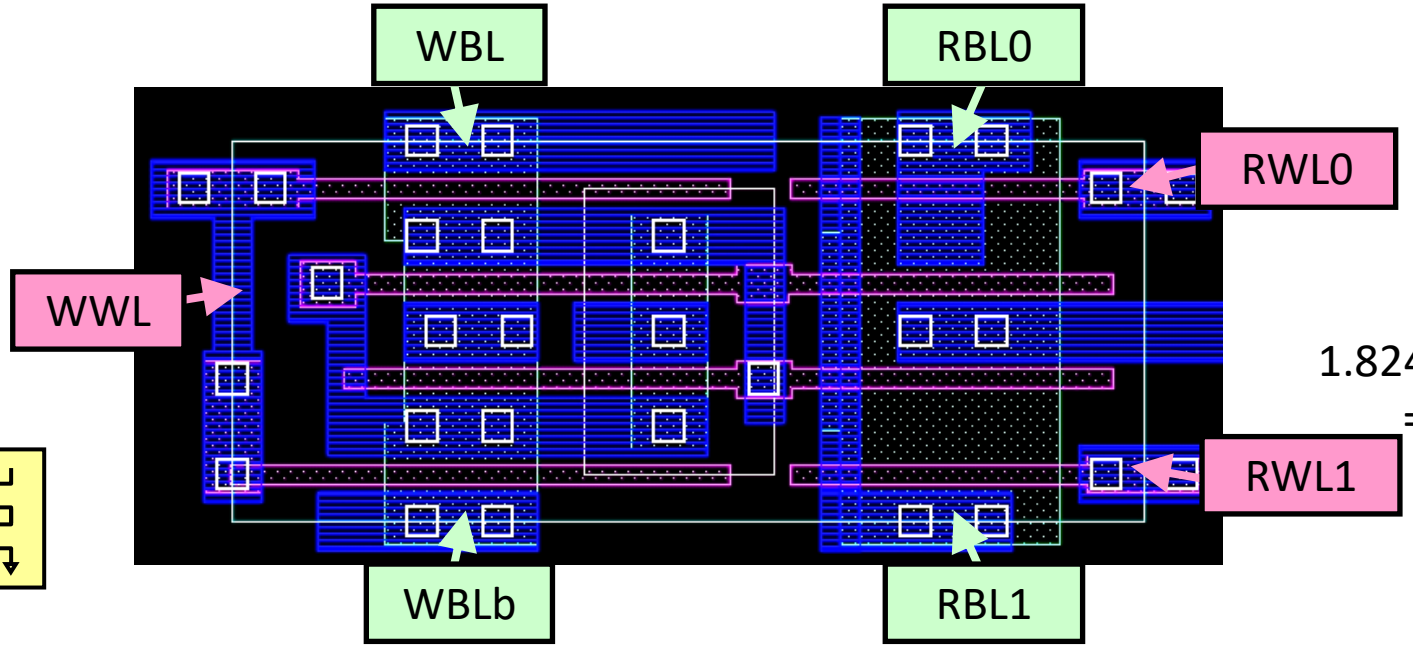
Double Pump Read & Write

Similar array between 2 experiments

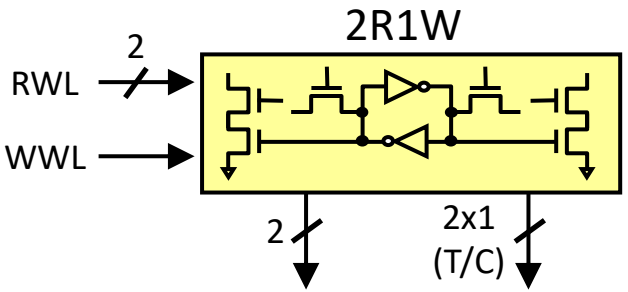
Sub-Array - placement of pre-decoders, decoders et al :



2R1W Cell Layout Optimization. Rotated in 45 nm



Cell size
1.824μm × 0.76μm
= 1.39μm²



- Three ports (2R1W) is optimal for cell layout density
  - Terminals at cell edge: Contacts shared w/neighboring cells
  - Neither strongly wire-limited or device-limited
- WL and BL width/spacing optimized to maximize performance

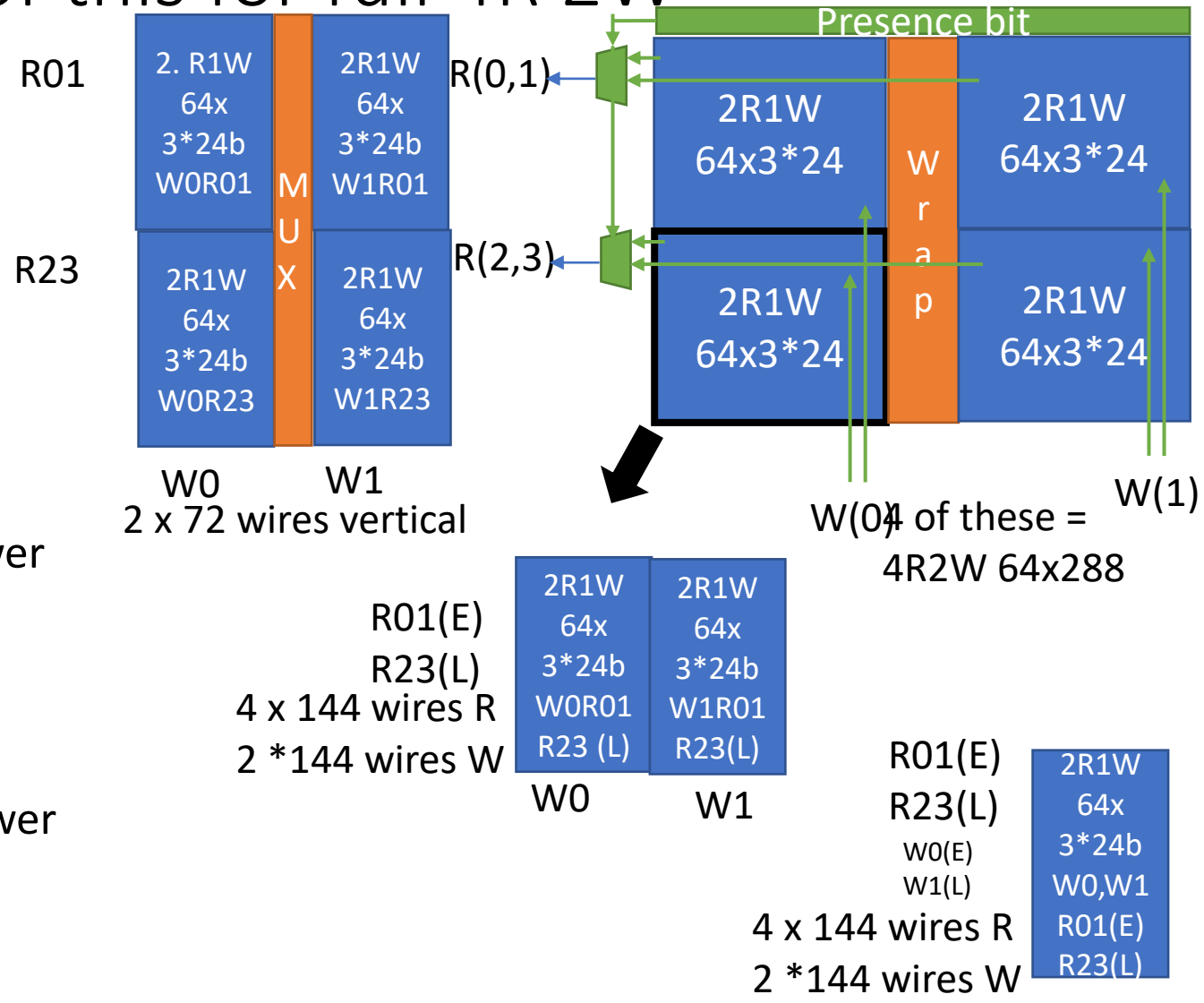
Compact/Composable, Double Pumped (4W2R 64 regs x 8B) for wiring impact, area power, delay
2 copies of this for full 4R 2W

<u>3 cases of increasing ckt effort</u>

1) SDR read and write. (VS REF design)
   - ~same area ~80% read ~140% write power
   - ~12FO4 latency

2) SDR – READ, DDR WRITE (VS REF)
   - ~50+% area ~40+% read ~140+% write power
   - ~10FO4 latency

3) For DDR- READ+WRITE (VS REF)
   - 2-4 layers of wires opened
   - ~25+% area ~40+% power ~70+% write power
   - ~10FO4 & 20FO4 latency



Toy-SRAM key components

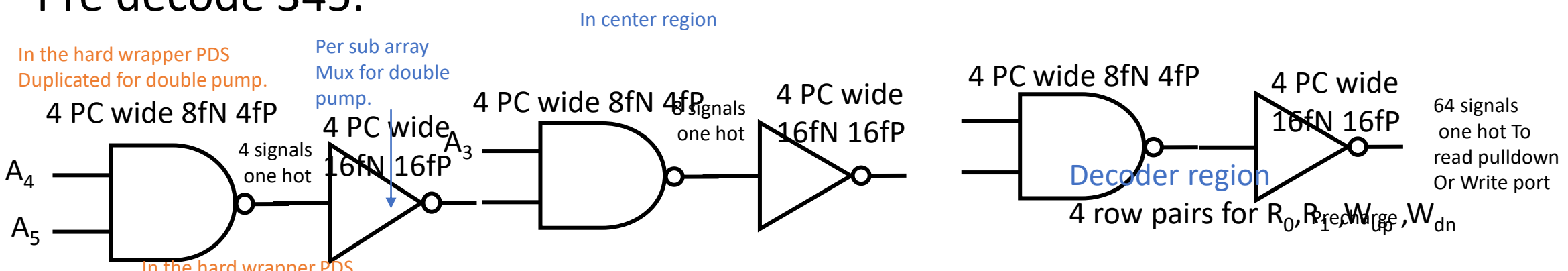
- SIZE-SPEED A 10T SRAM dense library cell and its associated, well-tuned hand placed, standard cell decoders, and I/O circuitry in a hard macro. The array of 64 registers of 24 bits so that 3 copies produce 72-bit SEC-DED error correction + redundancy with efficient wiring below on half of D5 and below, This array + wrapper is 2/3 of the POR 2R1W array's area , because it uses the 10T SRAM cell, and the decoders and I/O circuitry are hand placed and segregated from:
- CLOCKING-CONNECTION A CAD friendly "wrapper" with the latches, clocking system completely done with combinatorial logic, and muxes, for BIST, redundancy and others. When used correctly the SDR or early read is as fast as any high performing array and the late read is less than the delay of most.
- COPYING FOR ADDITIONAL PORTS An algorithm to expand read or write ports without redesigning the hard macro & A potential factor of 2 interleave, getting single port access to 2x array width for register pairs
- This system can replace many of the existing register files by double pumping Write & Read (if the latency is tolerable) and duplicating for any additional ports.
- It is the most labor efficient way to produce or extend a multi-ported file, and usually more silicon efficient, once we have the Toy-SRAM in silicon.

Sketches of non-traditional design changes

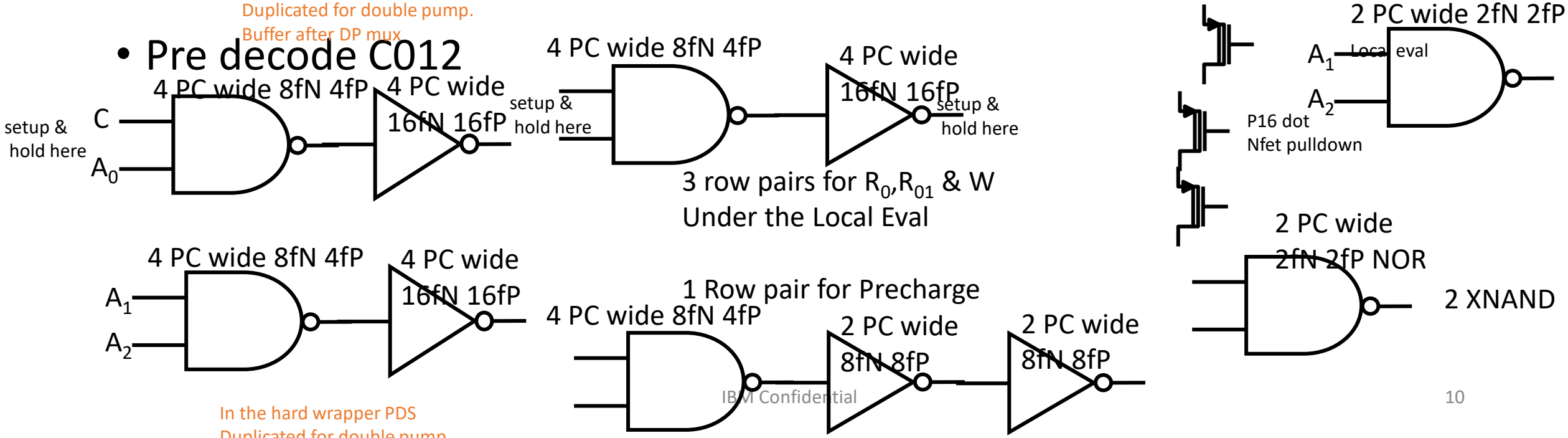
- This is a 6-bit address (64 register) decoder, which uses 6 bits and and enable pulse in order to minimize the custom core size and complexity we did a little pre-decode in the synthesized wrapper.
  - This requires 12 signals for 6-bit address and enable
    - Bit 0 and its complement is ‘ANDED’ with the enabled clock pulse
    - Bit 3 and is complement are brought into the array
    - Bits 1,2 and 4,5 are expanded to one hot signals out of 4 each.
- Distribute the local pre-decode in available spaces in the design
- Align a write circuit and 2 copies of 2 state holding XNAND latches (one early and one late for each read port) with the bit-cell.
- Allocate the wires from the wrapper, and global wires within the array
- Assumptions (only a little of the the deign depends on this):
  - 9 Track high cell library. a range of device widths and at least 1 M1 free across the row .
  - 13.5T x 4 PC 10T register file cell - a conservative (near ground rule clean) design

Read path 2, Write path 1 (w/dup decoder) Each circuit has a delay of about 1FO4.

- Pre decode 345.



- Pre decode C012



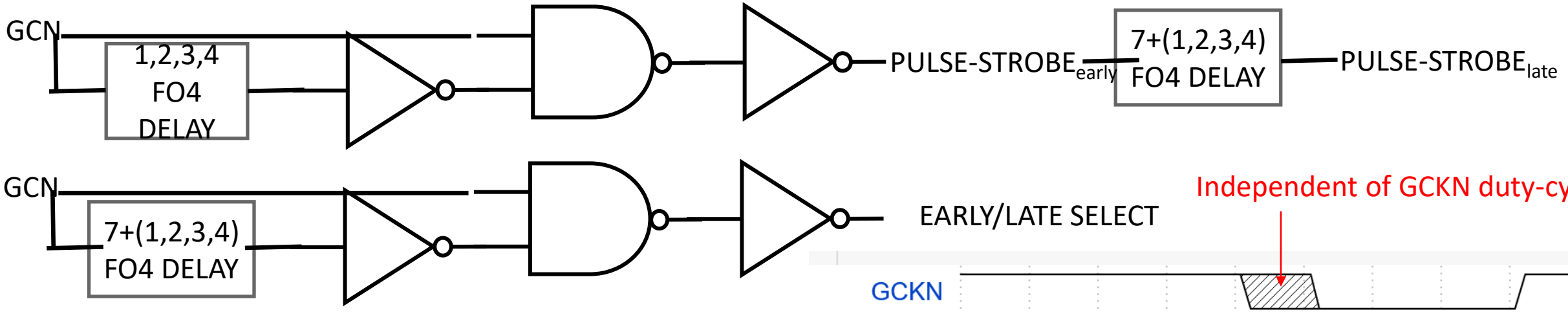
Details of double pump path

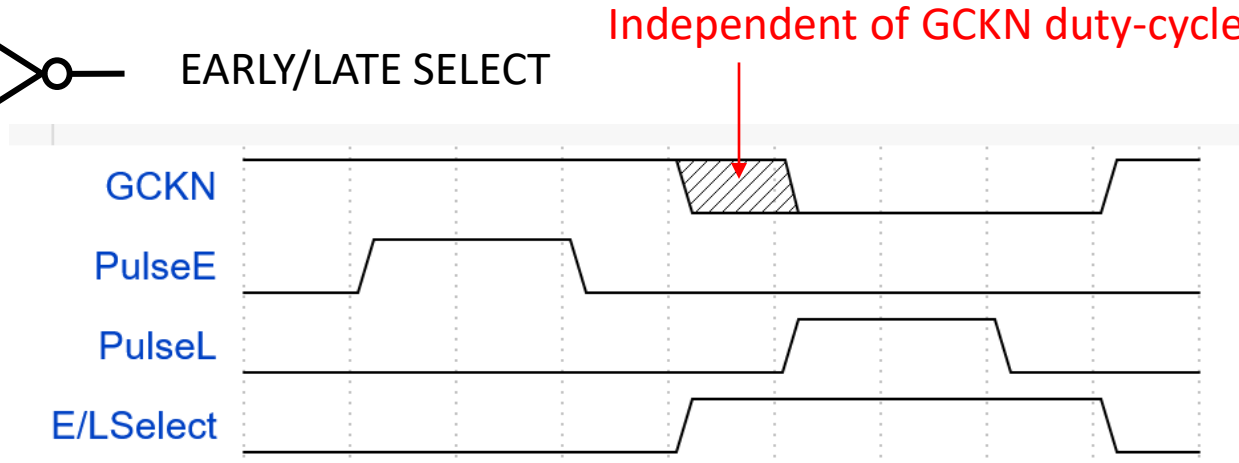
- For double pump, we duplicate in wrapper
  - For Read :Input address latches, first stage logic
  - For Write :Input address & data latches, first stage logic
- In the double pump timing loop in core:
  - For Read: predcode, decode, pulldown, LE, OR, XNAND
  - For Write: predcode, decode, Cell write.

Critical edges. 2x per cycle.

- Setup from the addresses to the clocks
  - A_0
  - A_{12}
  - A_{345}
- Minimum pulse up width
  - Write (coincident data)
  - Read (coincident pre-charge)
- Minimum pulse low width
- Double pumped path
  - Read: Pre-decode, Decode, Read pulldown, Local eval, NOR2, XNAND
  - write: Pre-decode, Decode, Cell write

Single edge Clock generator for Double Pump
Best electrical answer, all in the hard wrapper
Not verity correct, but better performance

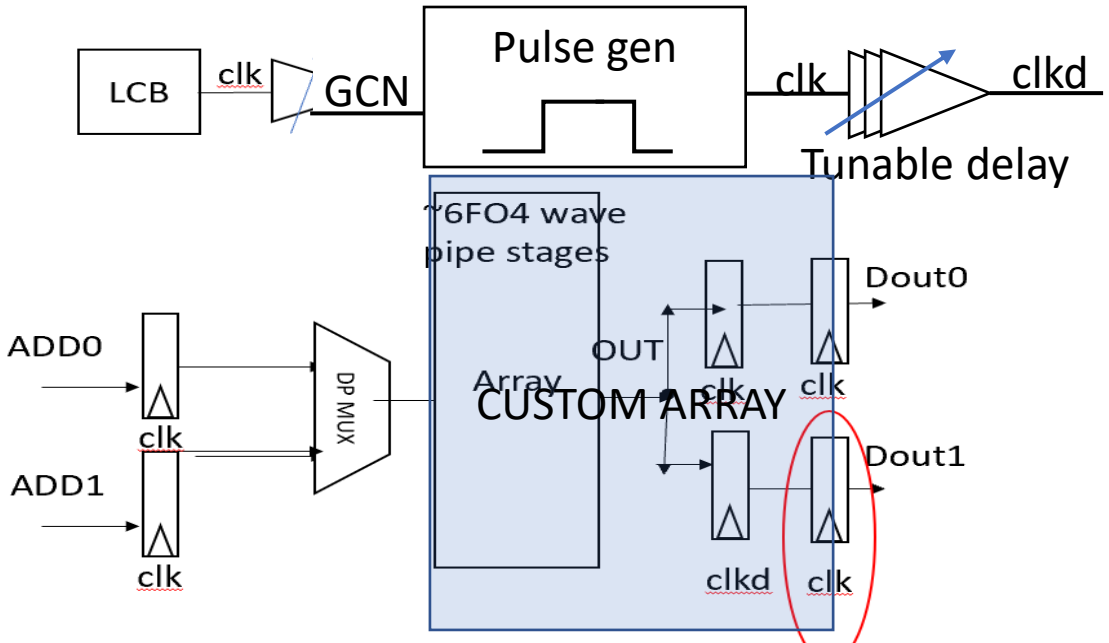




- PULSE-STROBE
  - Use PULSE STROBE$_{early}$ for XNAND$_{early}$
  - Use PULSE STROBE$_{late}$ for XNAND $_{late}$
  - OR PULSE STROBE$_{early}$ and PULSE STROBE$_{late}$ for address strobe

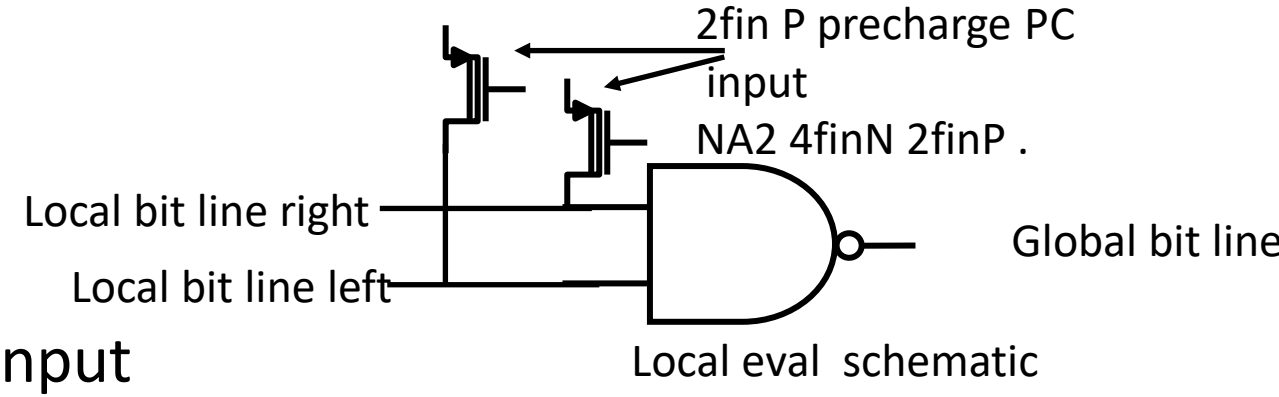
What is ”double pumped”

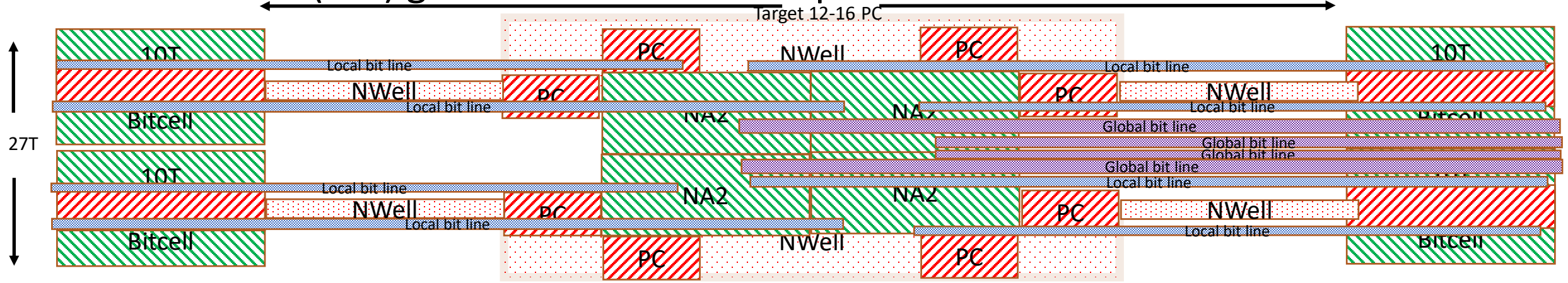
- Double pump read is sending two decode to read signals down the same logic within one processor cycle, each ending in a different XNAND
- Double pump write is sending two decode to write signals down the same logic within one processor cycle
- The shorter the path, the safer and easier it is to create timing rules. The early path finishes before the late path starts



Low level cell design - local eval migrate + shrink

- 4 x LE circuits Across 27 M3 pitches (2 cell rows = 3 9T rows)
  - the outside row is all PFET
  - This allows a wider N-well in LE
- Connect across to cell NW
  - Taps are on one edge of the total array
- Connect (M1) cell local bit line to LE input
- Connect (M3) global Bit line to LE output





16 of 10T SRAM
currently 12
5nm. 2-4-4-4 cell

6 pc x 3 (9T ckt rows Precharge gate contact @ edge
align to the 2 rows of 13.5T bit cells, connects the N-wells

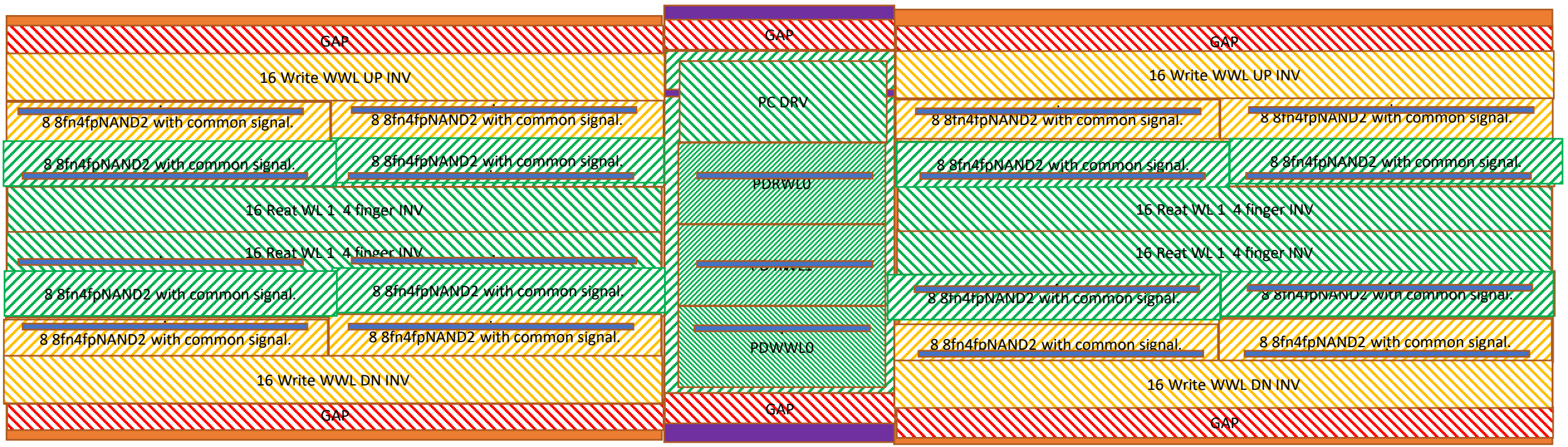
16 of 10T SRAM
currently 12
5nm. 2-4-4-4 cell

Decoder region - compressed and wired to allow 50% cell occupancy w/24 bits per sub-array with nothing above M3.

- Decoder circuits: 4pc NAND2 -8 fN 4fP & INV 16fN 16fP on 2 rows
  - The WWL decoder uses 2 of these – at top/bottom edge of the decoder
    - Balances the same load of 2x as many fins as RWL
    - Reduces M2 wires by 1 and wiring load of word lines by eliminating strap across mid
  - Final Pre-decoder uses the same arrangement
    - Pre-C012 under the Local eval (3x2 rows x 16 PC. Each side)
    - Pre-345 under the I/O spine (3x2 rows x 32PC middle)
  - The remaining 2 rows in the local eval are for PC driver
    - OR2 is created with Inverted inputs and 2 2 finger inv output
  - The remaining 2 rows in the center region are for buffering XNAND clocks

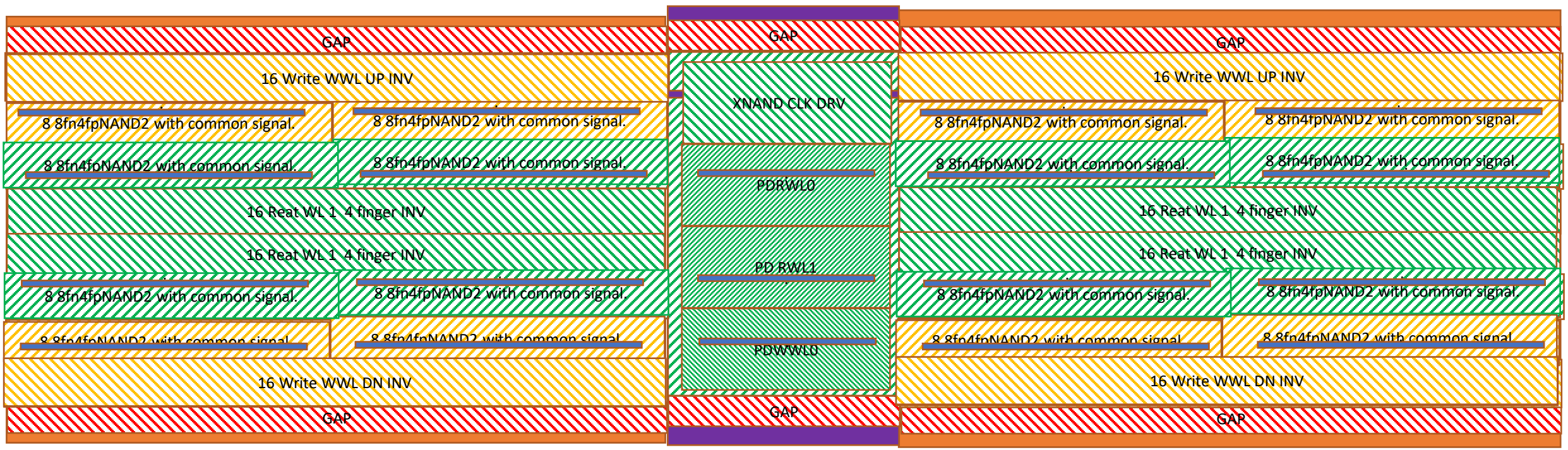
RIB – decoder (the WWLD is dupl)+Pre-dec 16PC under LE

- In the bit cell region
  - 8 rows of the region are decode (NA2+4 finger INV). With the WWL duplicated for fanout
- In the bit cell region under the local eval
  - 6 rows of the region under the LE are Pre-decode (NA2+4 finger INV)
  - 2 rows of the region under the LE are PC drv (NA2+2 *2 finger INV)



Center region is for the bit_{345} decoders + clock buf

- 6 rows of the center region are the last Pre-decode (NA2+4 finger INV)
  - They take 32 PC each
- 2 rows of the center region are for XNAND clk driver



M2 Wiring in decoder region and over bitcell

- Use a separate RWLD for the top half and the bottom half of the Write word line, and do not connect the two. This frees up M2 wires for wiring and powering the decoders
- Connect the 8 consecutive connects to the decoders on an M1 track that is free on either the N or P side of the 4-finger balanced NAND2 (8N 4P) Gate. Thus, we only need 1 M2 for input
- Total in decoder region out of 6M2 per 4 PC.
  - RWL0+RWL1 - 2
  - NAND INPUT - 1
  - VDD or GND 1
  - Strap and connect from NAND to INV 1. (WWL is outside this region)

Wiring across the decoder region

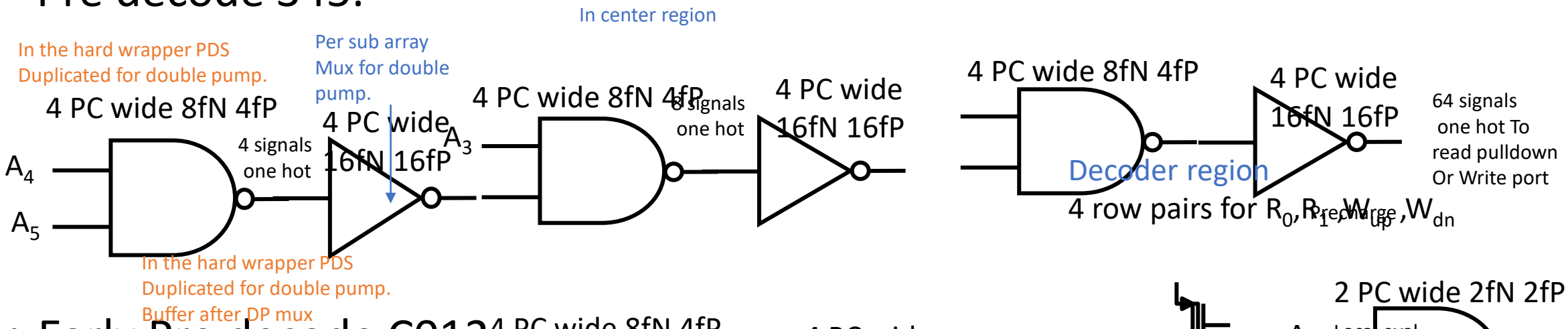
- From hard wrapper
  - High fanout on low resistance (D)
    - XN clocks (4)
    - C * $A_{0\,,}$ C *~A_0 (6), $A_{3\,,\,\sim}A_3$ (6)
  - Low fanout on M3
    - $PD_{2,3}$, $PD_{4,5}$, (12*2)
- From Pre decode
  - Center - High fanout on low resistance (D)
    - Over bit region: $R_{0,1}$ $PD_{0\text{-}7}$ (16), Over Decoder region: W $PD_{0\text{-}7\,,}$ (8)
  - Low fanout on M3
    - M3 branch for: W,$R_{0,1}$ $UpperPreDecode_{0\text{-}7\,,}$ (24). (from center region)
    - M3 branch for: W,$R_{0,1}$ $LowerPreDecode_{0\text{-}7\,,}$ (6) (from under the Local eval)

Bit line region 2 of these per 3 Cell rows

- Each group o f 3 circuit rows supports 2 bits
  - 2 NOR2 (5 fing) + 4 XNAND (48 fing), 2 output (6 * 2 fing), 2 BLDRV (6*2 fing)
  - 82PC/3 rows → 28 PC
- The border rows between the I/O and the center will have 2 14 PC groups that can be used for any remaining EC.
- The center consists of 8 rows is about of 32PC
  - 6 will contain the 345 pre-decodes for both read & Write.
  - 2 will contain the clock buffers for the XNANDS

Read path 2, Write path 1 (w/dup decoder) Each circuit has a delay of about 1FO4.

- Pre decode 345.



- Early Pre decode C012