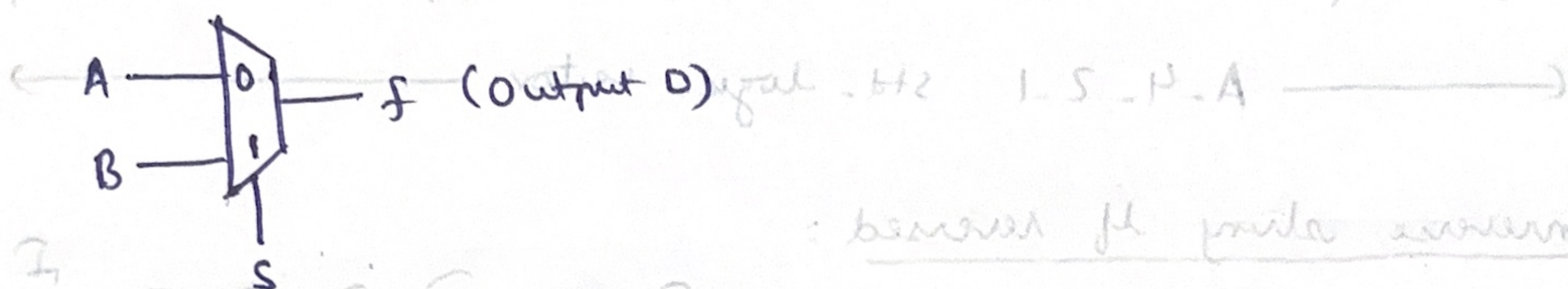


# lab 4:

## Universal rotator.

First define MUX-2-1 entity:

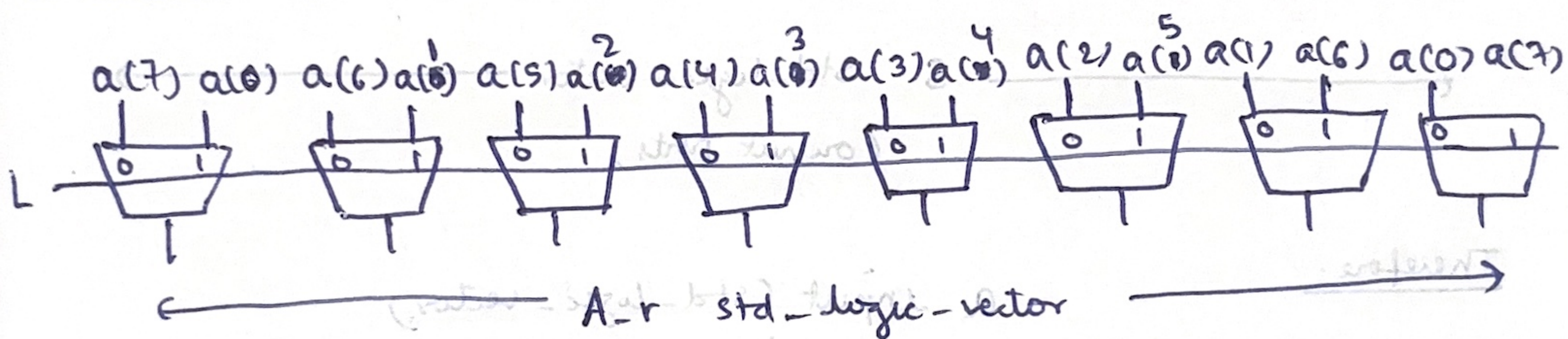
$$f = (A \text{ and } (\text{not } S)) \text{ or } (B \text{ and } S)$$



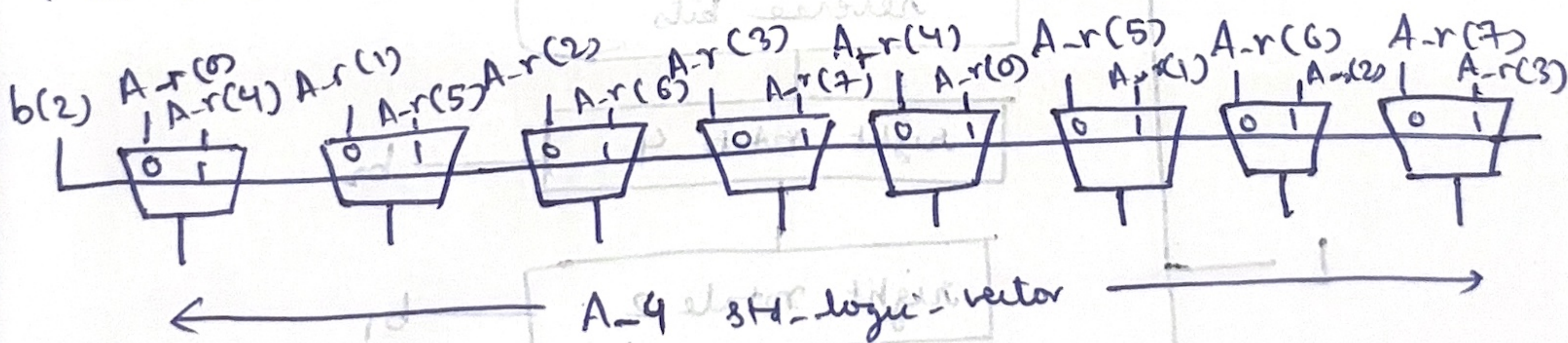
we use this mux to implement universal rotator.

left rotate check: (L)

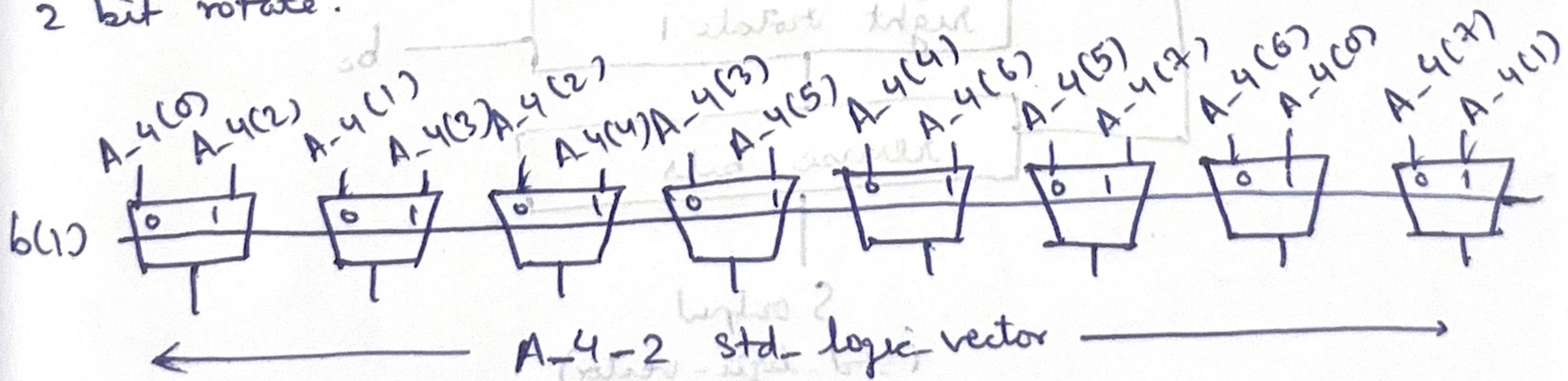
if L is high, rotate ~~right~~ <sup>left</sup>, else rotate right. Default rotate is right, hence if L high, we reverse string and unreverse at the end.



4 bit rotate:

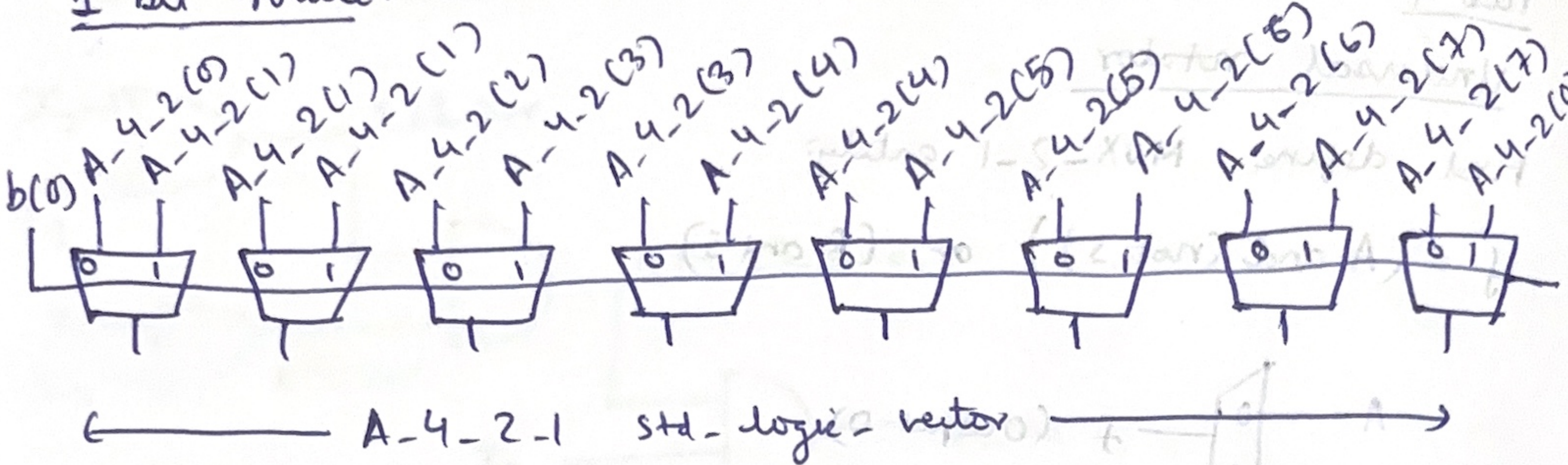


2 bit rotate:

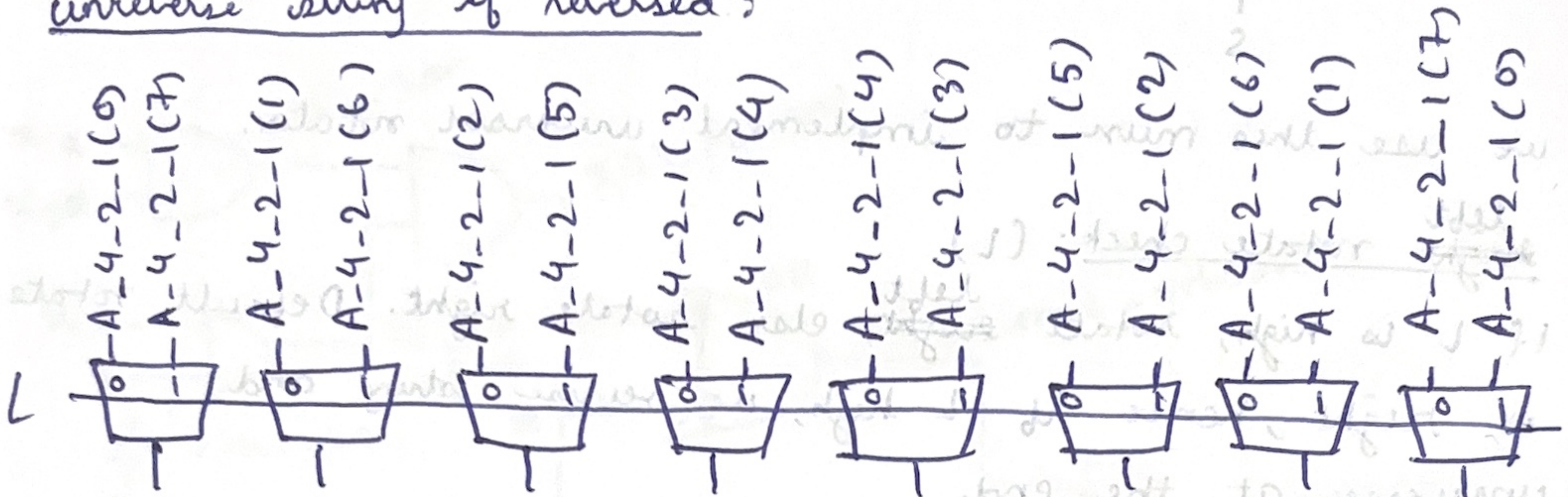




1 bit rotate:

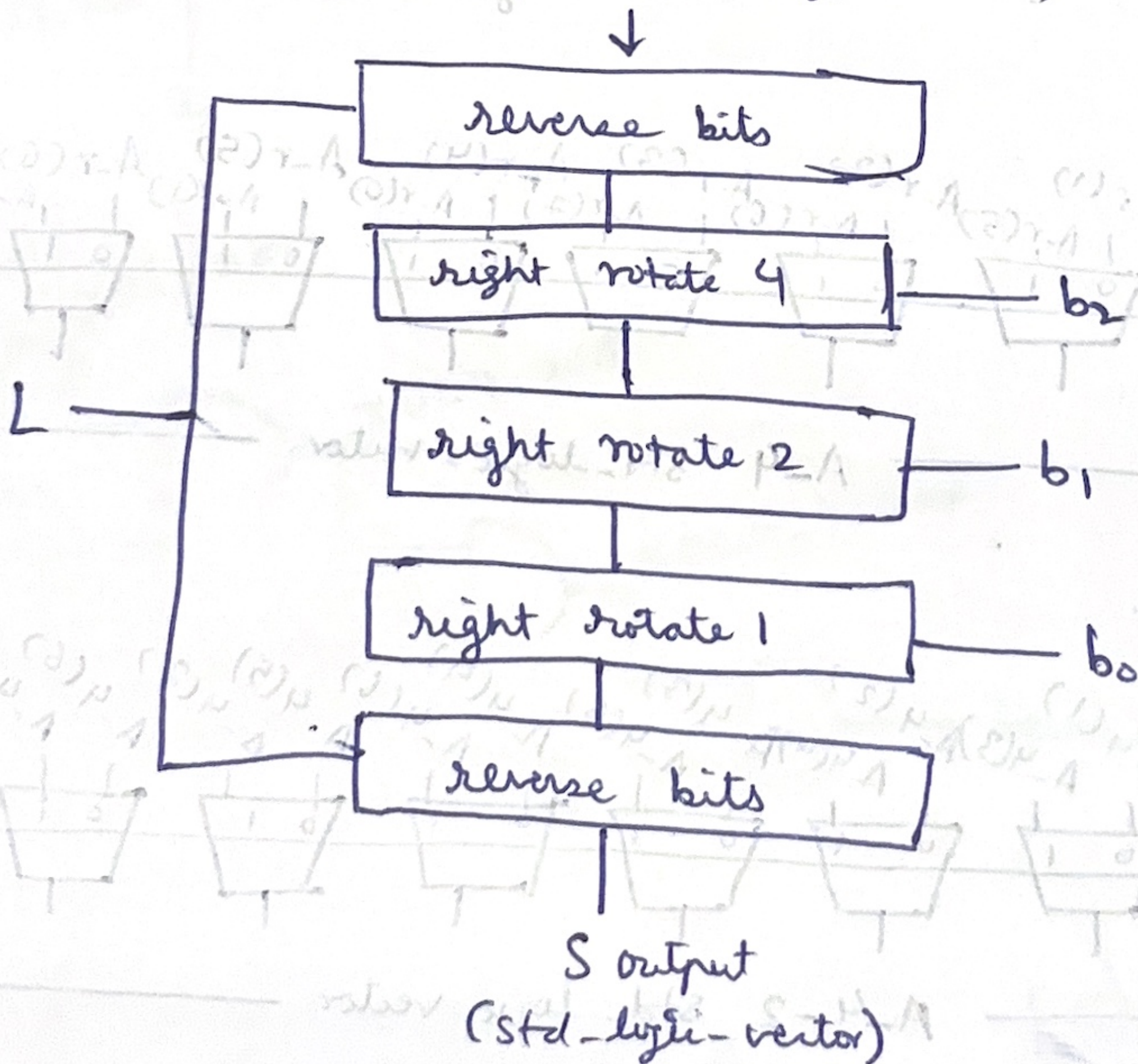


unreverse string if reversed:

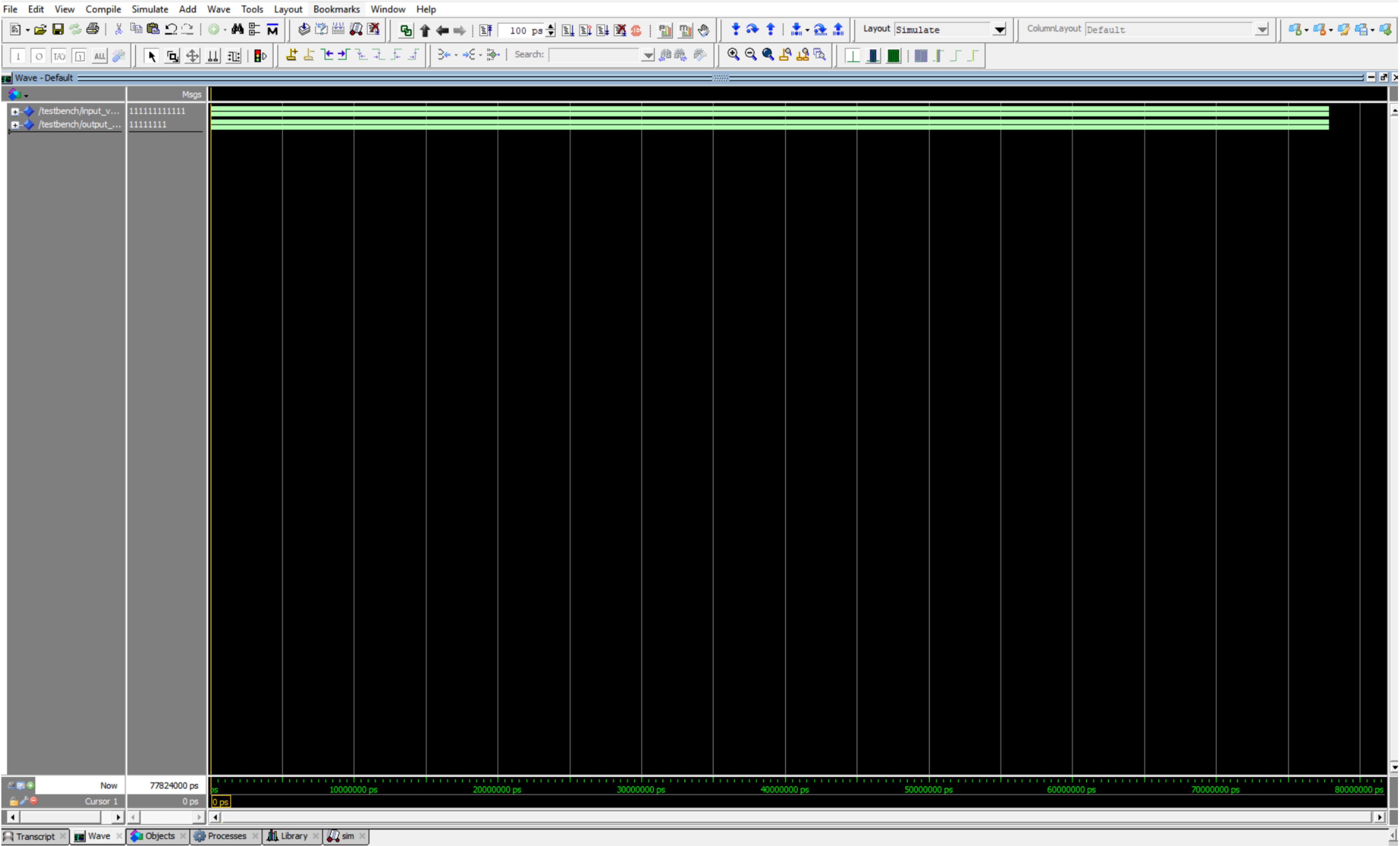
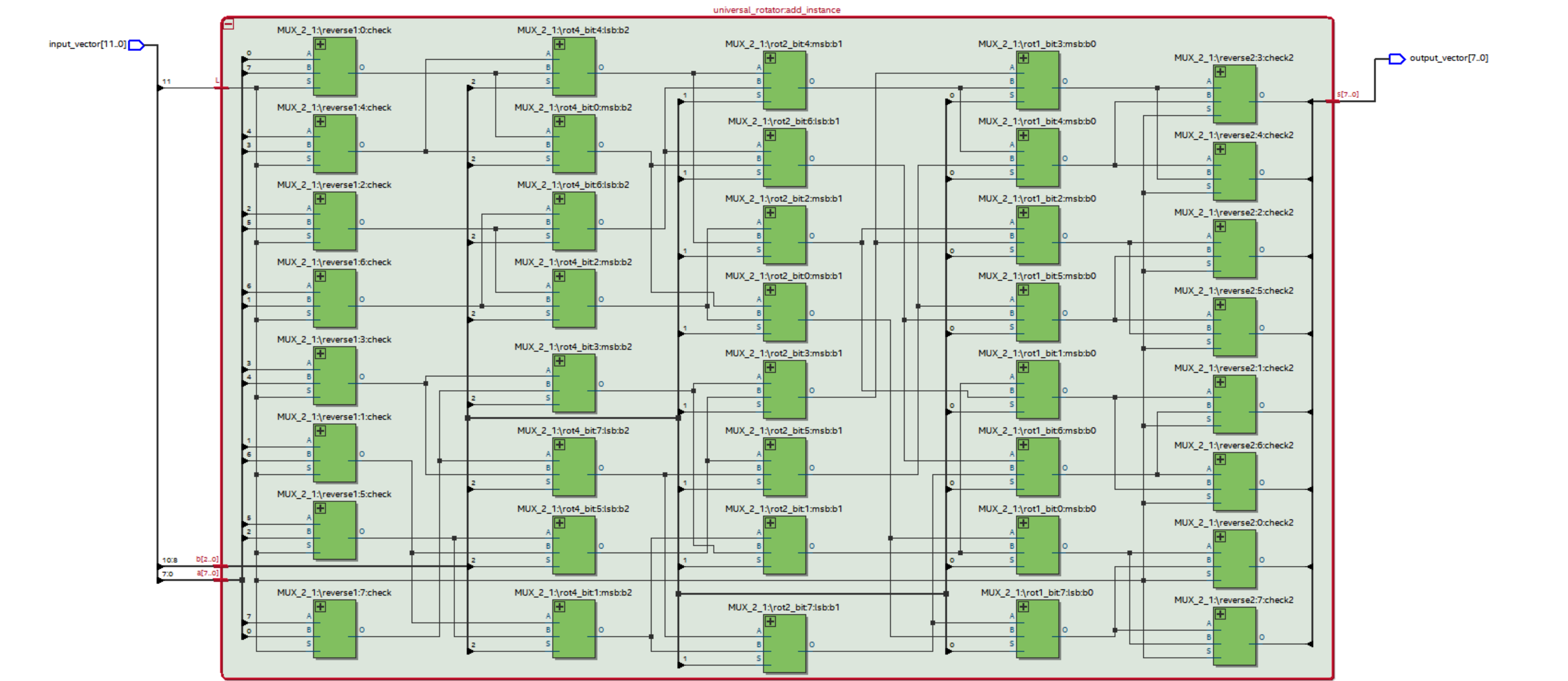


Therefore:

a input (std-logic-vector)







```
# vsim -t lps -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L fiftyfivenm -L rtl_work -L work -voptargs="+acc" Testbench
# vsim -t lps -L altera -L lpm -L sgate -L altera_mf -L altera_lnsim -L fiftyfivenm -L rtl_work -L work -voptargs="+acc" Testbench
# Start time: 10:57:33 on Aug 26, 2025
# Loading std.standard
# Loading std.textio(body)
# Loading ieee.std_logic_1164(body)
# Loading work.testbench(behavior)
# Loading work.dut(dutwrap)
# Loading work.gates
# Loading work.universal_rotator(struct)
# Loading work.mux_2_1(logic)
#
# add wave *
# view structure
# .main_pane.structure.interior.cs.body.struct
# view signals
# .main_pane.objects.interior.cs.body.tree
# run -all
# ** Note: SUCCESS, all tests passed.
# Time: 77824 ns Iteration: 0 Instance: /testbench
```