



Final Demo

Group 7: Hsin-Ling Lu, Hsiang-Yang Fan, Daniel Yu, Yong-Xin Huang, Yu-Ting Shiau

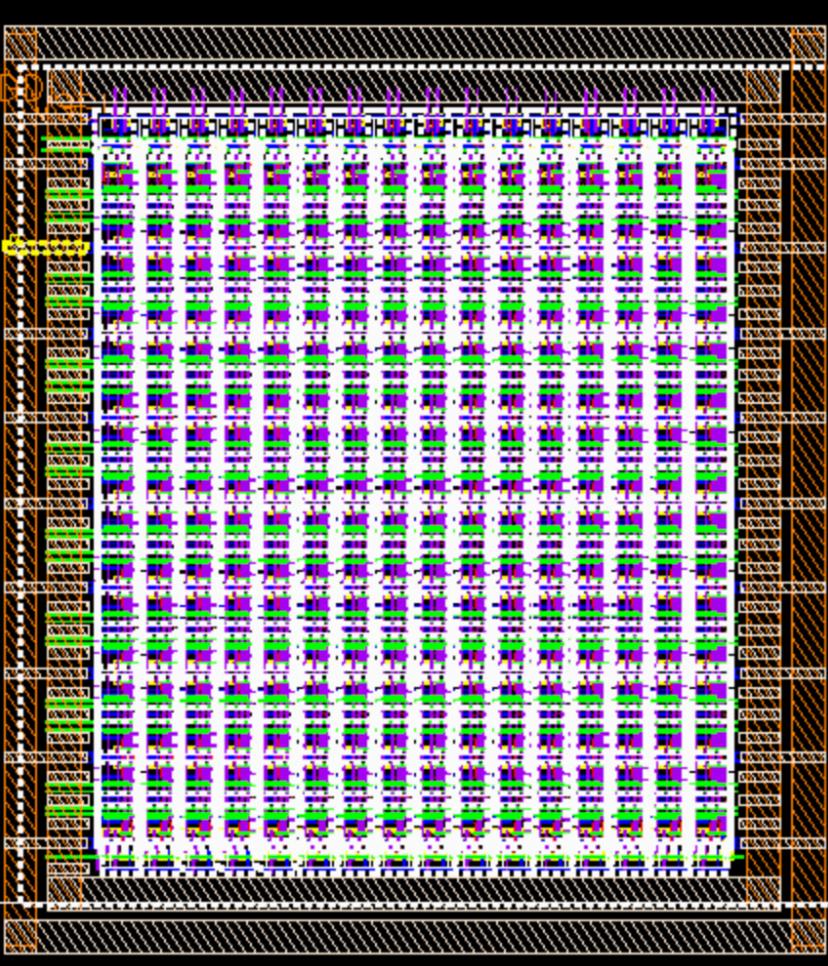
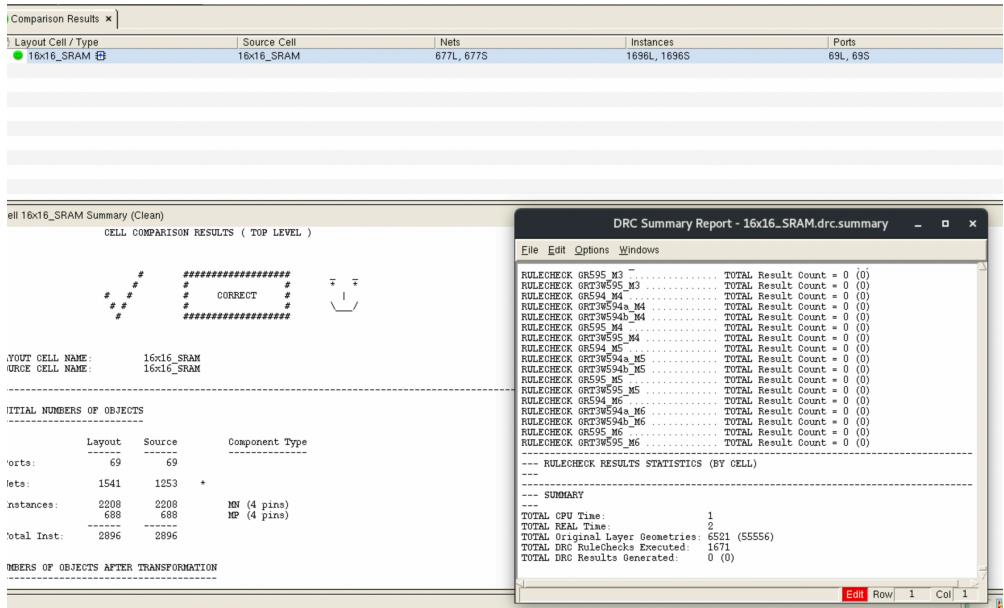
Outline

- I. Full Processor Layout with our Custom Block
- II. Baseline Processor
- III. Instruction for the Custom Block

I-a Custom Block- DRC & LVS

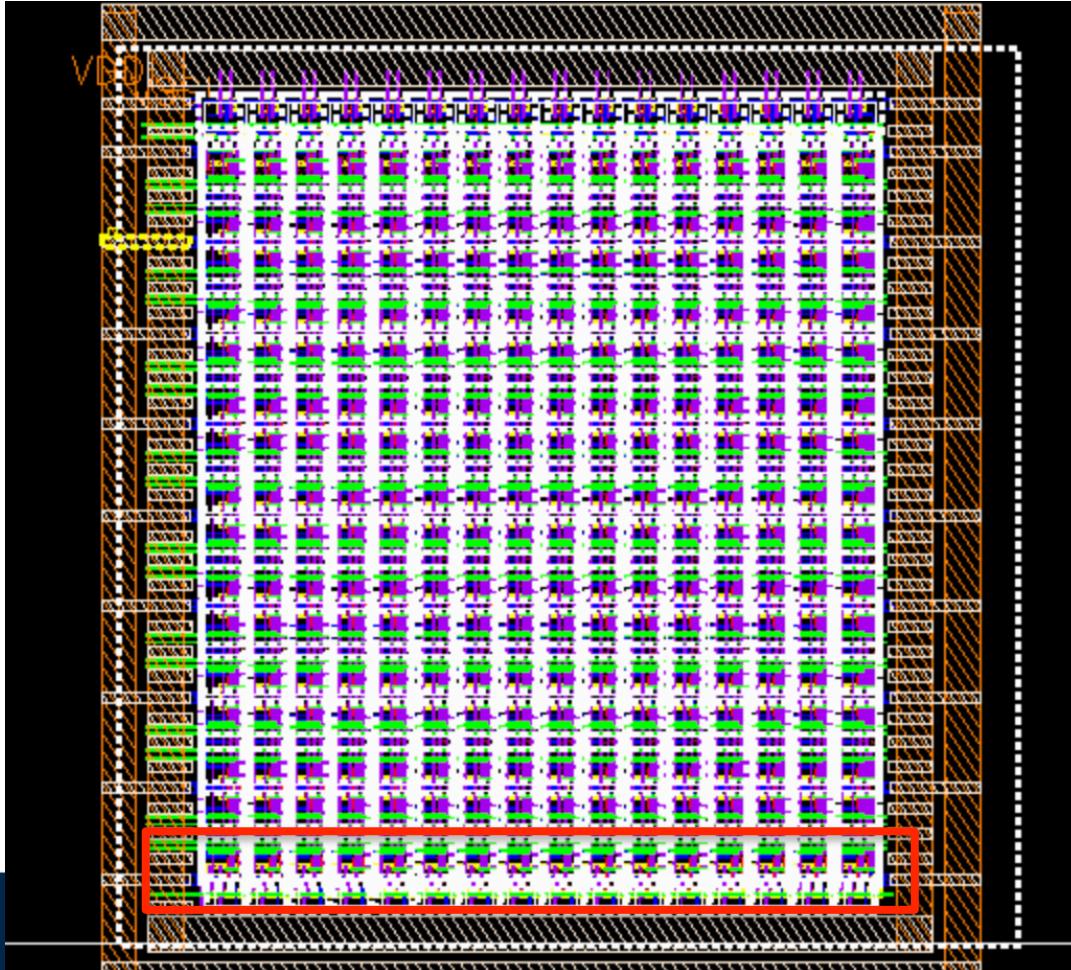
I-b Custom Block- Layout

- Pass in DRC & LVS
- Demo: Layout, DRC & LVS



I-b Custom Block- Layout

Zoom in



I-c. Processor with Pads - DRC & LVS

```
#####
##      CALIBRE      SYSTEM      ##
##      L V S      R E P O R T      ##
##      #####
REPORT FILE NAME:          eecs427_top.lvs_report
LAYOUT NAME:              /afs/u mich.edu/class/eecs427/w23/group7/FINAL/Calibre/LVS/eecs427_top.sp ('eecs427_top')
SOURCE NAME:               /afs/u mich.edu/class/eecs427/w23/group7/FINAL/Calibre/LVS/eecs427_top.src.net ('eecs427_top')
RULE FILE:                /afs/u mich.edu/class/eecs427/w23/group7/FINAL/Calibre/LVS/_cmrf8sf.lvs.cal_
CREATION TIME:             Wed Apr 19 04:17:22 2023
CURRENT DIRECTORY:         /afs/u mich.edu/class/eecs427/w23/group7/FINAL/Calibre/LVS
USER NAME:                seanfan
CALIBRE VERSION:          v2021.3_15.9   Tue Jul 6 13:44:37 PDT 2021

OVERALL COMPARISON RESULTS

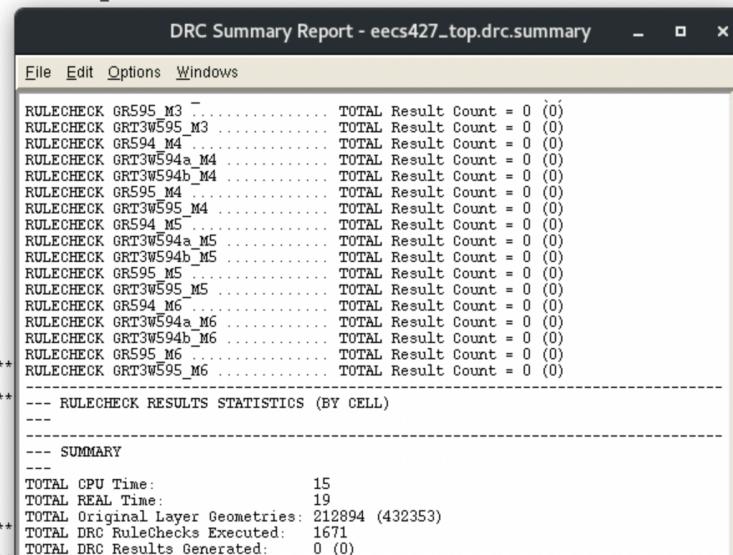
#      #####      #
#      #      #      CORRECT      #      *      *
#      #      #      #      |      #
#      #####      #

Warning: Components with non-identical power or ground pins.
Warning: Ambiguity points were found and resolved arbitrarily.

***** CELL SUMMARY *****

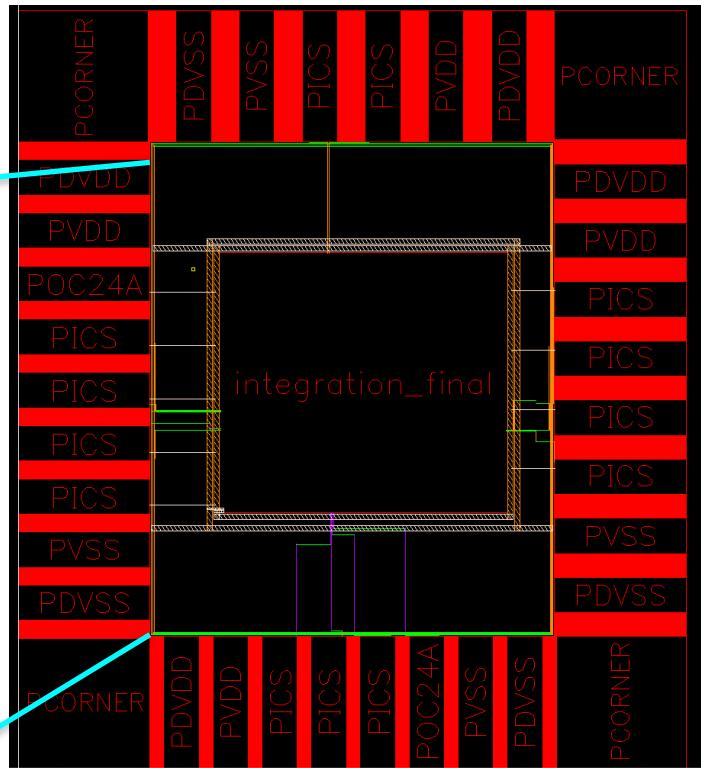
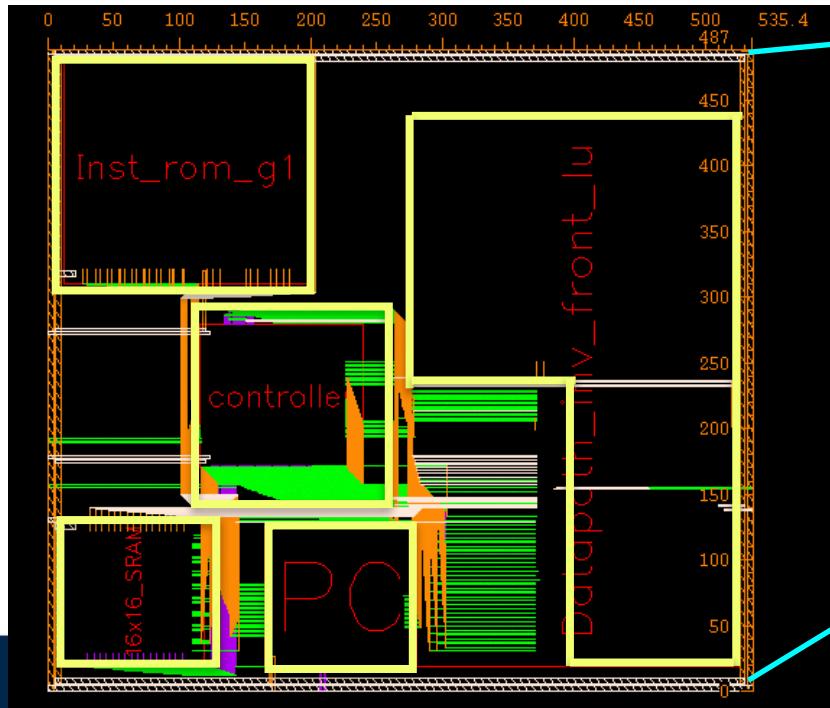
Result      Layout      Source
-----  -----
CORRECT    eecs427_top    eecs427_top

***** LVS PARAMETERS *****
```



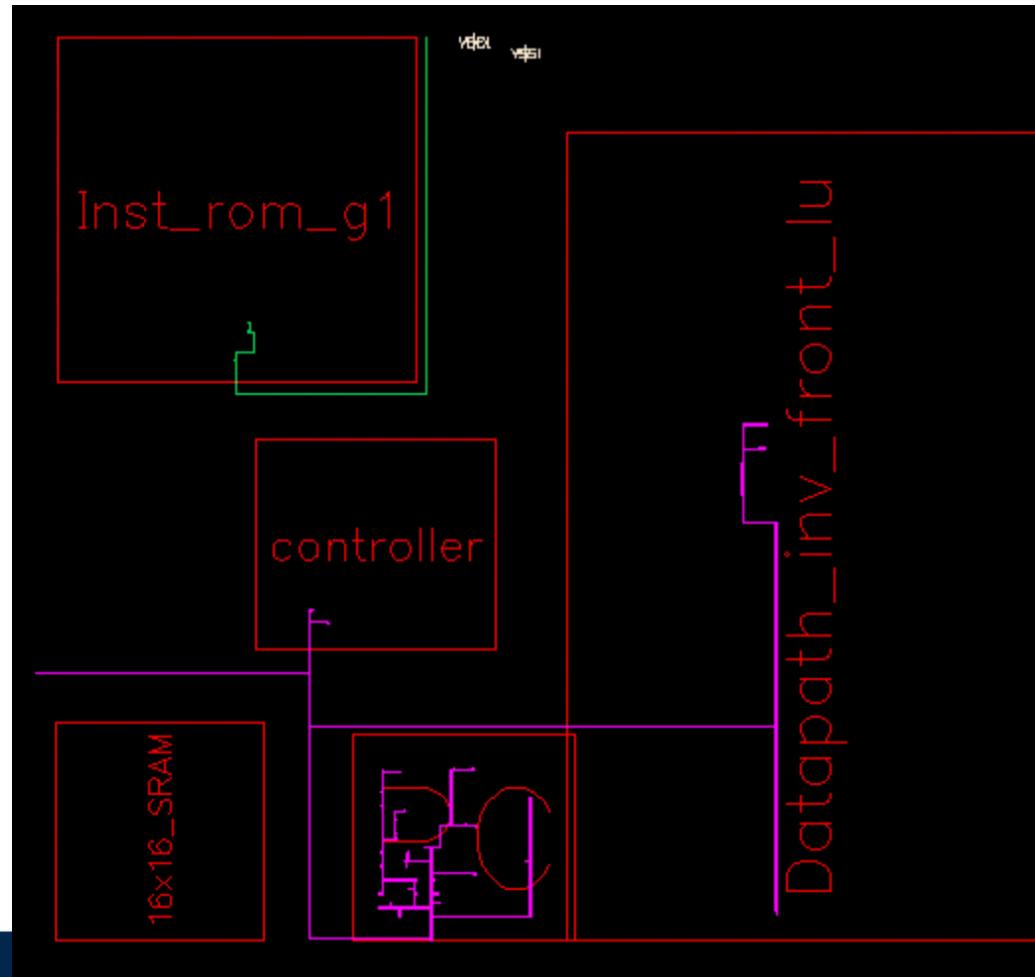
I-d. Processor with Pads - Layout & Core Density

- Total area = 260739.8um²
- Core density = 0.56
- Yellow outline show the total area of the cells



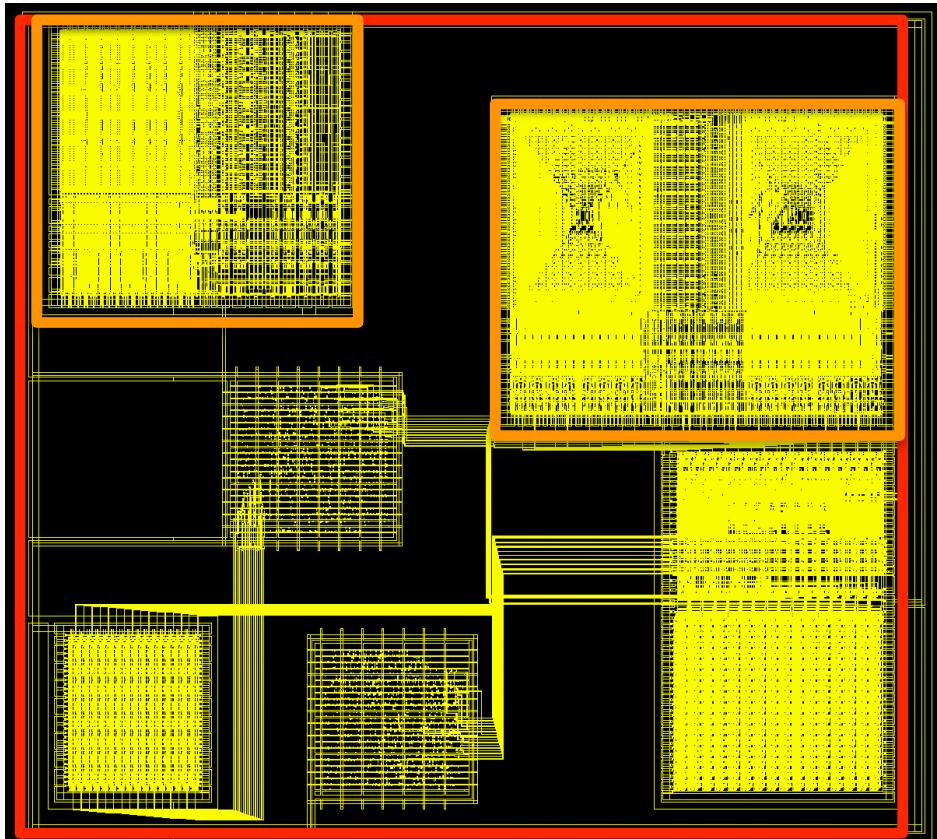
I-e. Clock distribution

- Pink route: CLK
- Green route: CLK_b
- Distribution considerations: **Improve performance**
 - Positive skew
 - CLK routes through PC and controller then Datapath



I-f. Power routing/rings

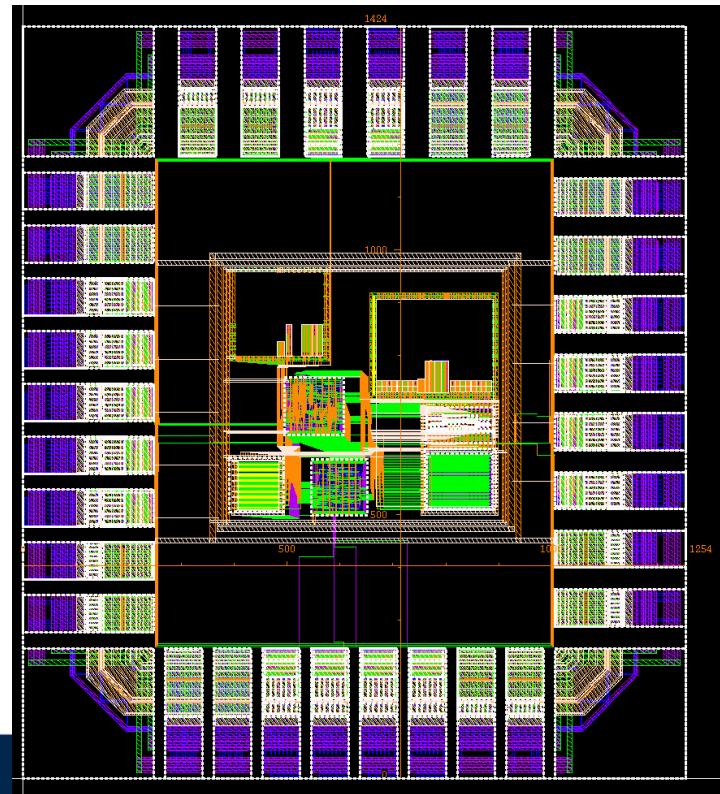
- Red: Power ring
- Orange: Block ring of IMEM and DMEM
- Yellow: Local power outing



I-g. Total area

I-h. Show the number of off-chip inputs and the pads for the processor

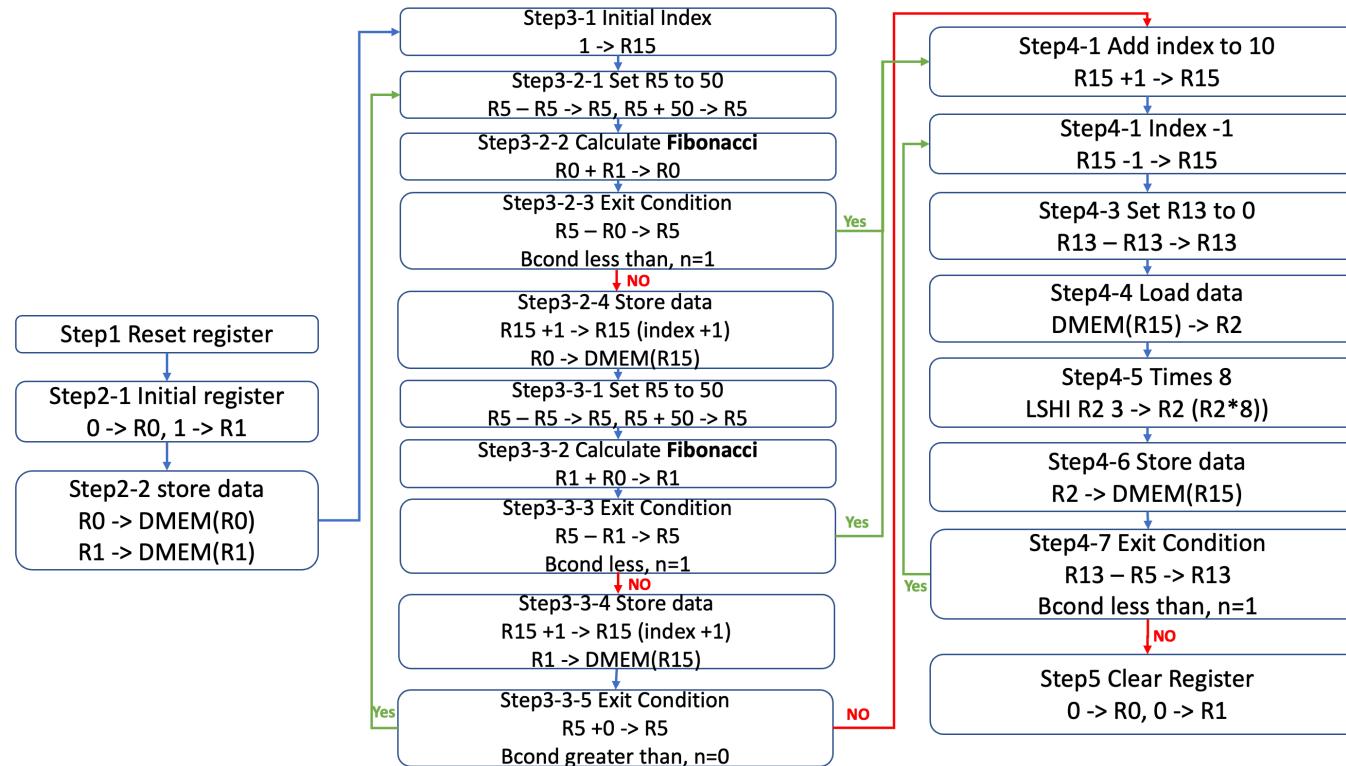
- Total area (with pads) = $1424 \times 1254 \mu\text{m}^2$
- Total area (without pads) = $487 \times 535.4 \mu\text{m}^2$
- Total inputs:
 - CLK, CLK_b,
 - RST, RST_PC, CEN, eight,
 - scan_en1, scan_en2,
 - scan_in1, scan_in2,
- Total outputs:
 - scan_out1, scan_out2



II-a. Show the assembly code program

II-b. How we loop the calculation for Fibonacci Numbers and exit

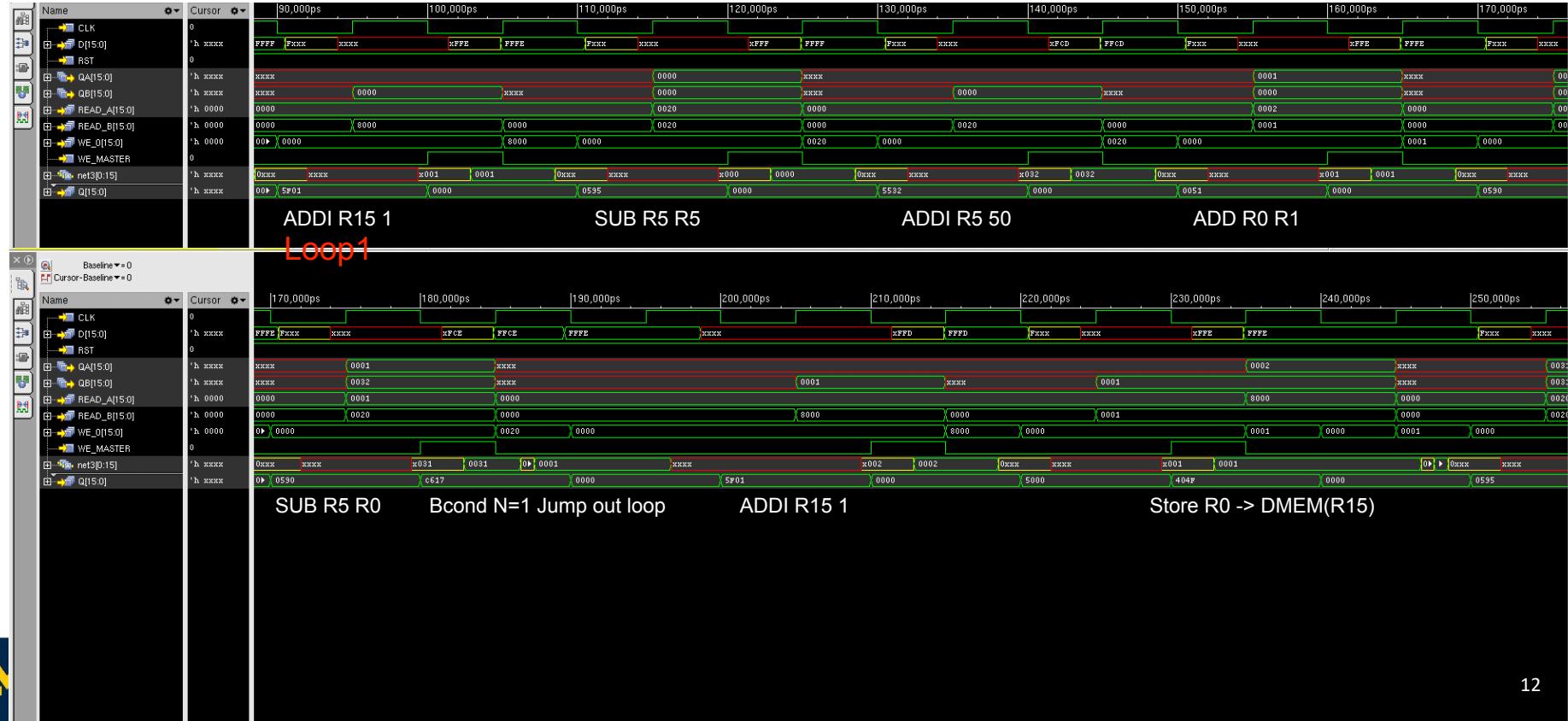
II-c. Explain one cycle of the calculation, including the load and store to RegFile



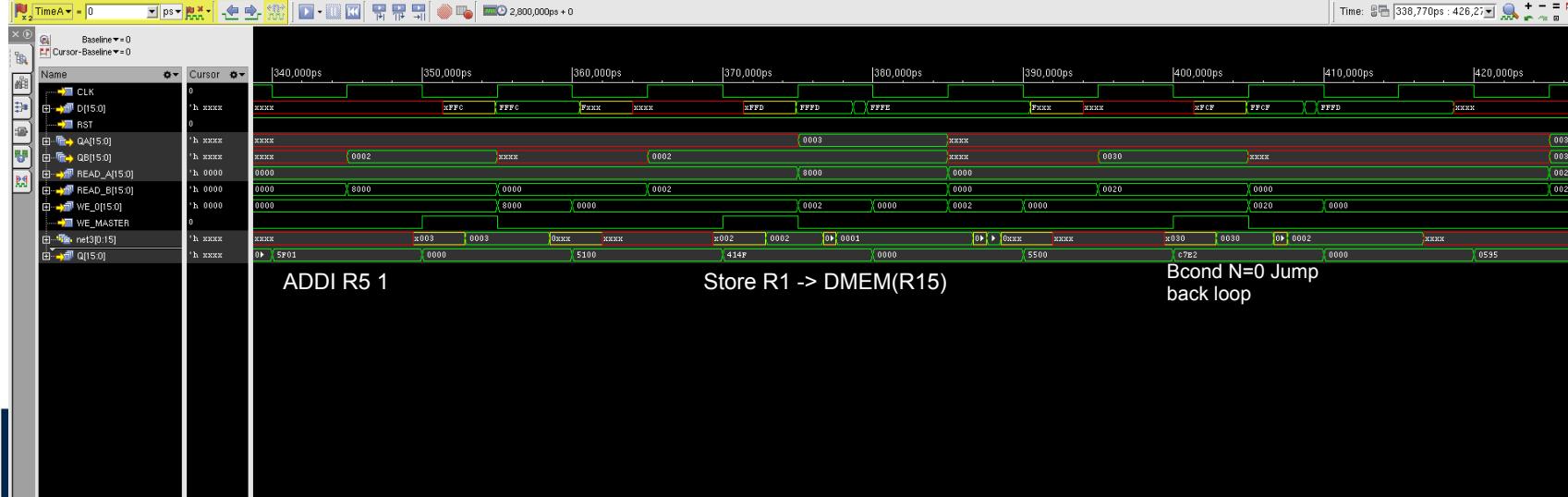
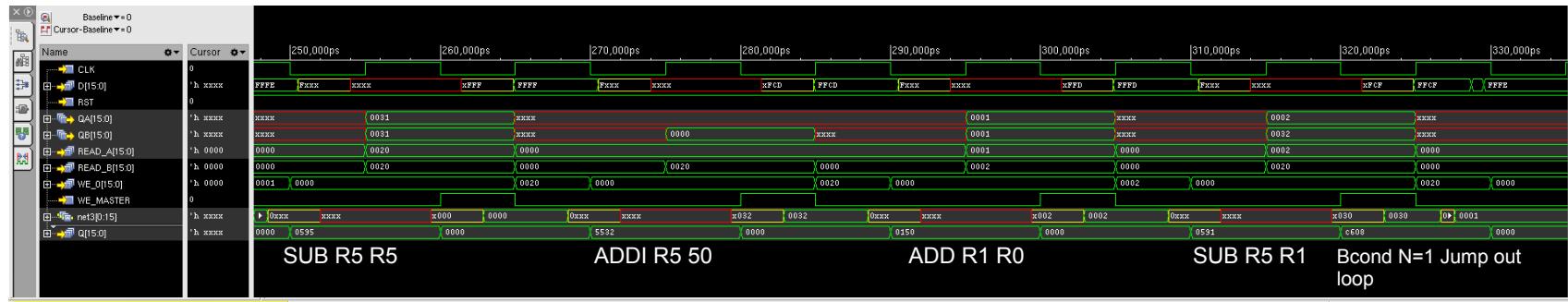
Fibonacci Numbers simulation



Fibonacci Numbers simulation



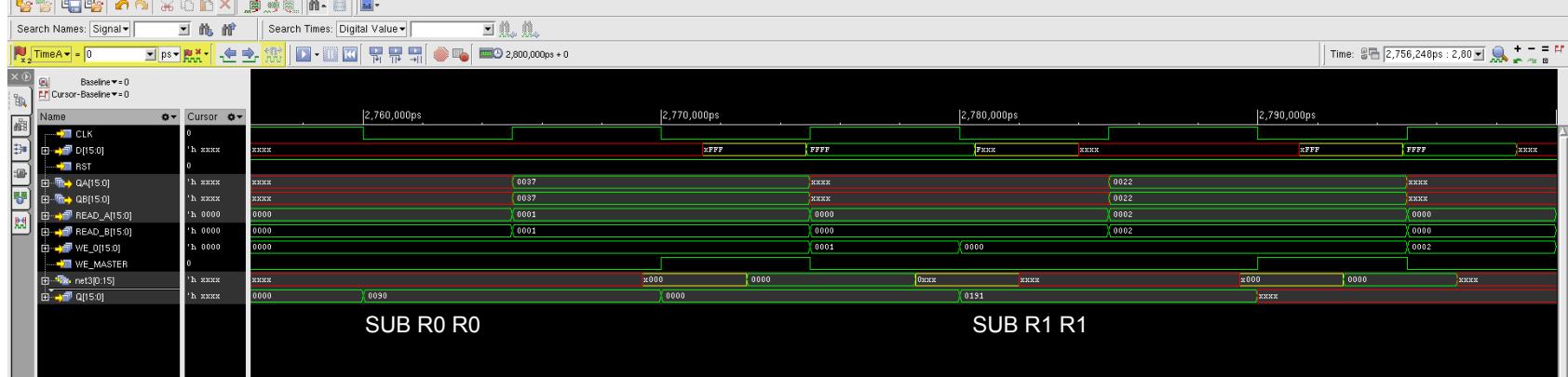
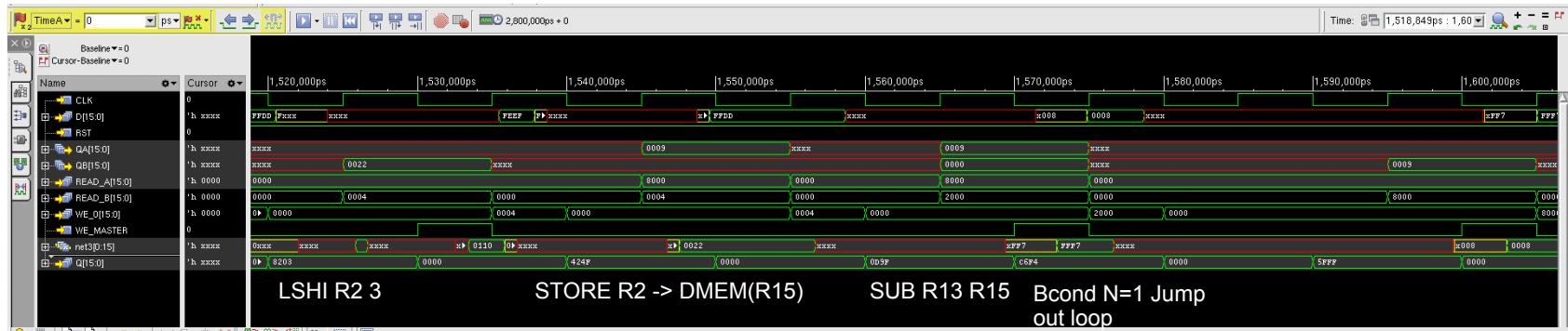
Fibonacci Numbers simulation



Fibonacci Numbers simulation



Fibonacci Numbers simulation



II-d. Report the max operating frequency for your baseline processor

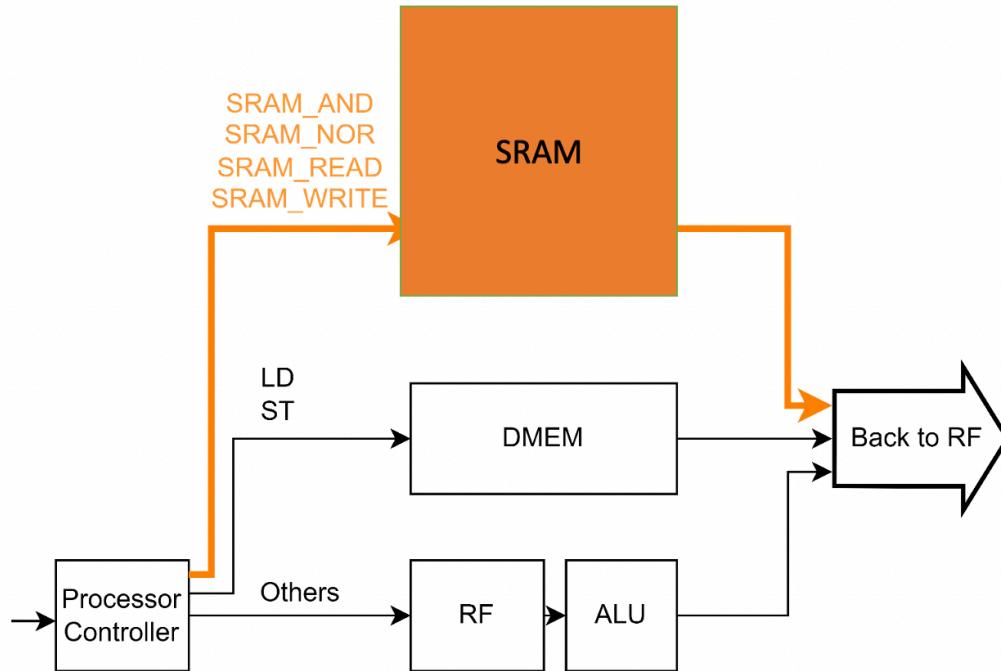
- Max operating frequency: $1/(10\text{ns}) = 100 \text{ MHz}$

III-a. Function of our custom block

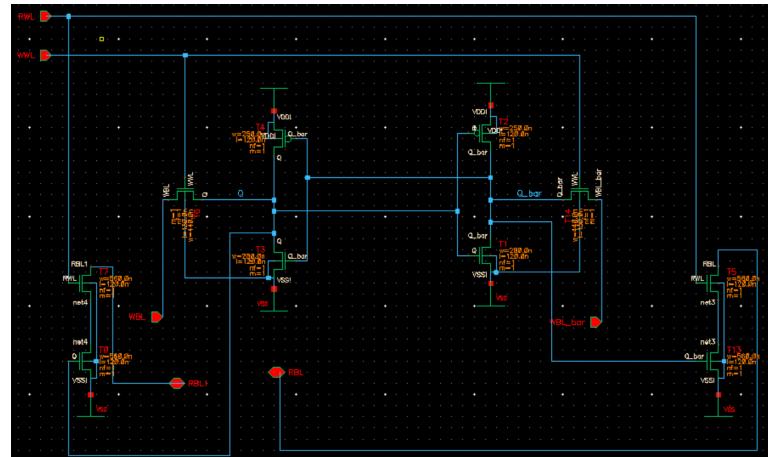
- Support 4 additional instructions for SRAM
 - SRAM_AND or SRAM_NOR
 - Execute bitwise Boolean operations in 2 SRAM rows and store the result in one of them
 - SRAM_LOAD or SRAM_STORE
 - Data transfer between RegFile and SRAM

| Mnemonic | 15~12 (OP Code) | 11~8 | 7~4 (OP Code Ext) | 3~0 |
|-------------------|-----------------|-----------|-------------------|-----------|
| SRAM_AND | 1010 | SRAM_addr | 0100 | SRAM_addr |
| SRAM_NOR | 1010 | SRAM_addr | 1000 | SRAM_addr |
| SRAM_LOAD | 1010 | Rdest | 1100 | SRAM_addr |
| SRAM_STORE | 1010 | SRAM_addr | 1111 | Rsrc |

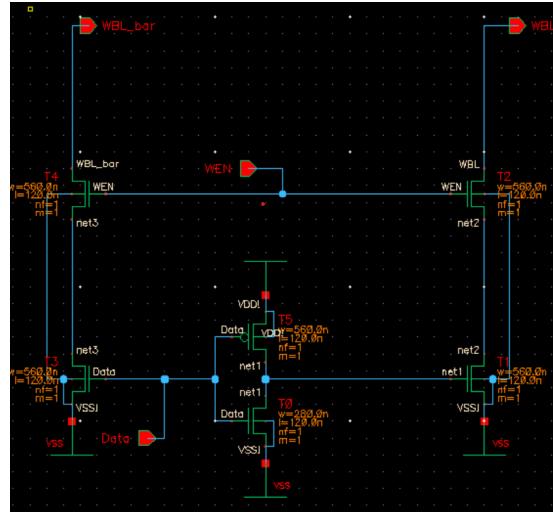
III-b. How we integrate the custom block into baseline



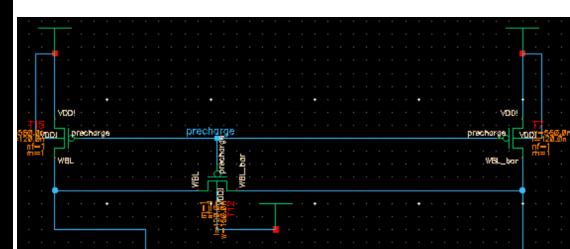
III-c. Design Considerations (Sizing)



10T differential read-decoupled cell



Discharge

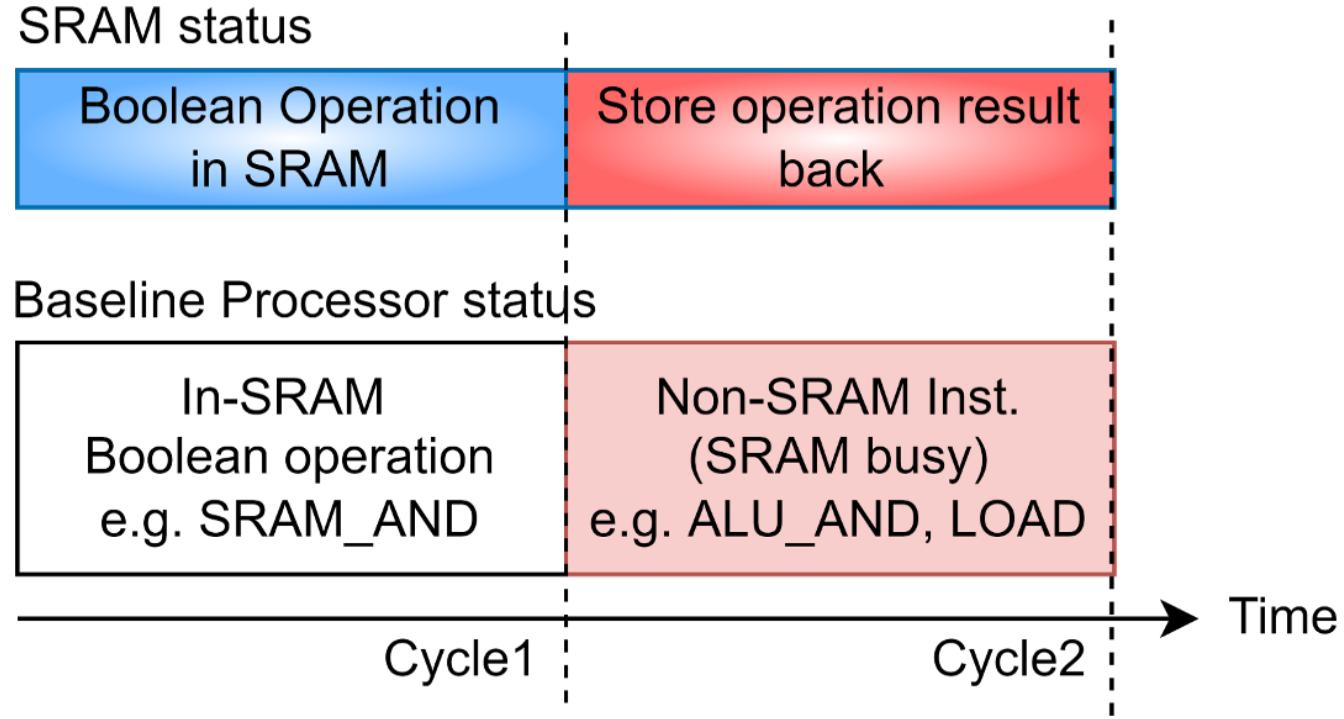


Precharge

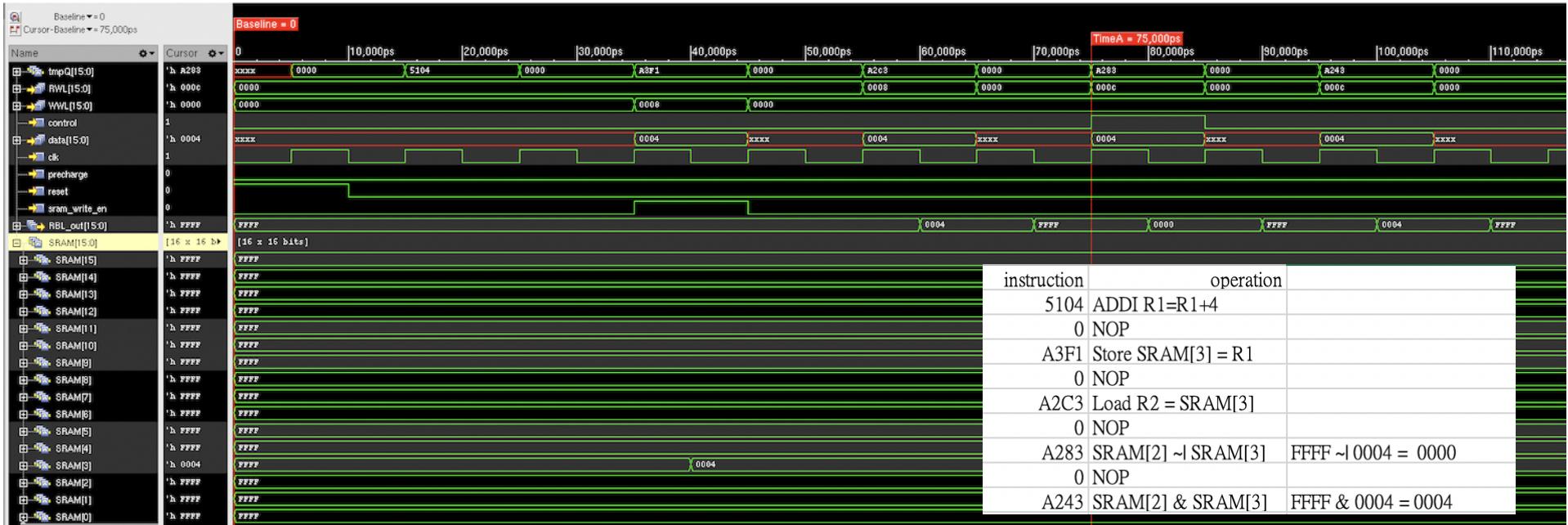
Boolean operation in SRAM

1. Controller will activate 2 read word lines & control signal
2. AND/NOR
 - Read bitline precharge high
 - Activate 2 read wordline
 - Control signal will choose to operate AND/NOR

III-c. Design Considerations (Cycles)



III-d. Show at least one operation for your custom block with baseline processor



SRAM_store

SRAM_Load

SRAM_NOR

SRAM_AND

Feedback

1. Q: No information for sense amplifier and level converter?

A: We add buffer in the bottom of our bitlines. It can strongly pull high or low in our circuits.

2. Q: In the full simulation results, data is available every cycle and there are XXXX in between. What's the reason?

A: We add NOP instruction between each instruction. Therefore, there are several Xs in our graph. The data is available every cycle, however, we have control signal sram_write_enable to control what time the data can be stored.

3. Q: Don't you need an XOR of the BLs to get the answer?

A: Our SRAM support 4 additional function. SRAM_AND, SRAM_NOR, SRAM_READ and SRAM_WRITE. When the controller receives the SRAM_AND or SRAM_NOR instruction, the controller will activate 2 addresses(two one hot addresses doing an OR). After receiving the address, the SRAM will then read the bitline out. Therefore, the controller will choose to activate 1 or 2 read word line. We don't need XOR to determine whether it is 1 or 2 word line that has been activated.