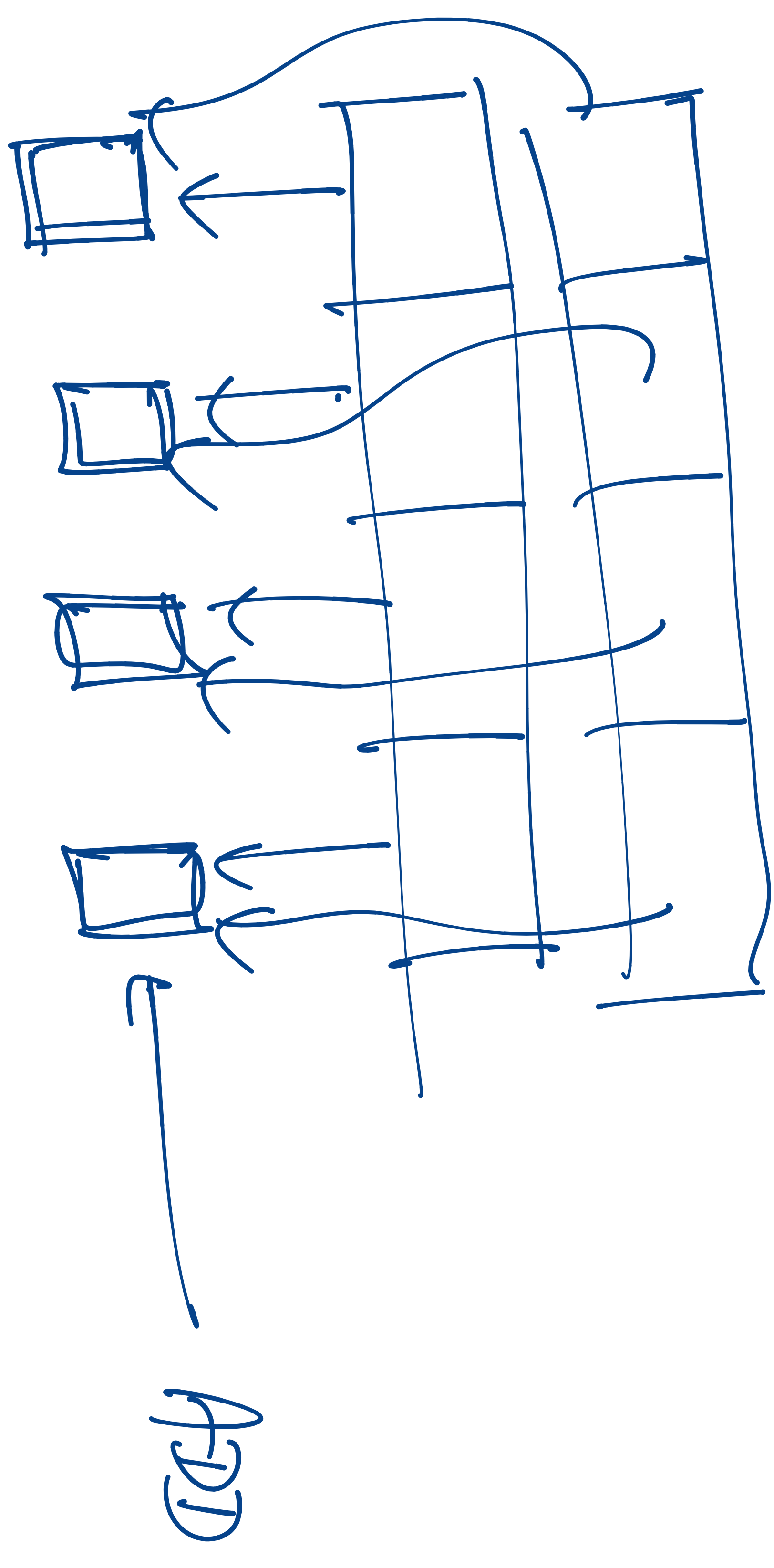
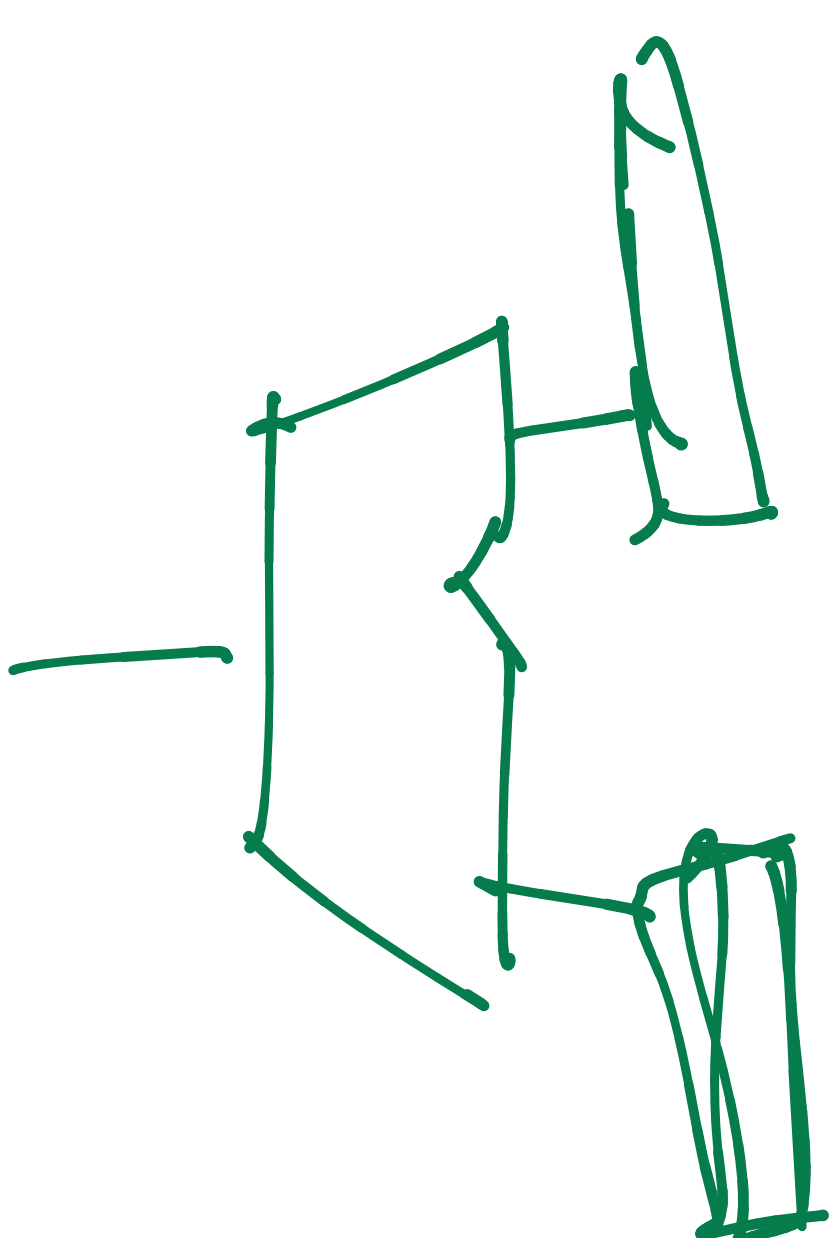
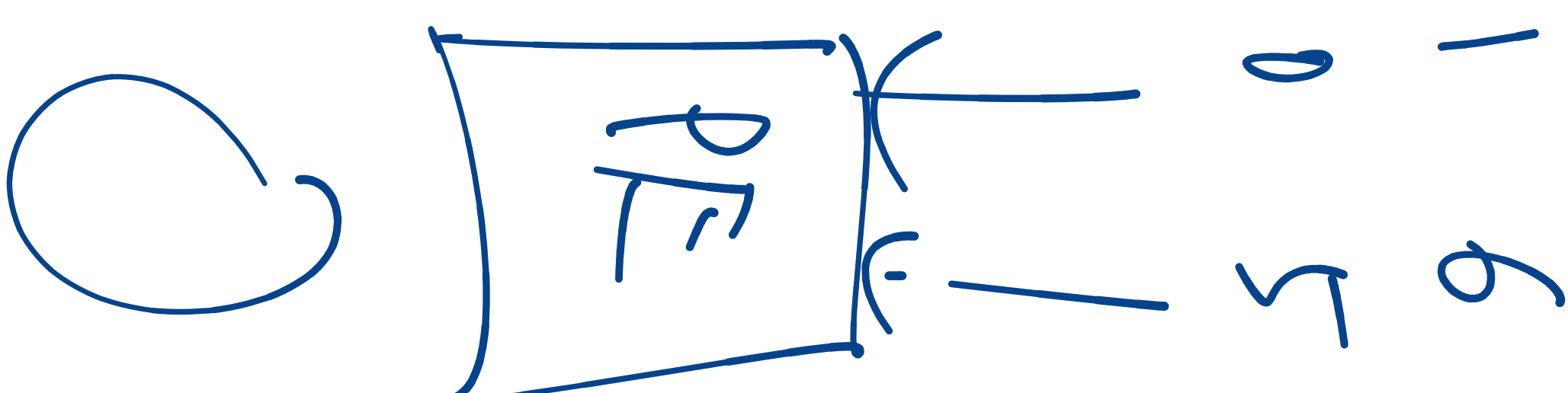
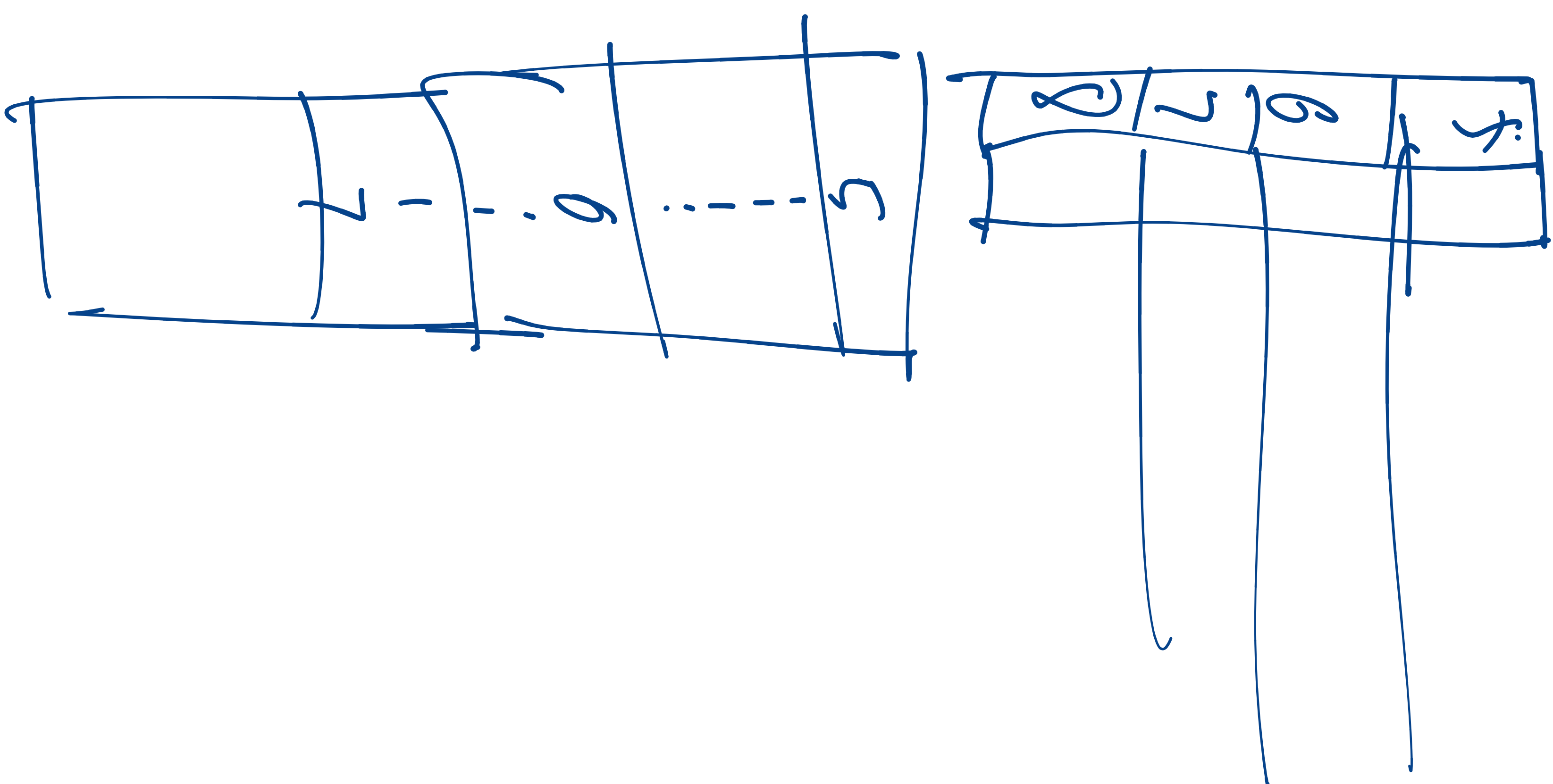
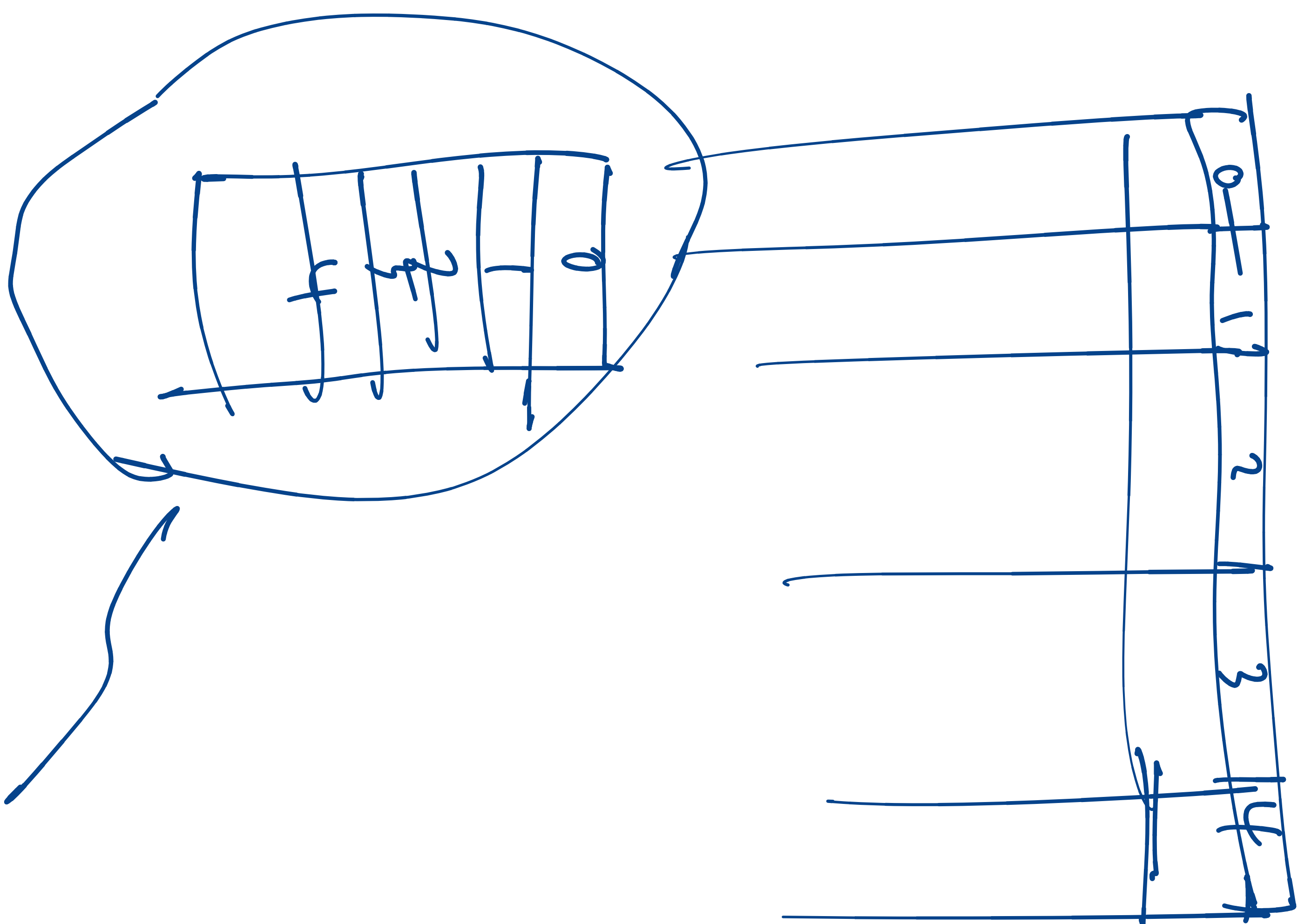


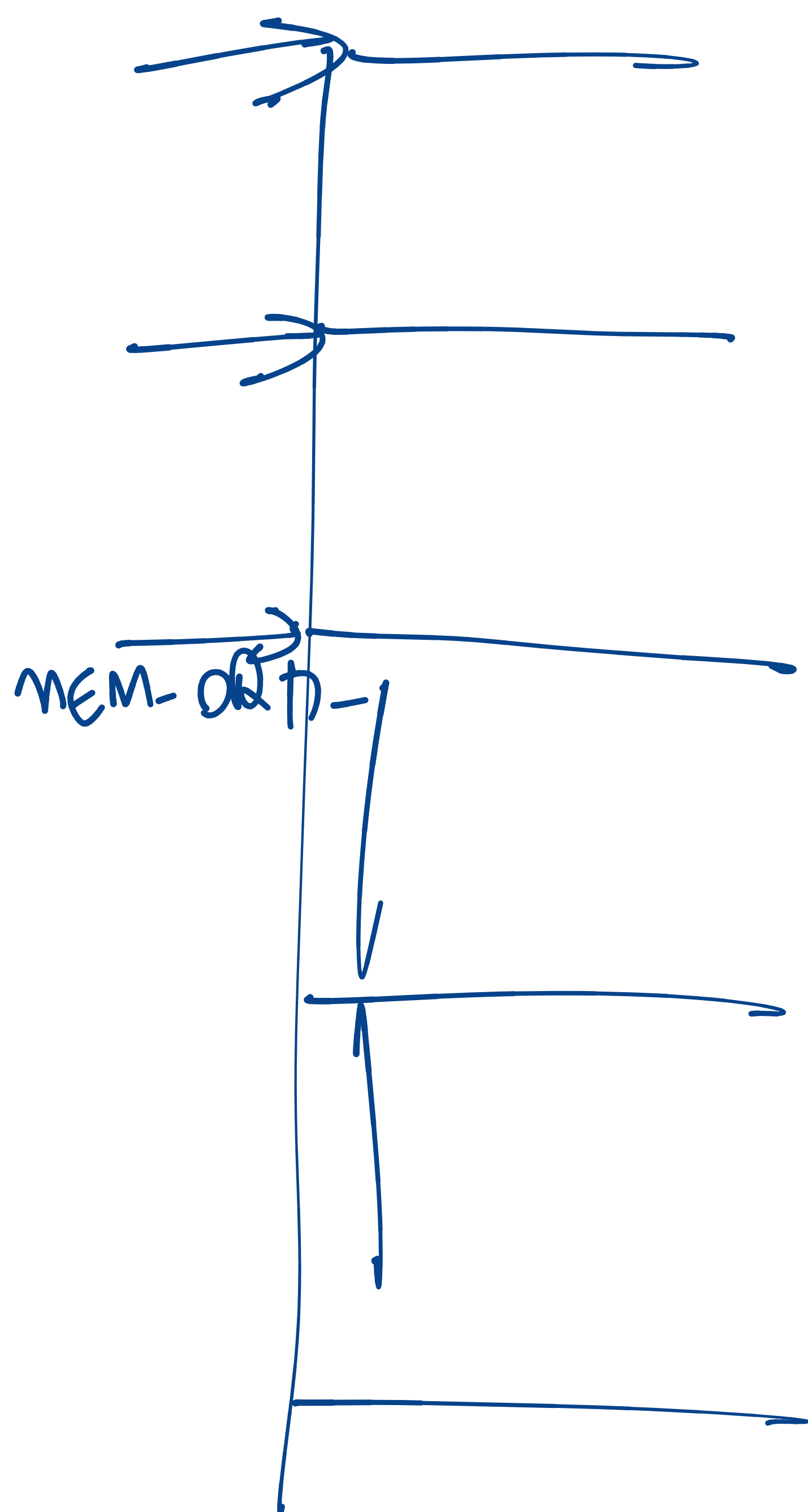
52MD



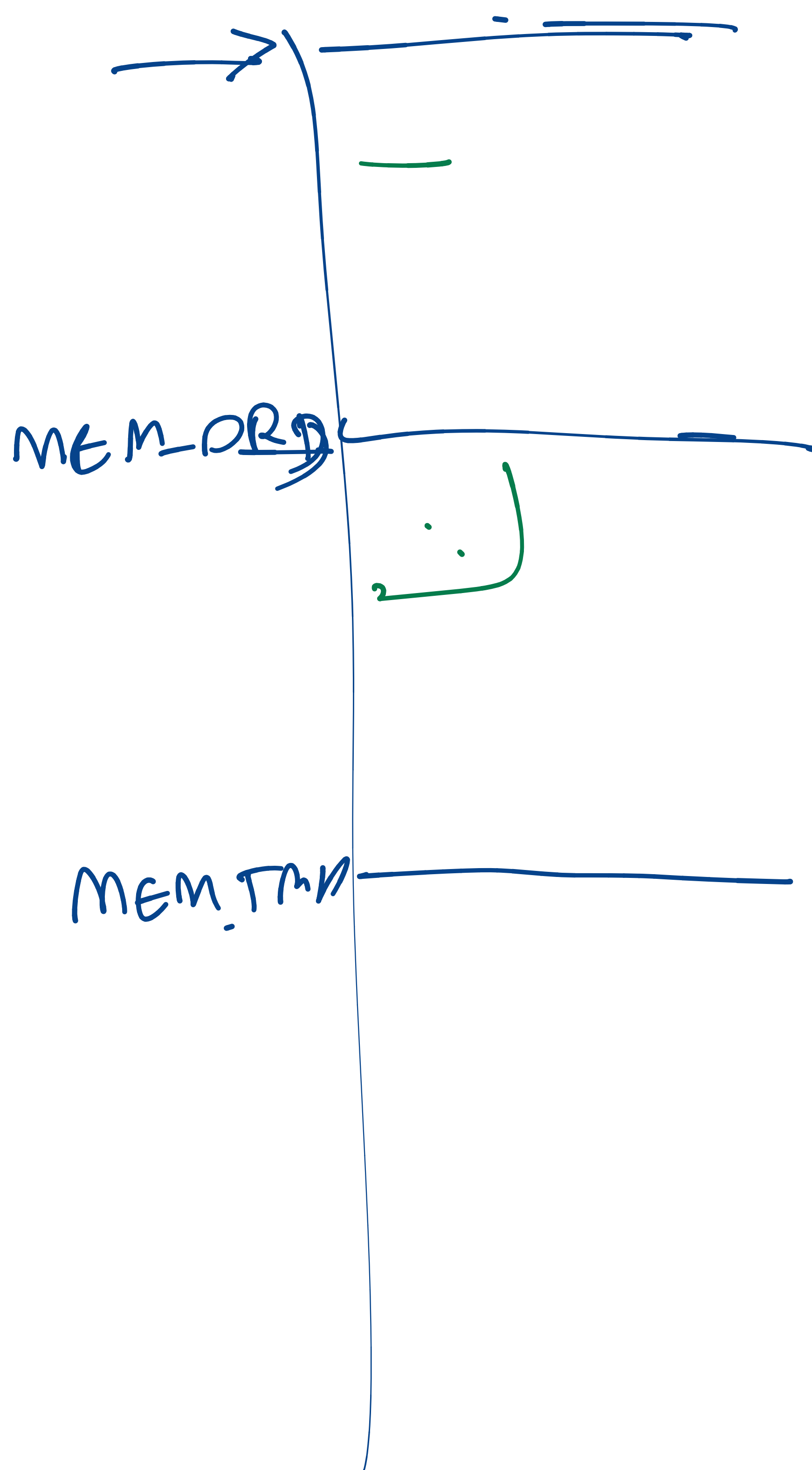
1



mem8

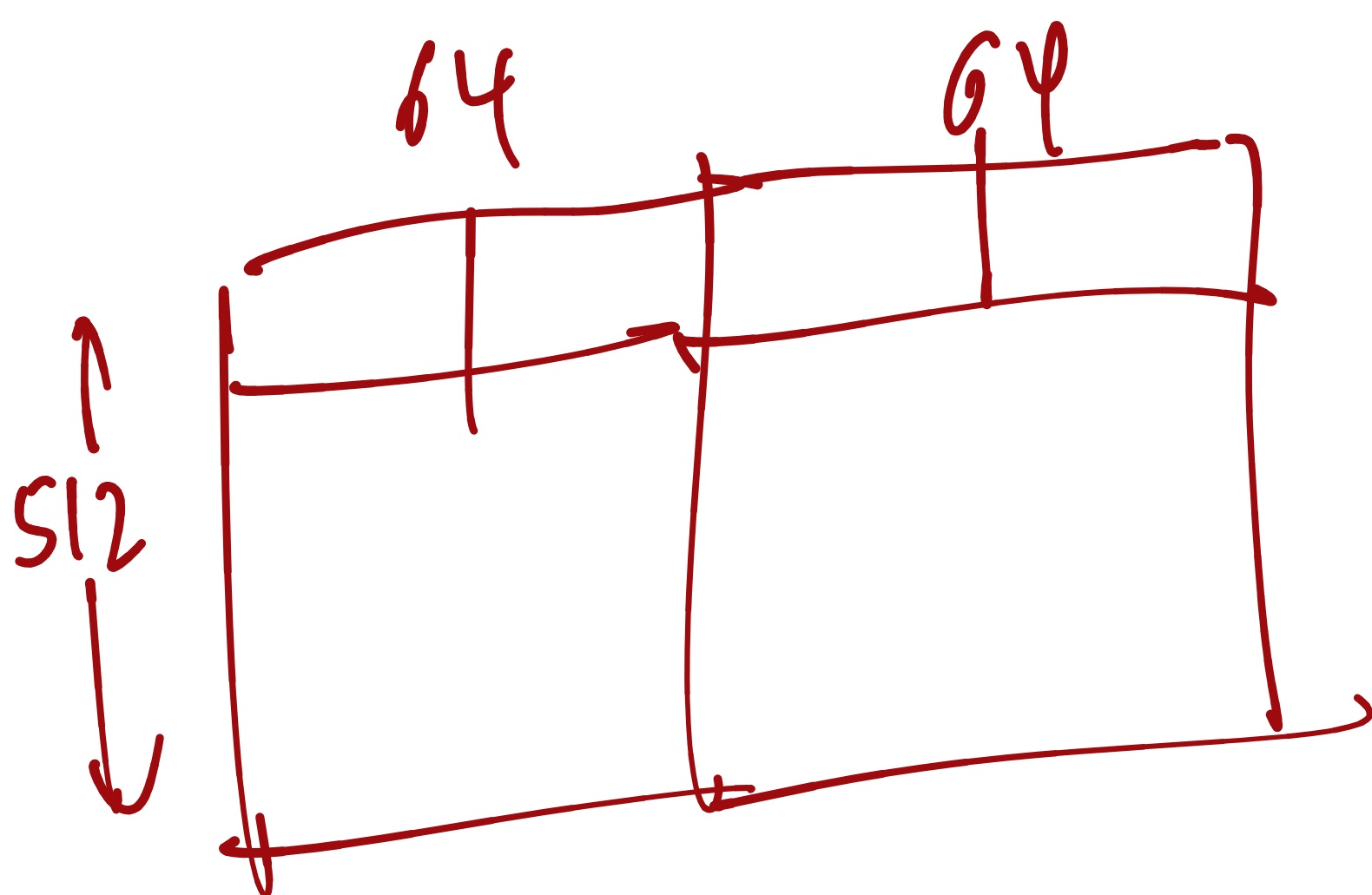
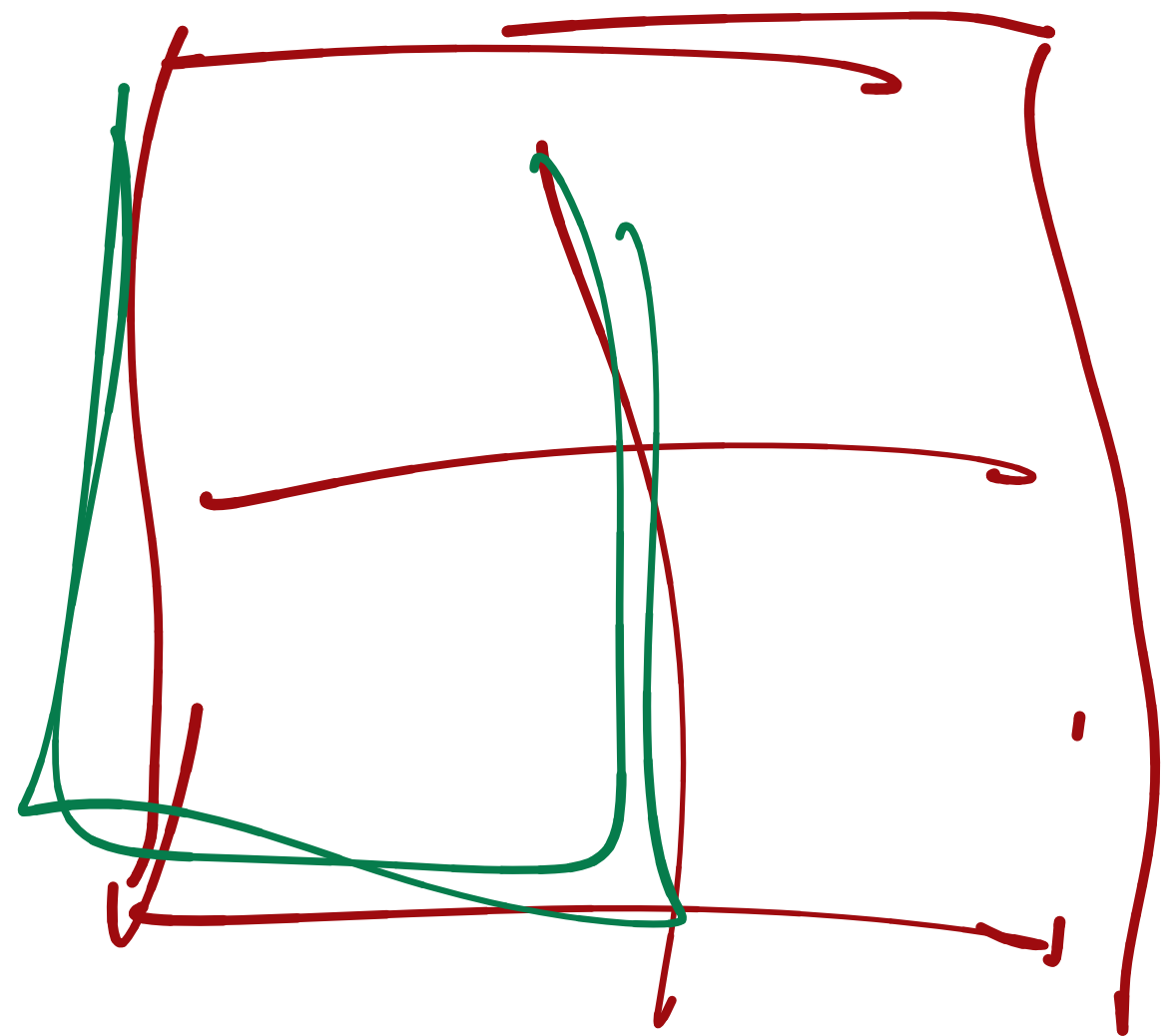
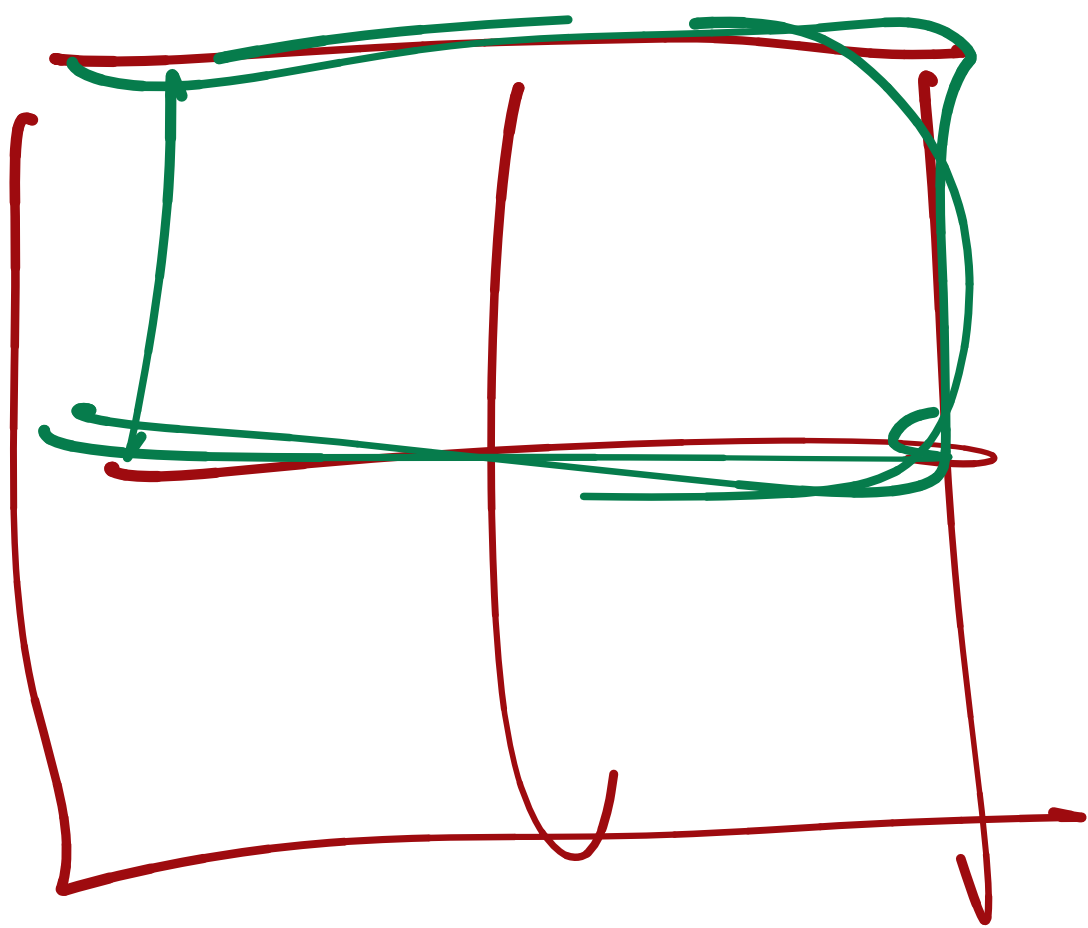
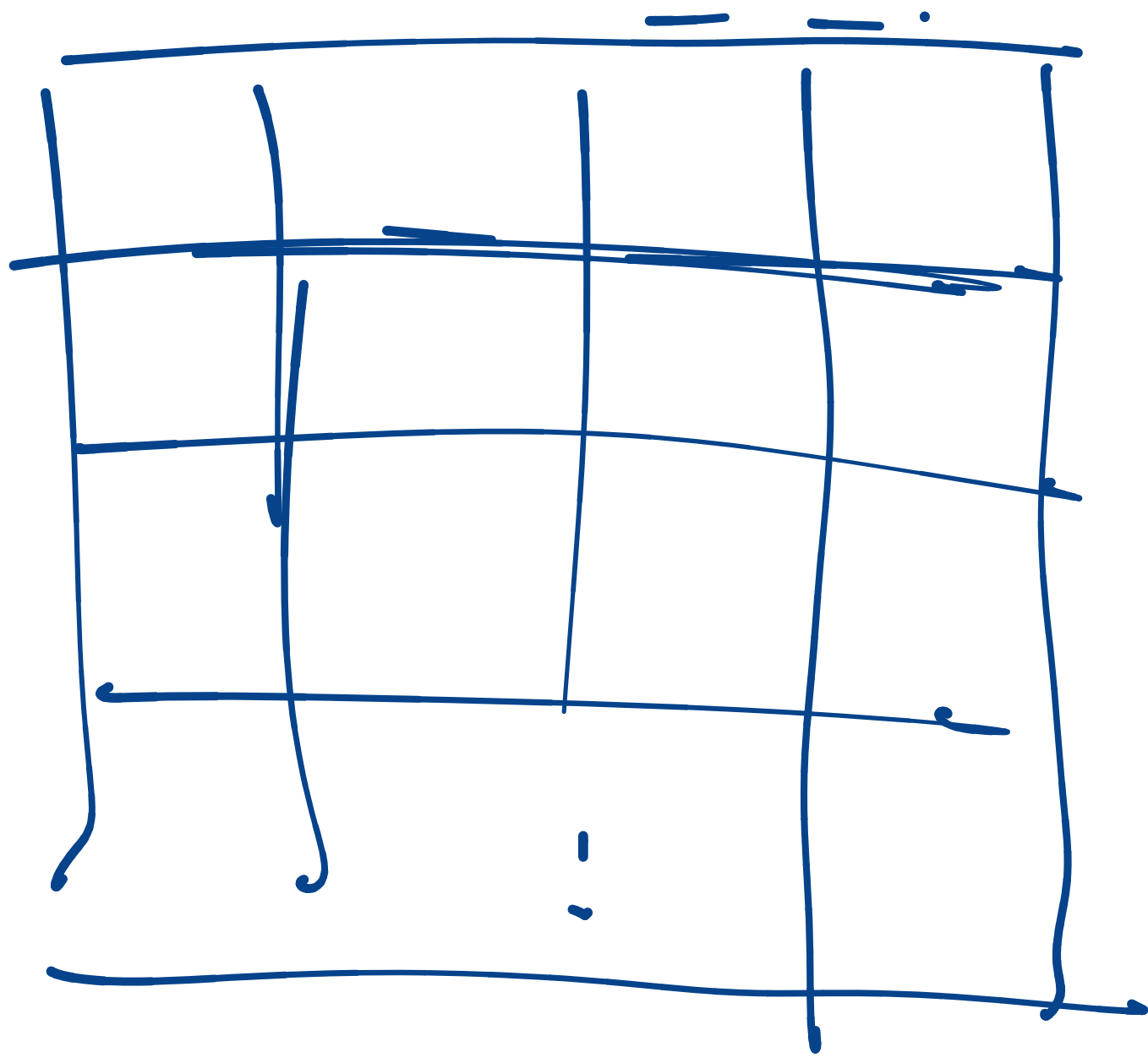
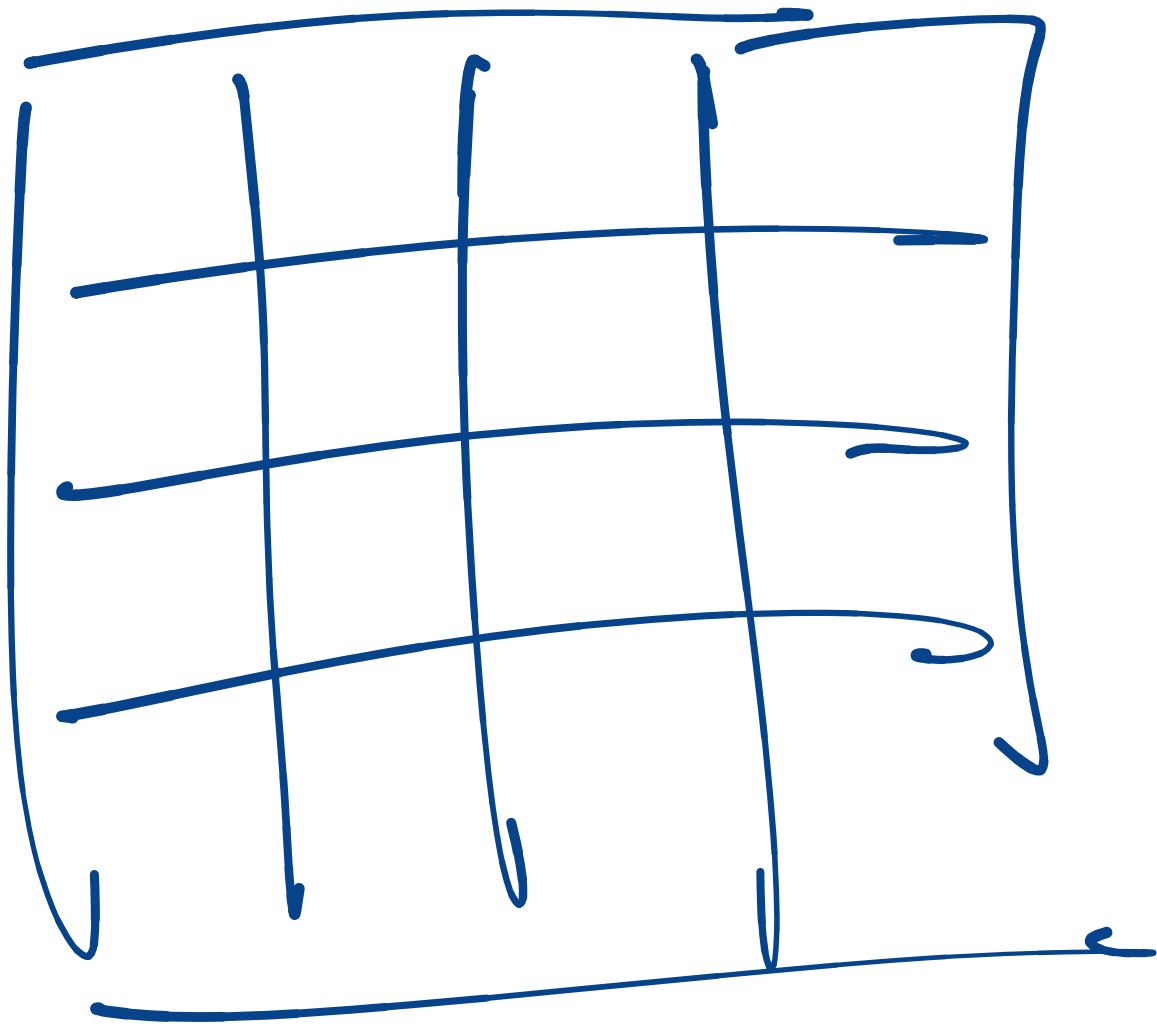


mem32

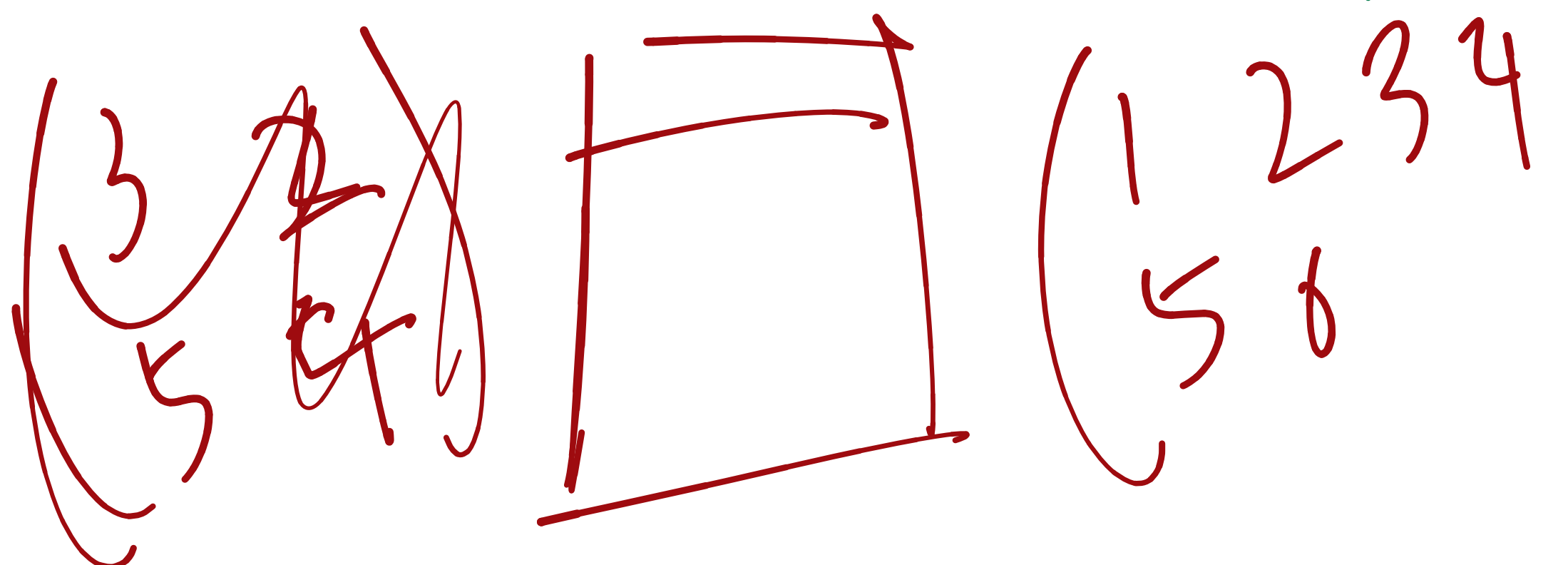
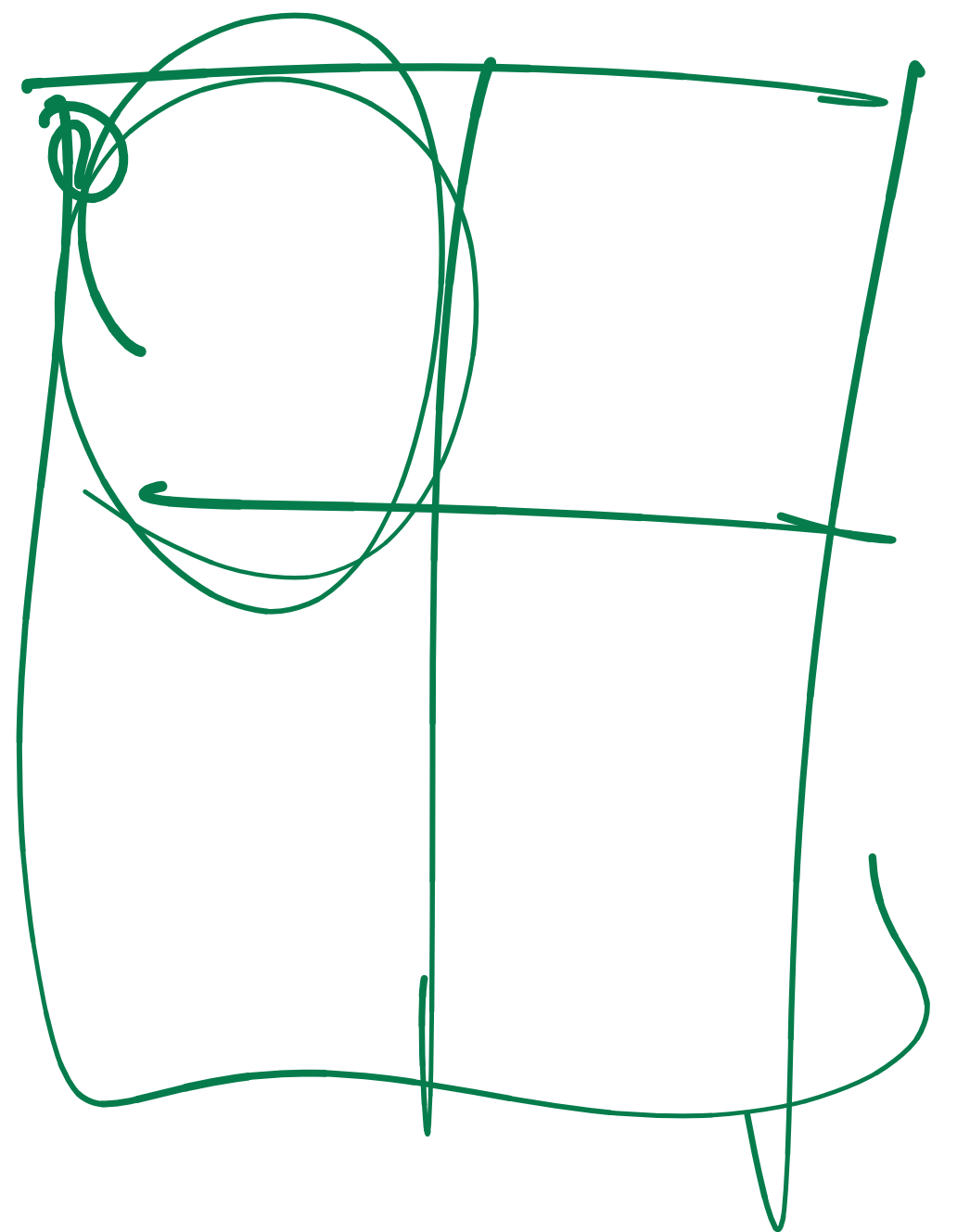


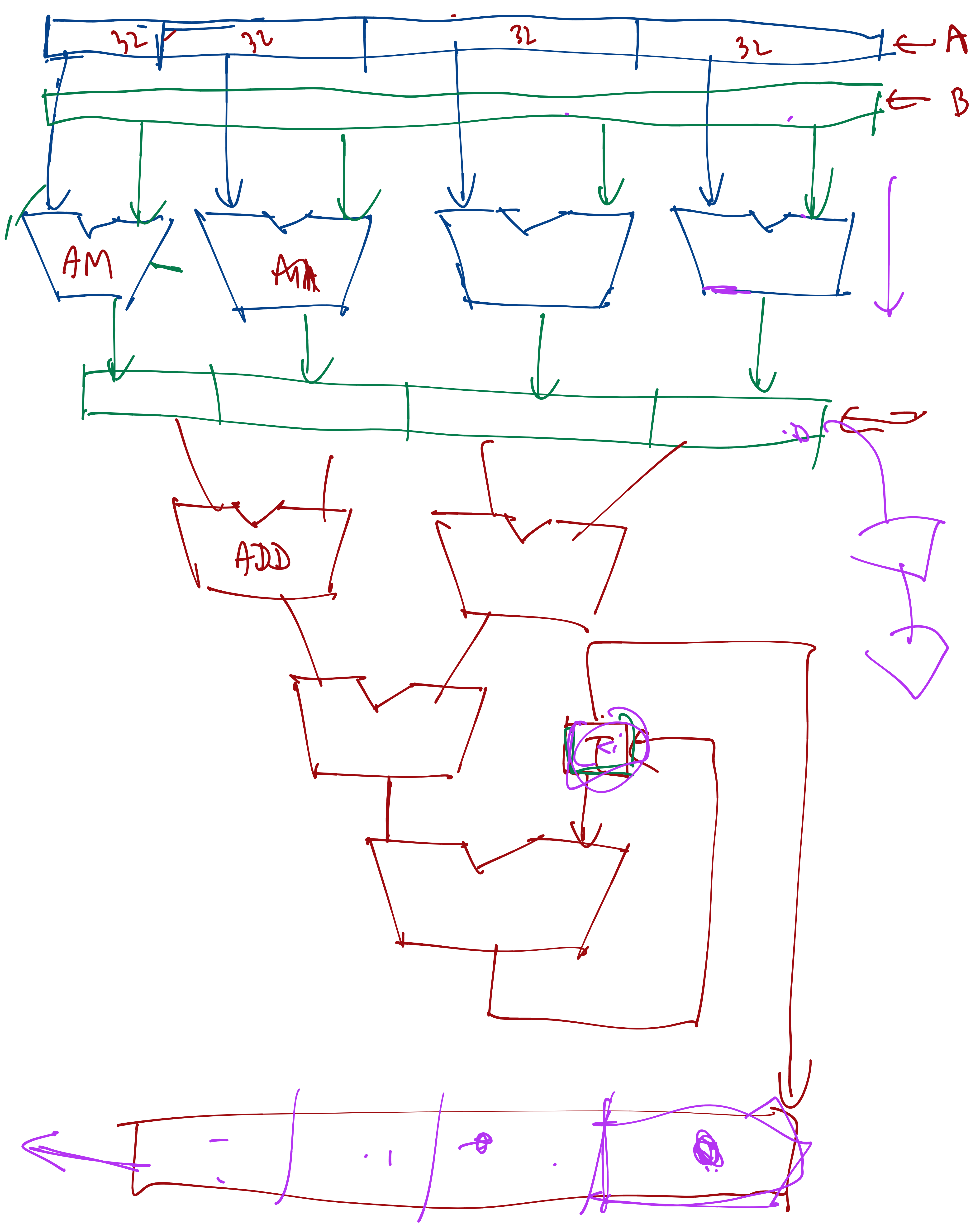
MAT-Size 4

PE Size 2

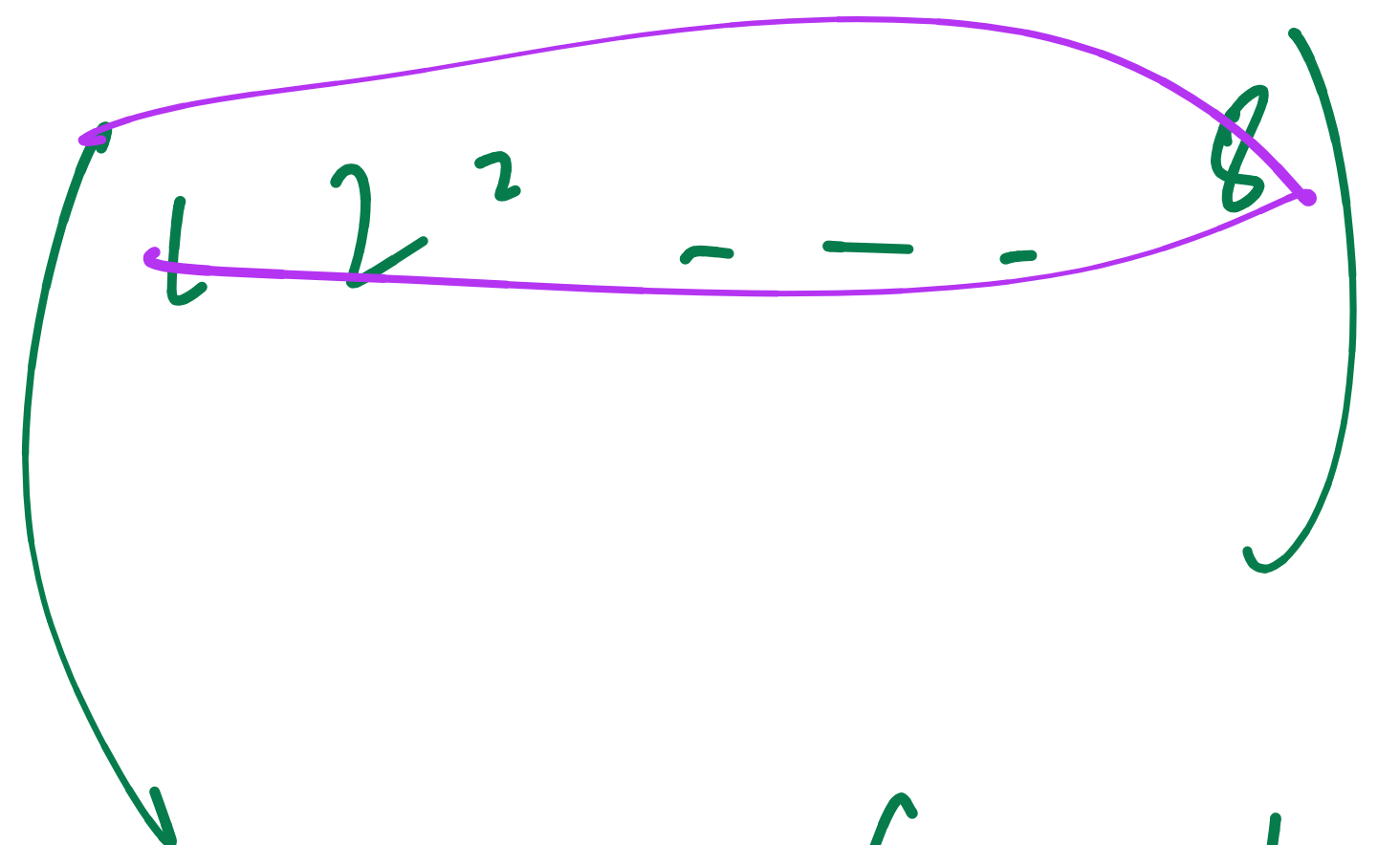


9 bit



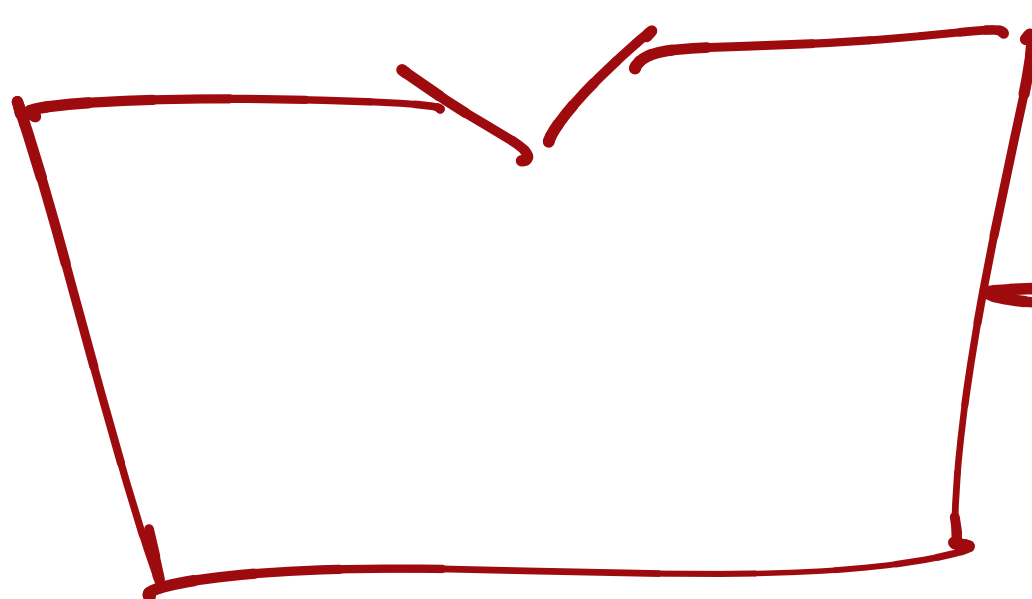
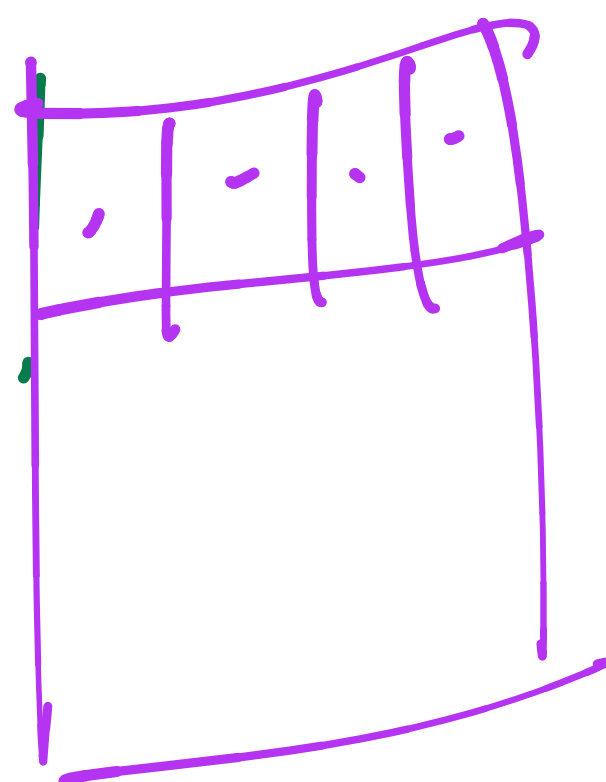
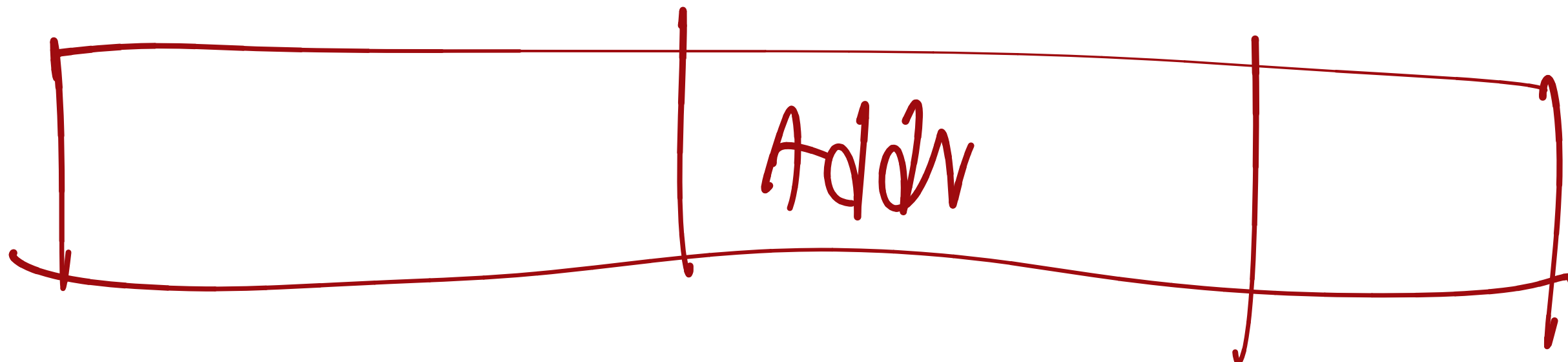


Load A  
CLRTMP  
DOT

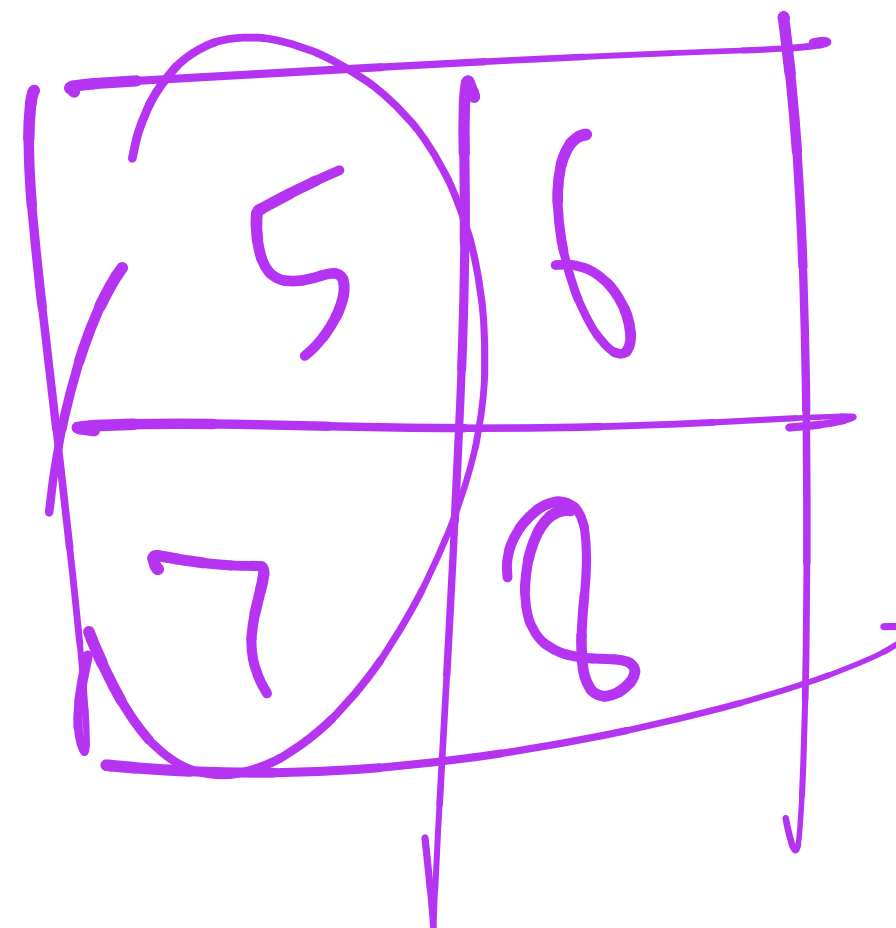
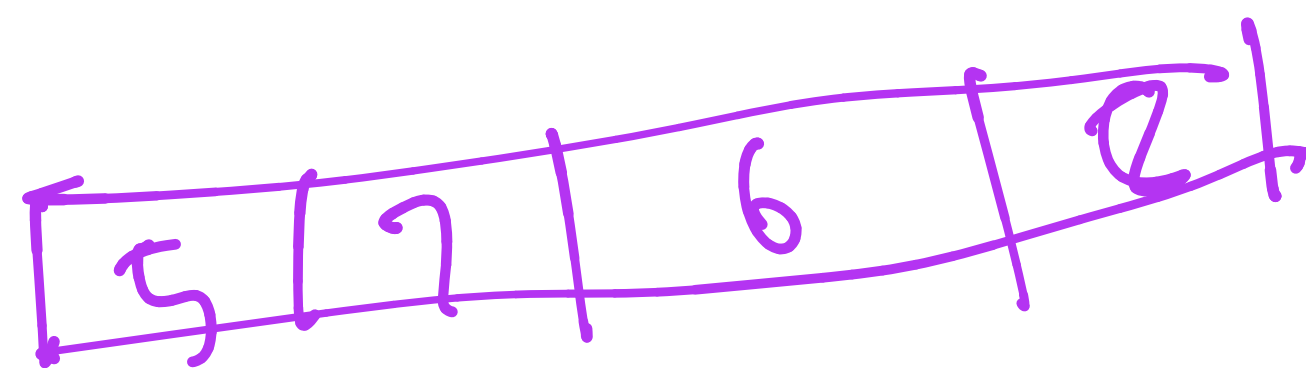
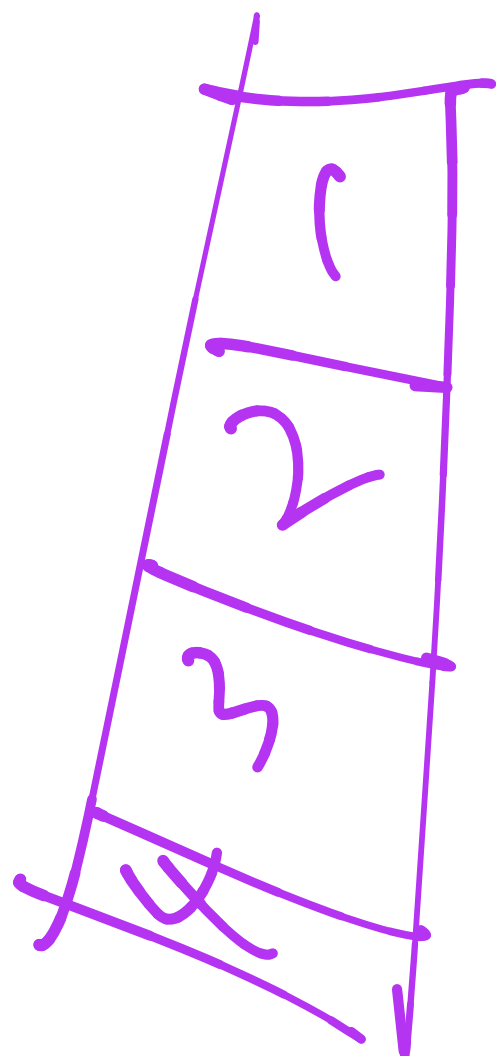
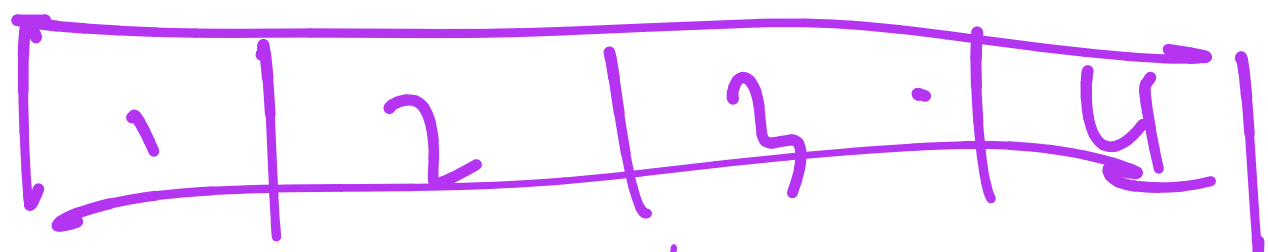
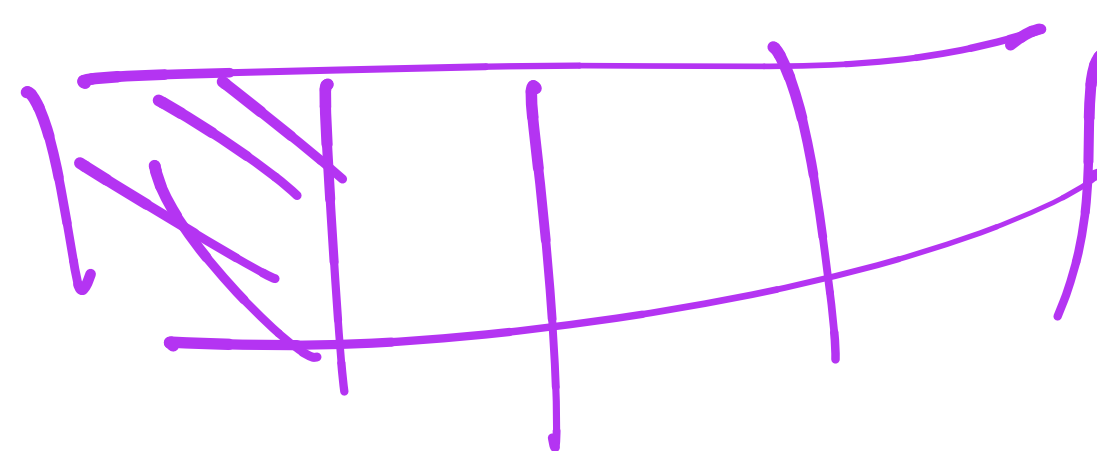
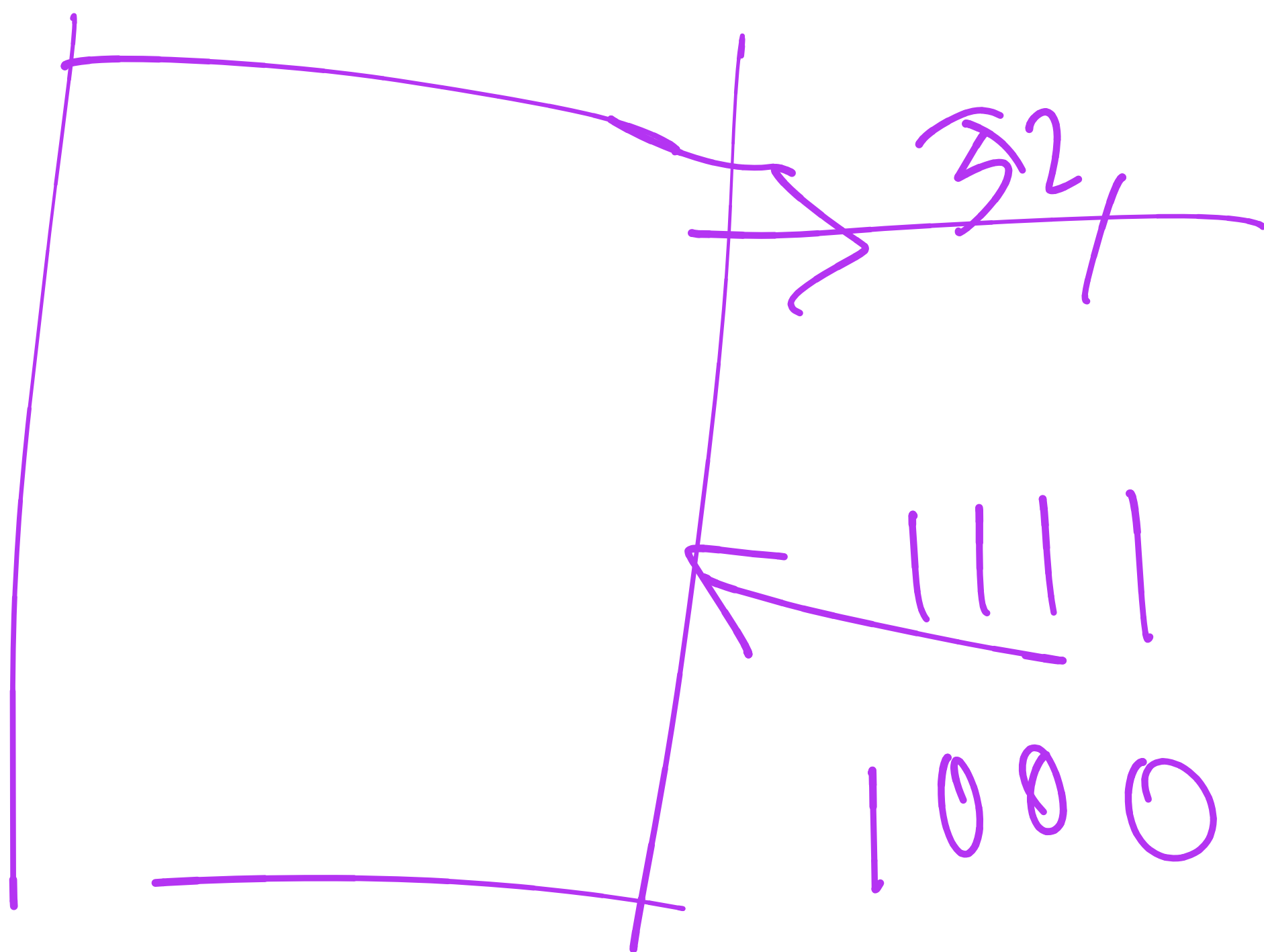


STR TMP, poshi

STORE



00 Mn)  
01 SUB  
10 ADD





32x1024

1 2 3  
4 5 6  
7 8 9

a b c  
d e f  
g h i

1	a
2	d
3	g
4	b
5	e
6	h
7	c
8	f
9	i

3 0

1 2 3 4 5

5x5

a b c d e

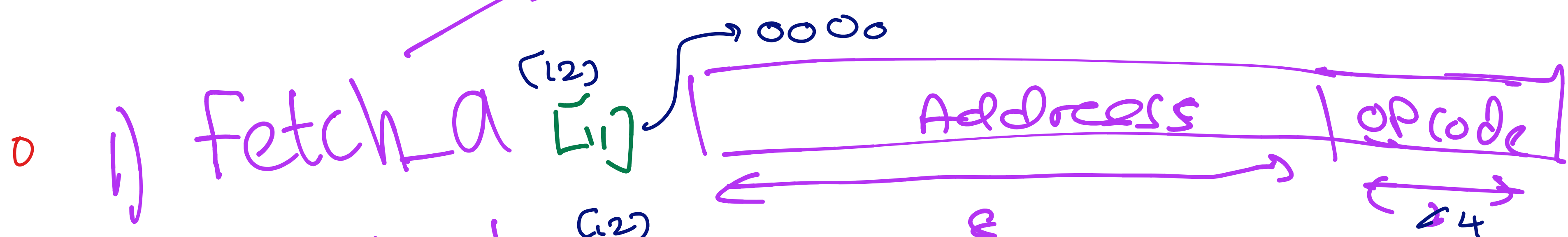
1	2	3	4
5	0	0	0

a	b	c	d
e	0	0	0

Block ram address = 8 bit

system verilog function  
\$readmem

Retrieve 4 words from RAM A.



1 2) fetch\_b <sup>(12)</sup>[11] → 0001

2 3) Add <sup>(4)</sup>[3] → 0100

3 4) subtract <sup>(4)</sup>[3] → 0101

4 5) Multipl <sup>(4)</sup>[3] → 0110

5 6) Dot Product <sup>(4)</sup>[3] → 0111

6 7) Store Temp S1 <sup>(4)</sup>[3] → 0010  
Store Temp S2 <sup>(4)</sup>[3] → 0011

8 8) STORE <sup>(4)</sup>[11] → 1001

9 9) STOP <sup>(4)</sup>[4] → 1111

\* First bit 0 ⇒ R/W  
1 ⇒ Arithmetic



BRAM A : width = 128 bits  
Depth = 256

BRAM B : width = 128 bits  
Depth = 256

Instruction BRAM : width = 16  
Depth = 4k

Result BRAM : width = 128 bits  
Depth = 256

PL.

### Fetch unit

- ✓input start. ← Things are done only if this is given.
- ✓input [12-1:0] instruction
- ✓input [128-1:0] data\_a
- ✓input [128-1:0] data\_b
- ✓input pe-stage-1-valid
- ✓input [128-1:0] pe-stage-1-output
- ✓input [32-1:0] pe-stage-2-output
- ✓input pe-stage-2-valid
- ✓output [3-1:0] pe-opcode.
- ✓output end ← This is held high when the program is finished.
- ✓output [128-1:0] result
- ✓output [12-1:0] program-counter