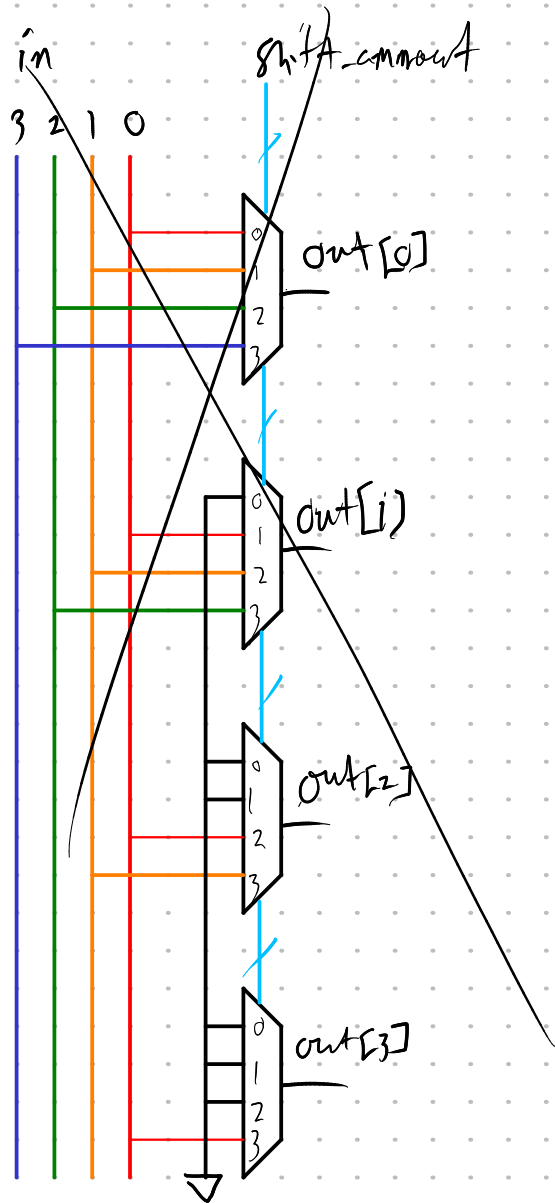


Logical Left Shift

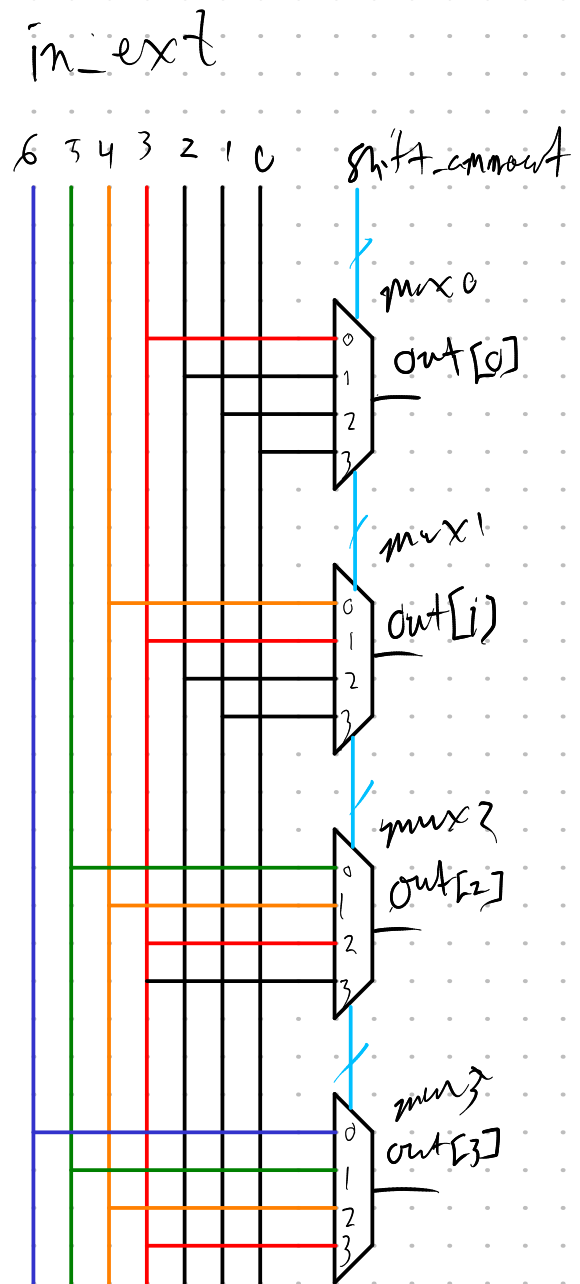
$in[n-1:0]$, $shift_amount[\lceil \log_2(n) \rceil - 1:0]$, $out[n-1:0]$

Schem uses $n=4$ for my sanity

IDEA: Concat in with bus of 0s.
Better for generate statements.



Wrong!
Backwards!

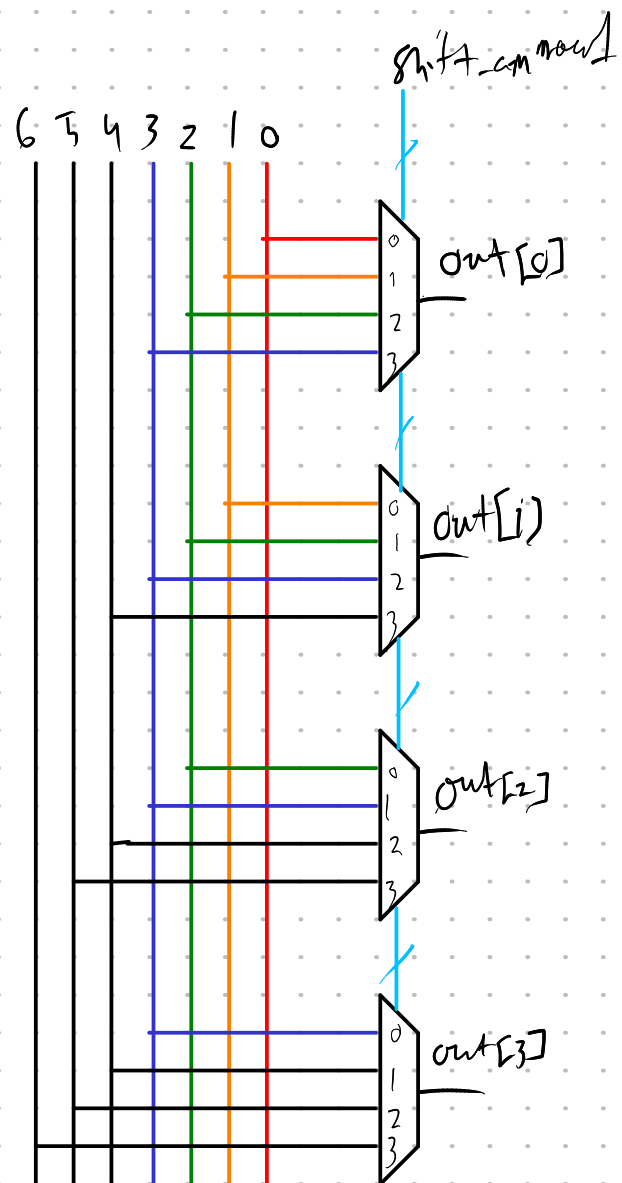
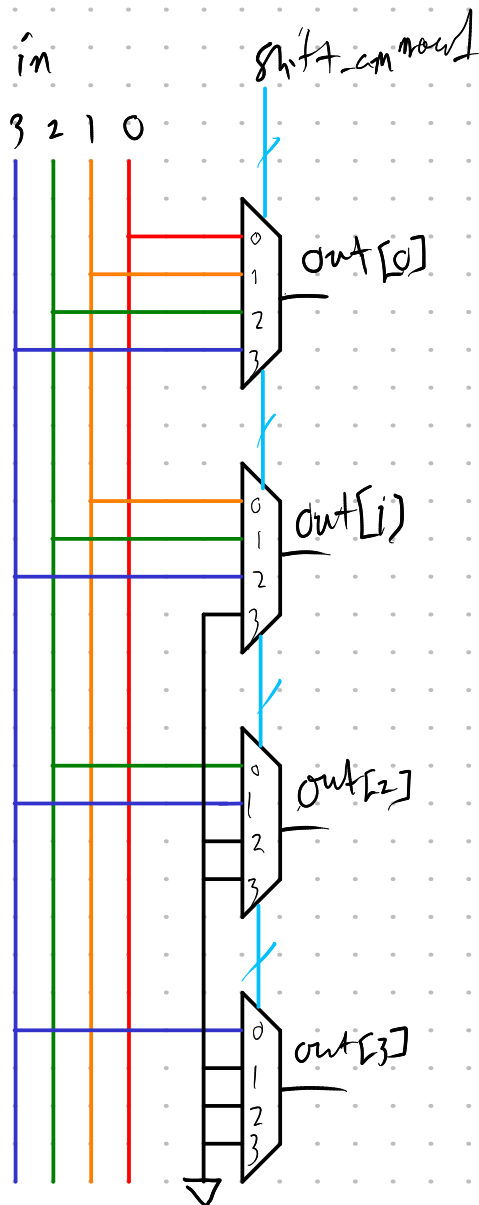


Logical Right Shift

$in[n-1:0]$, $shift_amount[\lceil \log_2(n) \rceil - 1:0]$, $out[n:0]$

Schem using $n=4$ for my
Sanity

Idea: concatenate with 0s

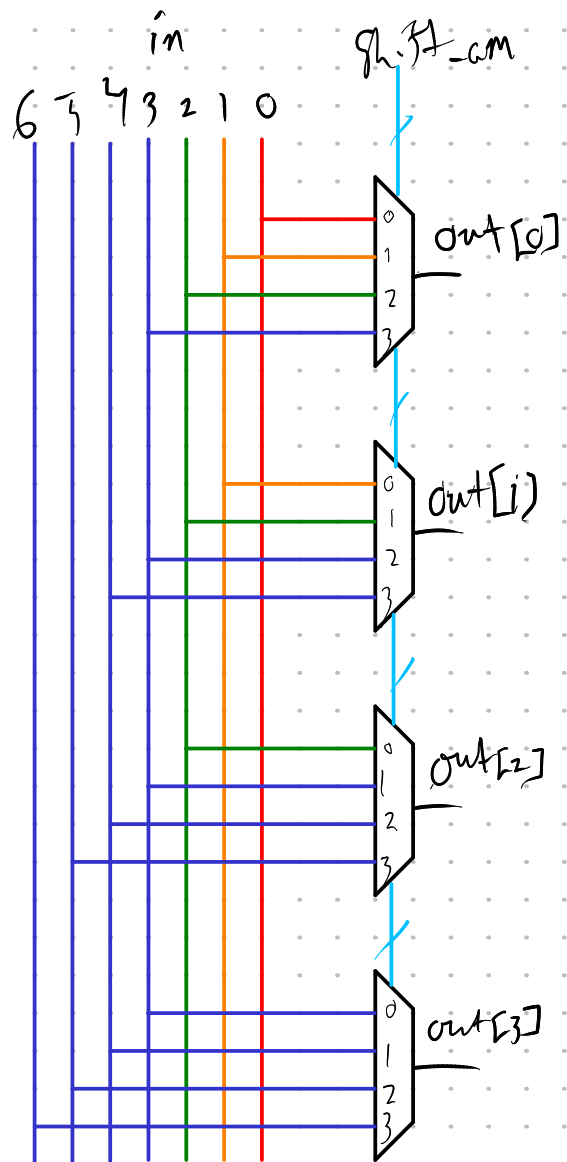
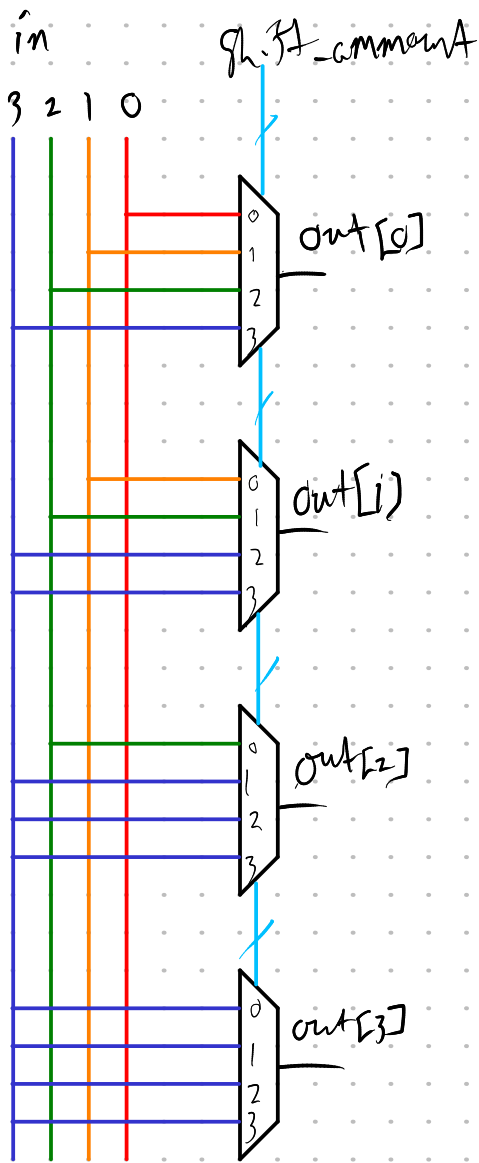


Arithmetic Shift Right

$in[n-1:0]$, $shift_amount[\log_2(n)-1:0]$, $out[n-1:0]$

Schem using $n=4$

Then extend bus again



ALU

ALU_AND = 4'b0001,
 ALU_OR = 4'b0010,
 ALU_XOR = 4'b0011,
 ALU_SLL = 4'b0101,
 ALU_SRL = 4'b0110,
 ALU_SRA = 4'b0111,
 ALU_ADD = 4'b1000,
 ALU_SUB = 4'b1100,
 ALU_SLT = 4'b1101,
 ALU_SLTU = 4'b1111

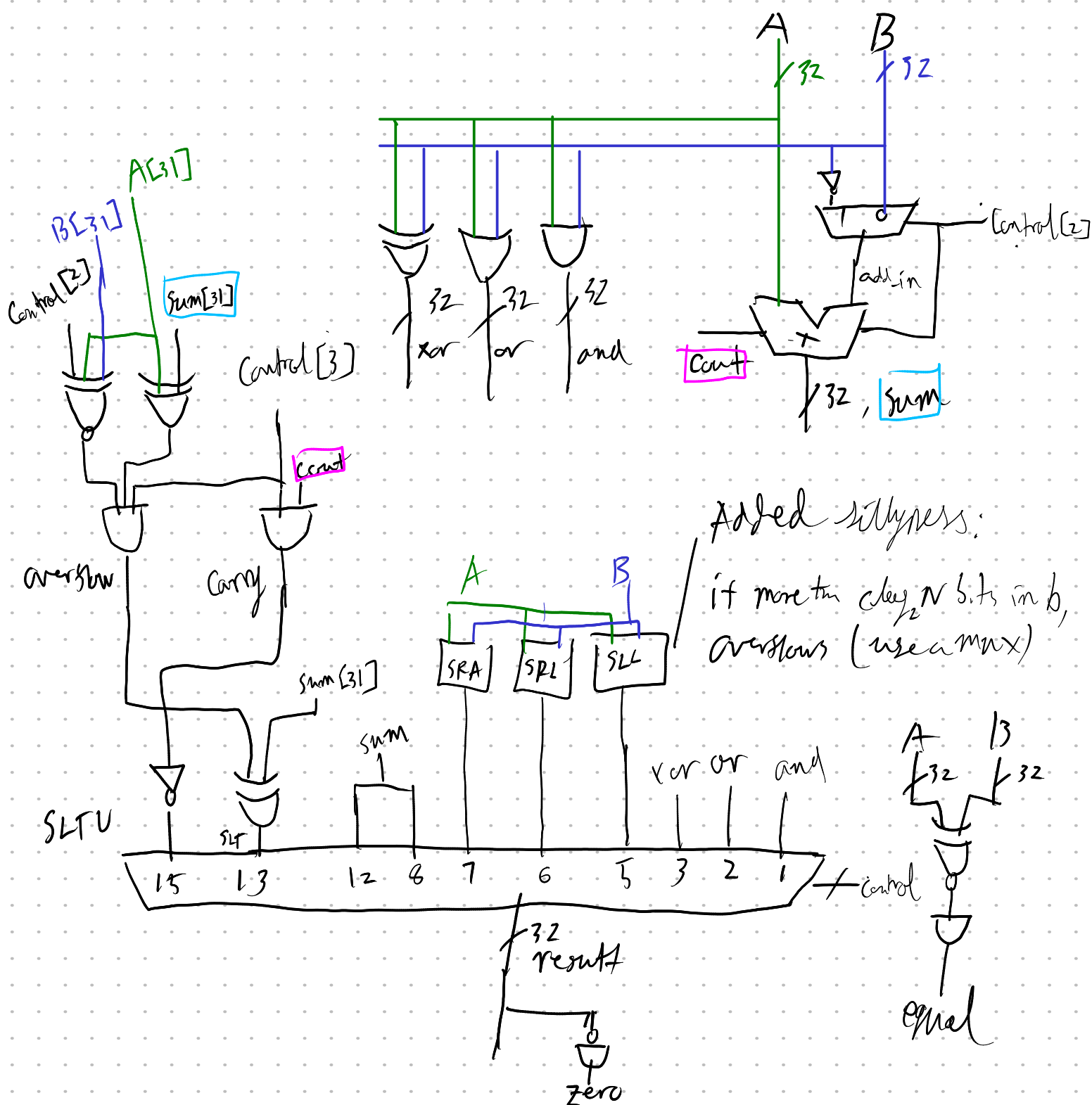
Bitwise

less than, signed

less than, unsigned

in: a, b, Control

Out: Result, overflow, Zero, equal



Register file

N : Bit width (word size)
 A : \log_2 Number of registers

