

VLSI Design

Lecture 14: Switch networks, Combinational testing

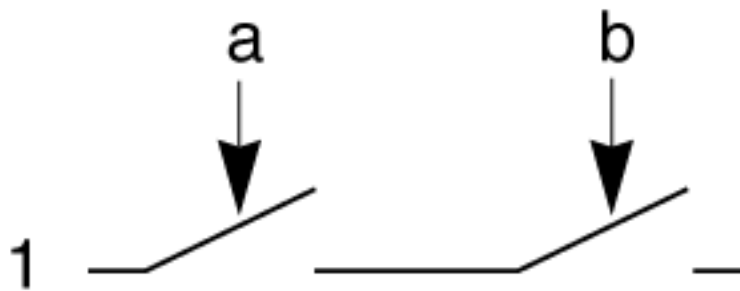
Shaahin Hessabi

Department of Computer Engineering

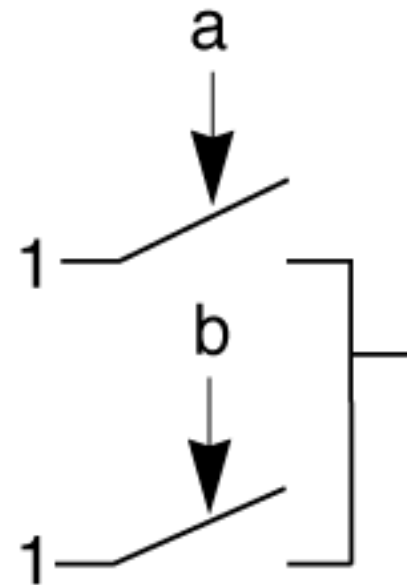
Sharif University of Technology

Adapted, with modifications, from lecture notes prepared by the
author (from Prentice Hall PTR)

Boolean functions and switches



pseudo-AND

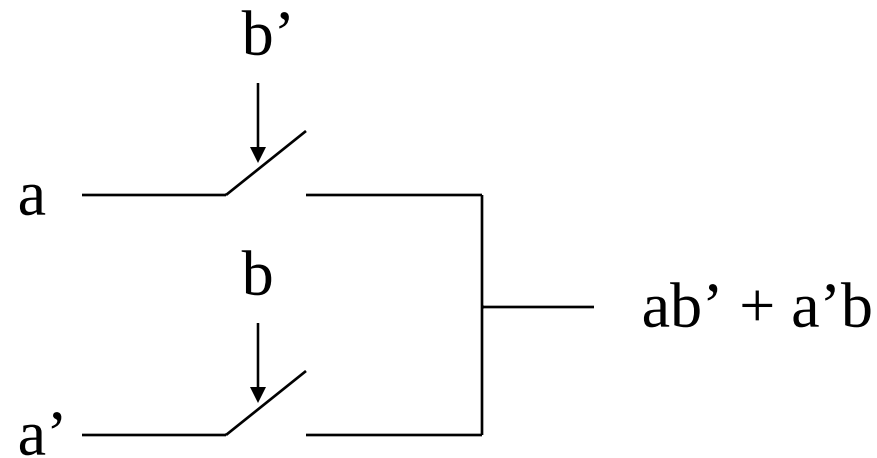


pseudo-OR

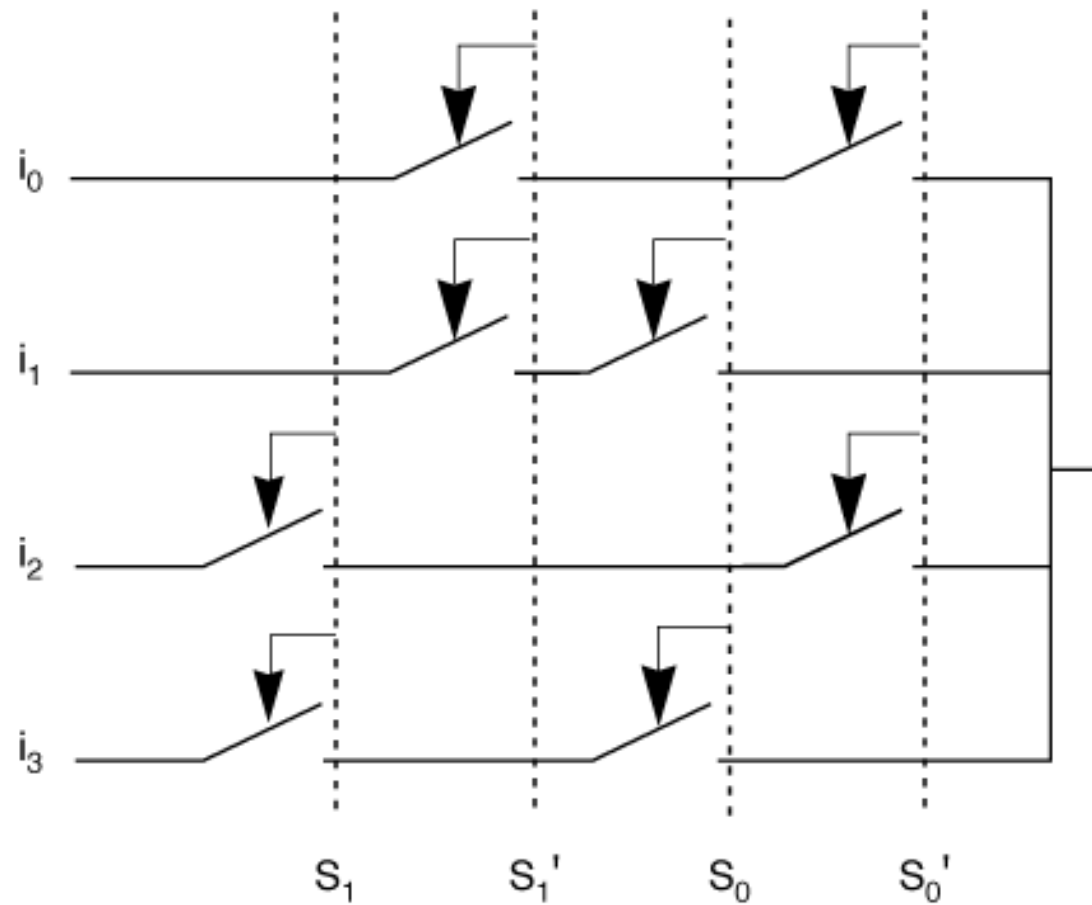
Driving switch outputs

- If switch network output is not connected to power supply through switch path, output will float.
- Switch network inputs may be connected to power supply or logic signals.

Switching logic signals

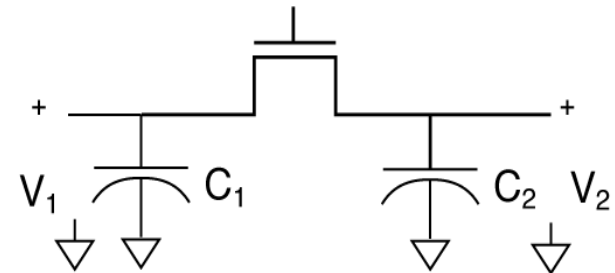


Switch multiplexer



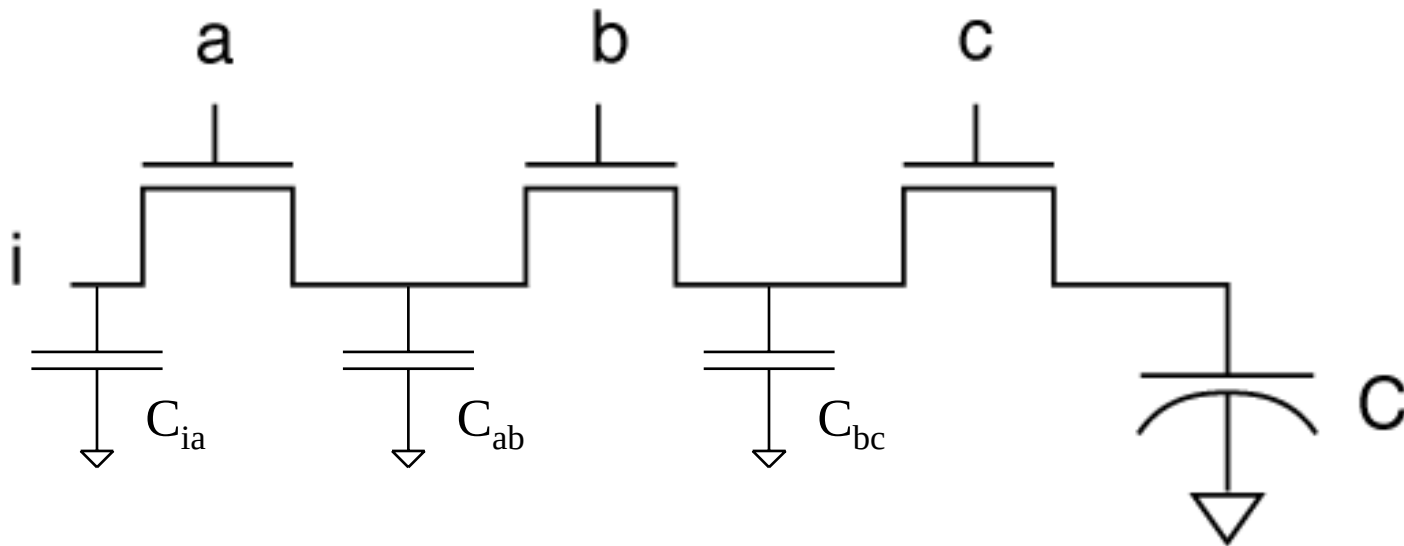
Charge sharing

- Interior nodes in a switch network may not be driven.
- Charge can accumulate on small parasitic capacitances.
- Shared charge can produce erroneous output values.
- At undriven nodes, charge is divided according to capacitance ratio.



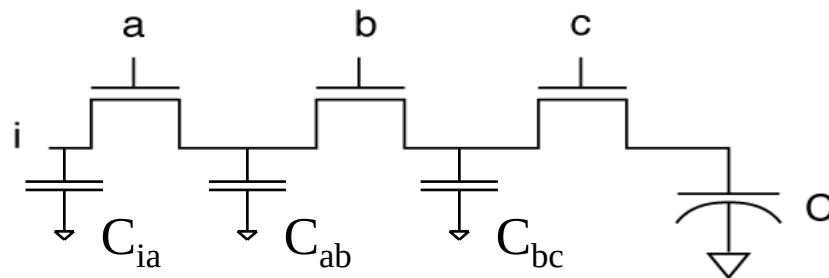
Charge sharing example

- Long chains of switches have intermediate nodes which may be disconnected from power supplies.



Charge over time

time	i	C_{ia}	a	C_{ab}	b	C_{bc}	c	C
0	1	1	1	1	1	1	1	1
1	0	0	1	0	0	1	0	1
2	0	0	0	1/2	1	1/2	0	1
3	0	0	0	1/2	0	3/4	1	3/4
4	0	0	1	0	0	3/4	0	3/4
5	0	0	0	3/8	1	3/8	0	3/4



Avoiding charge sharing

- Make sure that for every input combination there is a path from the power supply to the output.

Manufacturing testing

- Errors are introduced during manufacturing.
- Testing verifies that chip corresponds to design.
- Varieties of testing:
 - functional testing;
 - performance testing (binning chips by speed).
- Testing also weeds out infant mortality.

Testing and faults

- Fault model:
 - possible locations of faults;
 - I/O behavior produced by the fault.
- Good news: if we have a fault model, we can test the network for every possible instantiation of that type of fault.
- Bad news: it is difficult to enumerate all types of manufacturing faults.

Stuck-at-0/1 faults

- Stuck-at-0/1: logic gate output is always stuck at 0 or 1, independent of input values.
- Correspondence to manufacturing defects depends on logic family.
- Experiments show that 100% stuck-at-0/1 fault coverage corresponds to high overall fault coverage.

Testing procedure

- Testing procedure:
 - set gate inputs;
 - observe gate output;
 - compare fault-free and observed gate output.
- Test vector: set of gate inputs (and outputs) applied to a system.

Stuck-at faults in gates

a	b	OK	SA0	SA1
0	0	1	0	1
0	1	1	0	1
1	0	1	0	1
1	1	0	0	1

NAND

a	b	OK	SA0	SA1
0	0	1	0	1
0	1	0	0	1
1	0	0	0	1
1	1	0	0	1

NOR

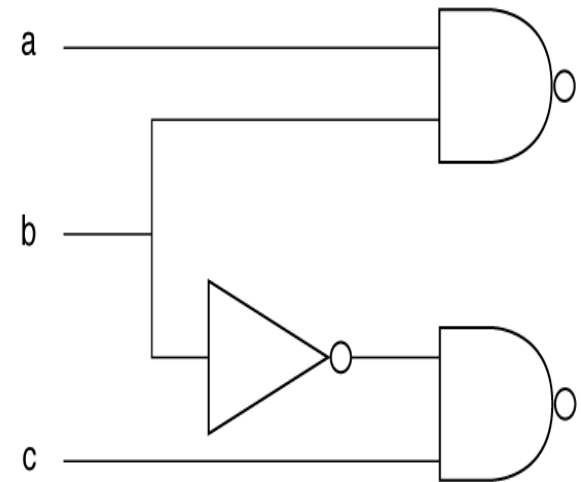
- Three ways to test NAND for stuck-at-0; only one way to test it for stuck-at-1.
- Three ways to test NOR for stuck-at-1; only one way to test it for stuck-at-0.

Testing combinational networks

- 100% coverage: test every gate for
 - stuck-at-0;
 - stuck-at-1.
- Assume that there is only one faulty gate per network.
- Most networks require more than one test vector to test all gates.

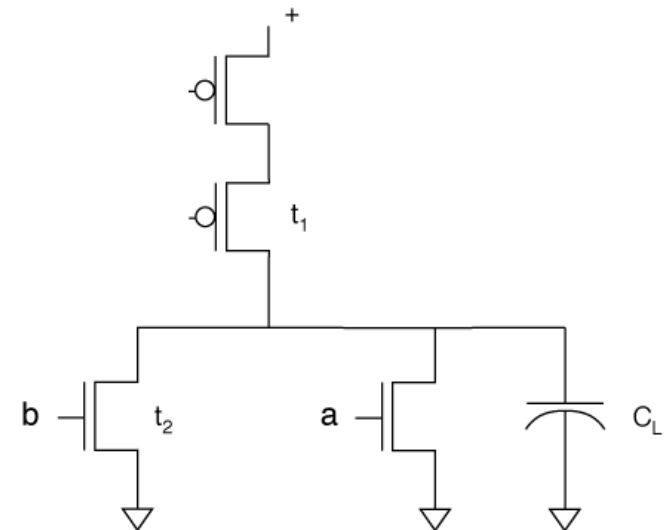
Multiple test example

- Can test both NANDs for stuck-at-0 simultaneously ($abc = 000$).
- Cannot test both NANDs for stuck-at-1 simultaneously due to inverter. Must use two vectors.
- Must also test inverter.



Stuck-open/closed model

- Models transistors always on/off.
- If t_2 is stuck open (switch cannot be closed), there can be no path from V_{SS} to output capacitance.
- Testing requires two cycles:
 - must charge capacitor;
 - try to operate t_2 to see if capacitor can be discharged.



Delay fault

- Delay falls outside acceptable limits:
 - gate delay fault assumes that all delays are lumped into one gate;
 - path delay fault models delay problems along path through network.
- Delay problems reduce yield:
 - performance problems;
 - functional problems in some types of circuits.

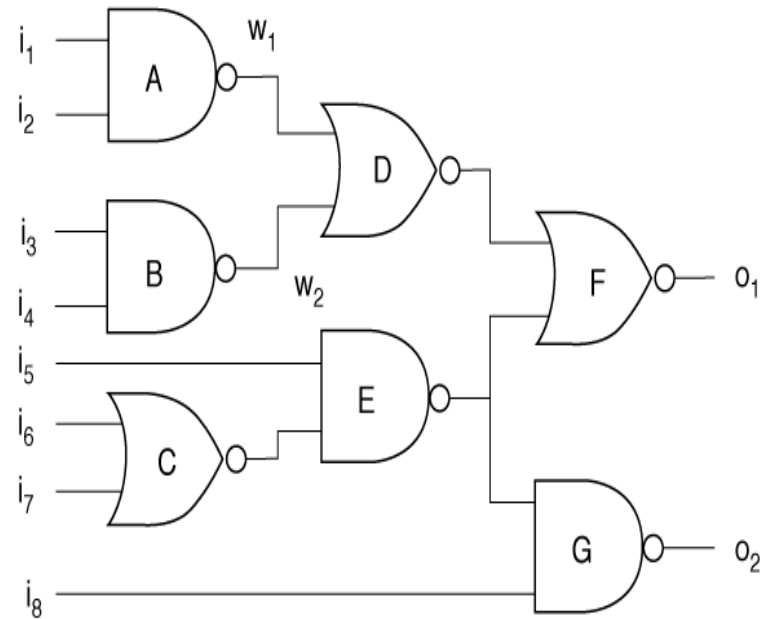
Combinational network testing

Two parts of testing:

- controlling the inputs of (possibly interior) gates;
- observing the outputs of (possibly interior) gates.

Combinational testing example

- Goal: test gate D for stuck-at-0 fault.
- First step: justify 0 values on gate inputs.
- Work backward from gate to primary inputs:
 - $w_1 = 0$ (A output = 0);
 - $i_1 = i_2 = 1$.

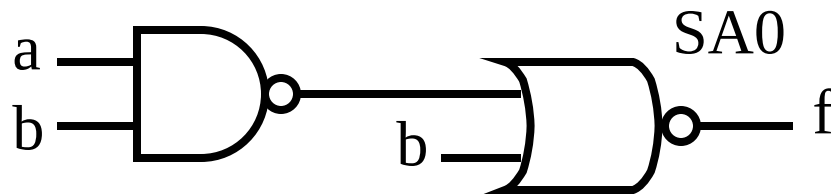


Testing procedure, cont'd

- Observe the fault at a primary output:
 - o1 gives different values if D is true/faulty.
- Work forward and backward:
 - Gate F's other input must be 0 to detect true/fault.
 - Justify 0 at E's output.
- In general, may have to propagate fault through multiple levels of logic to primary outputs.

Fault masking

- Redundant logic can mask faults:



- Testing NOR for SA0 requires setting both inputs to 0.
- Network topology ensures that one NOR input will always be 1.
- Function reduces to 0:
 - $f = [(ab)' + b]' = [a' + b' + b]' = 0.$

Redundancies and testing

- Redundant logic cannot be controlled.
- Observations requiring control of redundant logic may not be possible.
- Redundant logic should be minimized to eliminate redundancy. Redundancies can introduce delay faults and other problems.