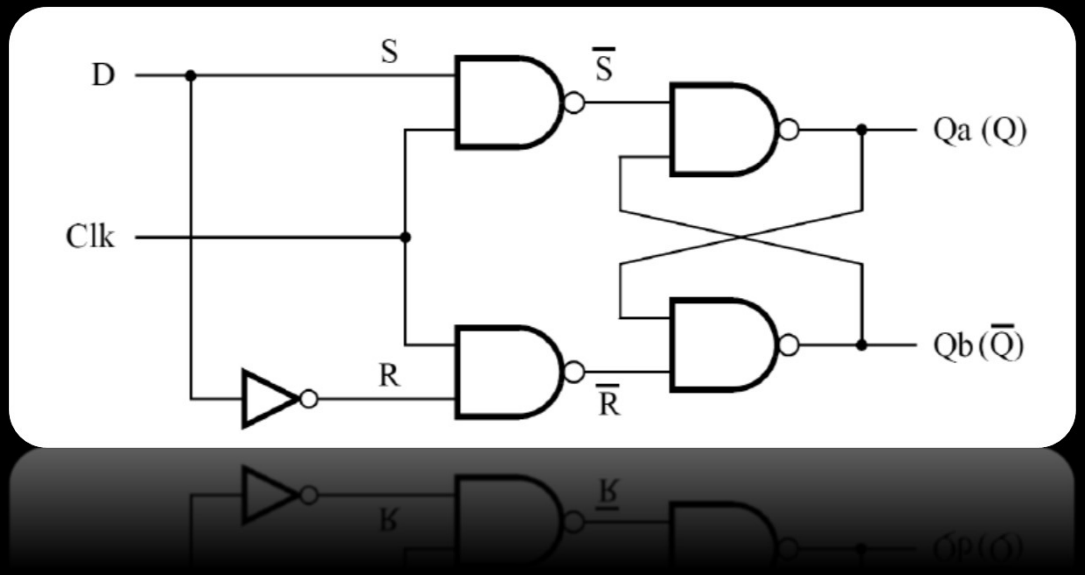
Lab 4 Preparation

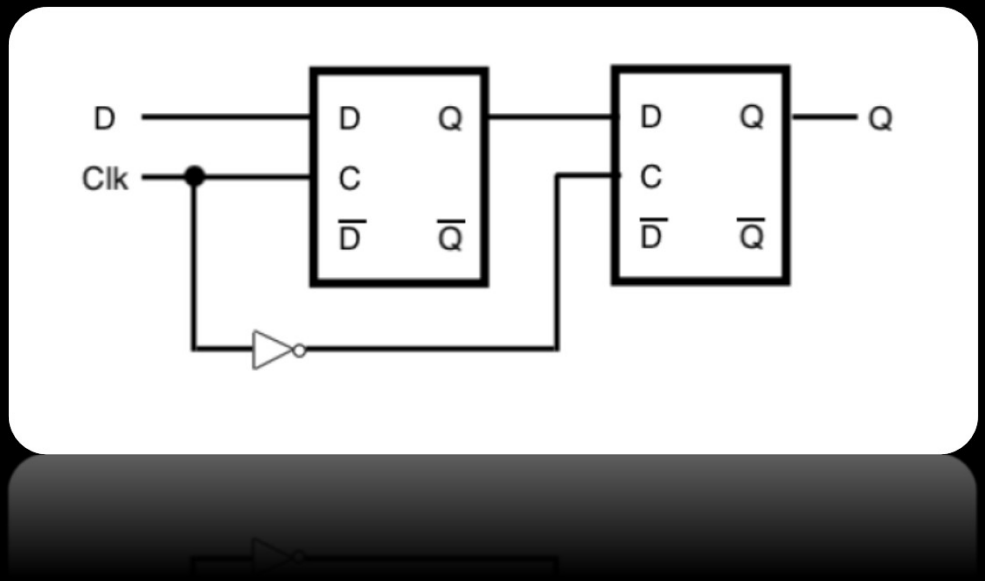
Lab 4 – Part I

- Create a D latch out of NAND gates.



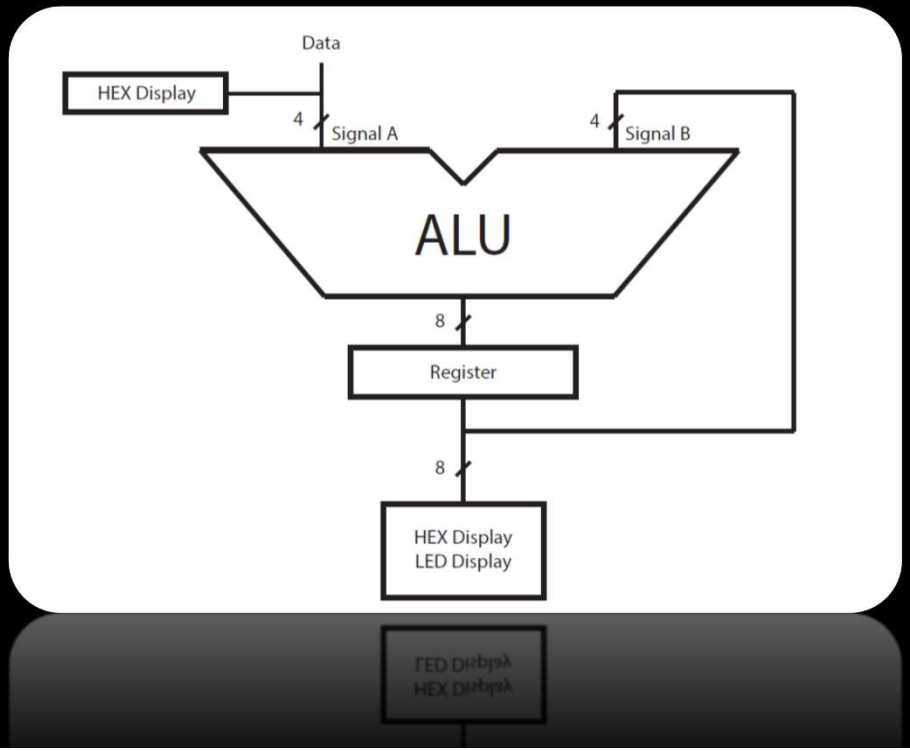
Lab 4 – Part I

- Then, create a D flip-flop out of D latches.



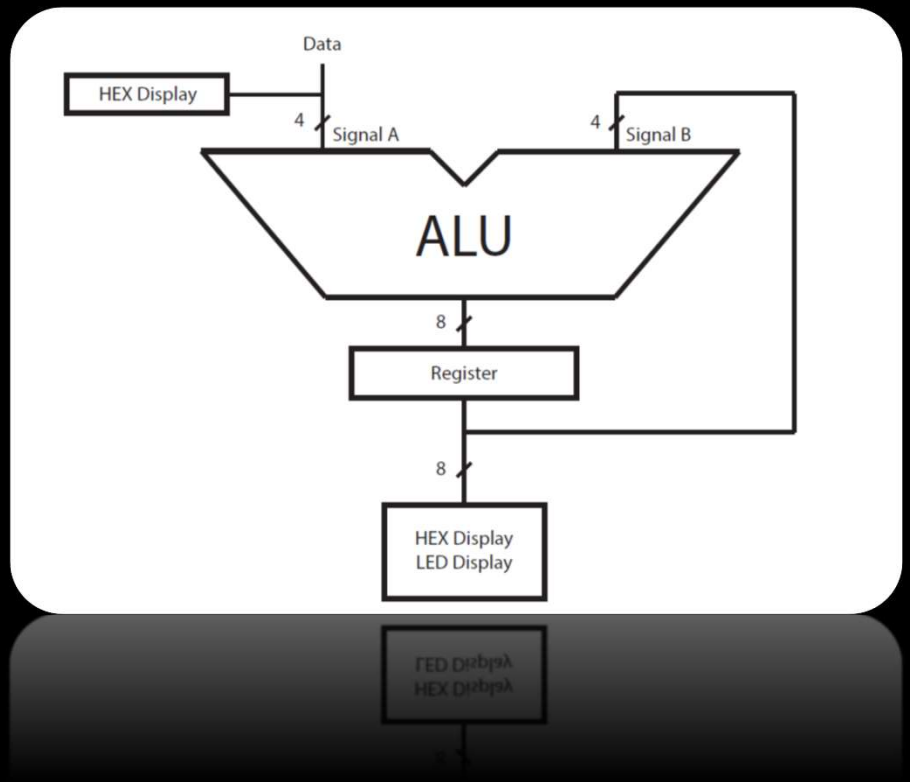
Lab 4 – Part II

- Enhance the ALU from last week:
  - More operations (e.g. multiplication, shifting)
  - Memory (aka registers)



Lab 4 – Part II

- New operations:
  - Left shift
  - Logical right shift
  - Multiplication
- All of these are supplied through the collection of components in Logisim.



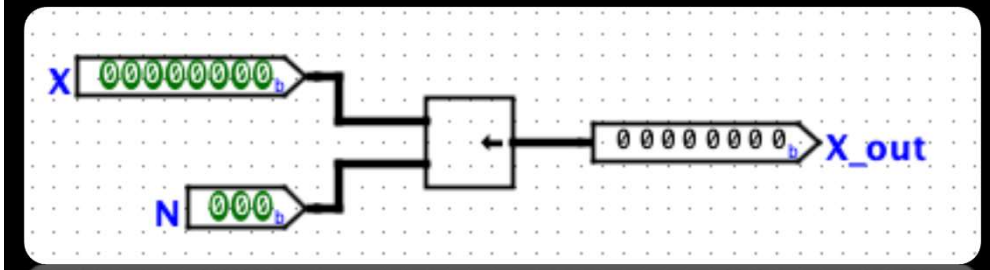
Logic vs. Arithmetic Shift

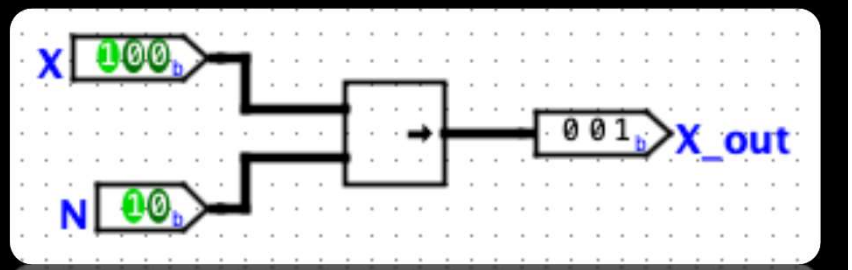
- Arithmetic right shifts replicate the sign bit instead of using zero to fill in the most-significant bit(s).
  - Needed if dealing with signed numbers (e.g., 2's complement notation)
- Logical right shifts fill in missing bits with zeroes.

- Example: Shift `10010000` right by 3 bits
  - Arithmetic → **`111`**`10010`
  - Logical → **`000`**`10010`

Shifter Components

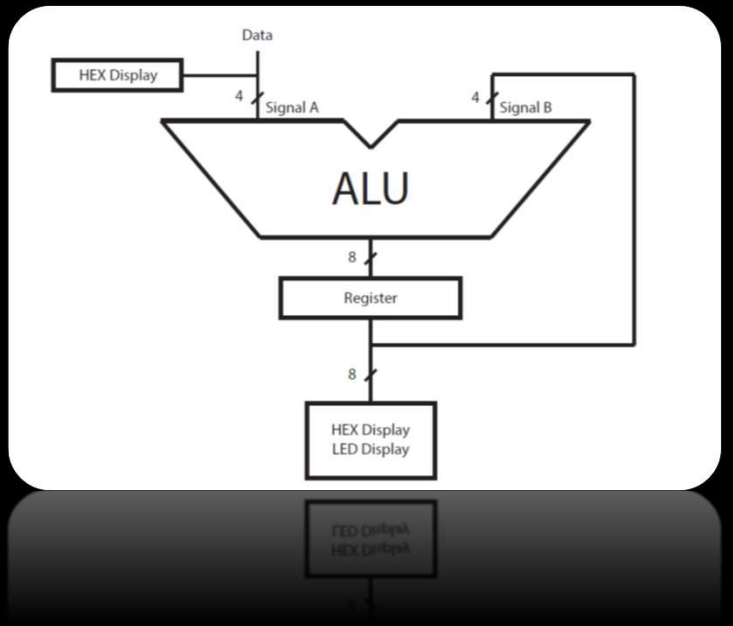
- Can be found under Arithmetic > Shifter.
  - You can change the shift type in Properties.
  - More details can be found at:
    - http://www.cburch.com/logisim/docs/2.3.0/libs/arith/shifter.html
- Logic Shift (left or right)





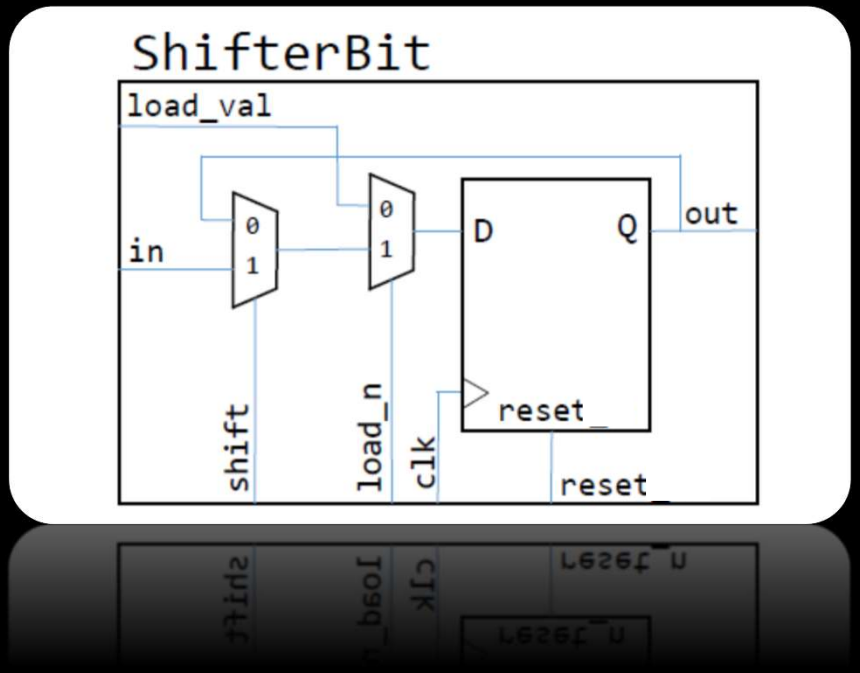
Lab 4 – Part II

- Note: ALU output is now stored in a set of flip-flops called a register.
  - 8 bits (flip-flops) long
  - Compoment found in Memory > Register.
  - You can change the Number of flip-flops in Properties > Data bits.



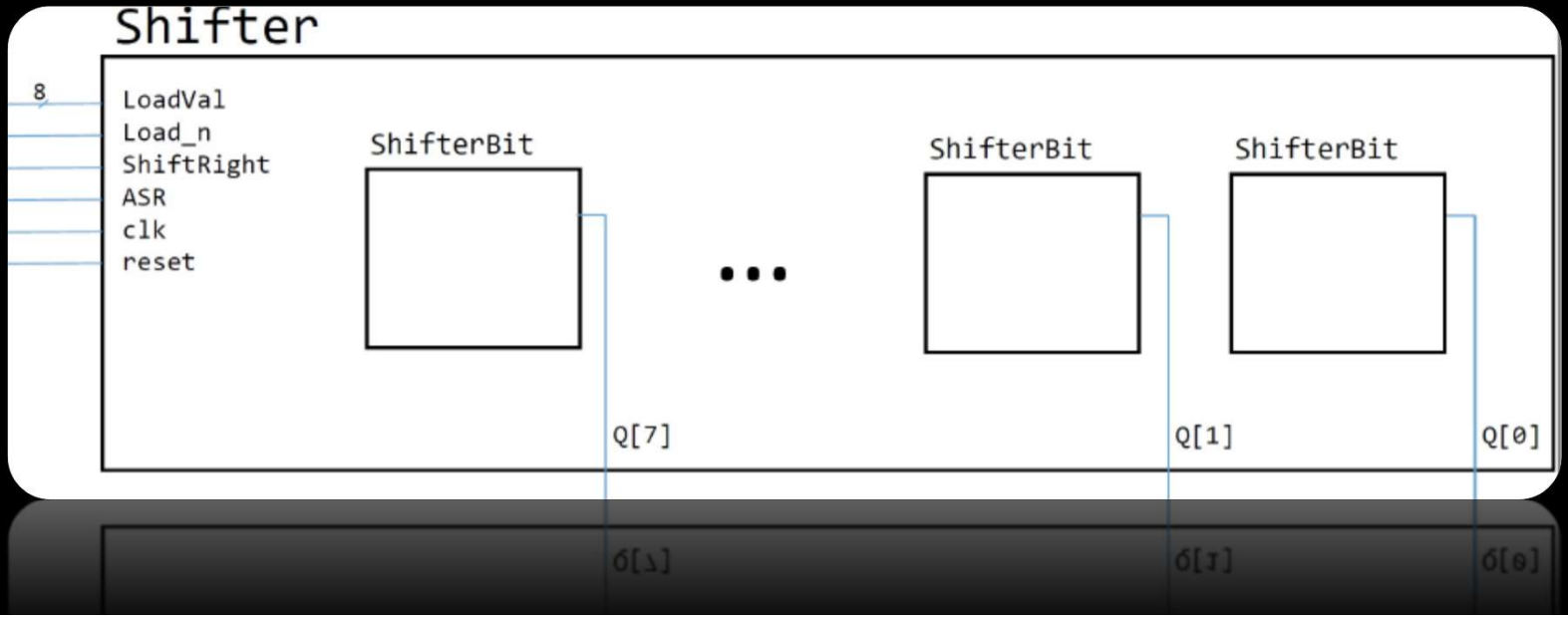
Lab 4 – Part III

- Make a shifter unit, for a single bit.
  - See diagram below.



Lab 4 – Part III

- Connect 8 shifter units together.
- Note:
  - `Load_n` is active-low, loads all bits from `LoadVal` into ShifterBit units when `clk` goes high.
  - `reset` is asynchronous (doesn't wait for `clk` signal).
  - `ShiftRight` tells you when to shift, `ASR` tells you what kind of shift to do.



Why is Shift Important?

- What is the sum of `01101101` and `01101101`?

`11011010`

Remember what happens here!

- Try the following:
  - `00110` logic shift right by 1 bit
  - `00110` logic shift left by 1 bit

A logic shift right of `A` by `N` bits results in: $A*2^N$
A shift left of `A` by `N` bits results in: $A/2^N$

Load register

- N-bit number = N D-flipflops synchronized to the same clock signal
- You can load a register's value (all bits at once), by feeding signals into each flip-flop:
  - In this example: a 4-bit load register.