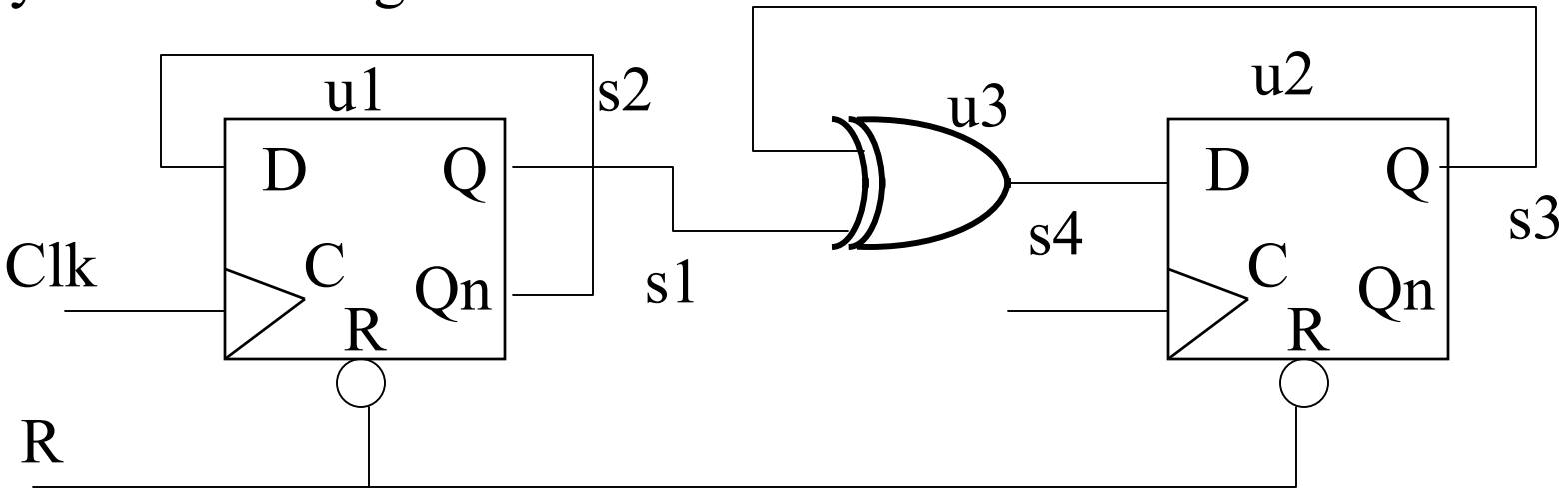
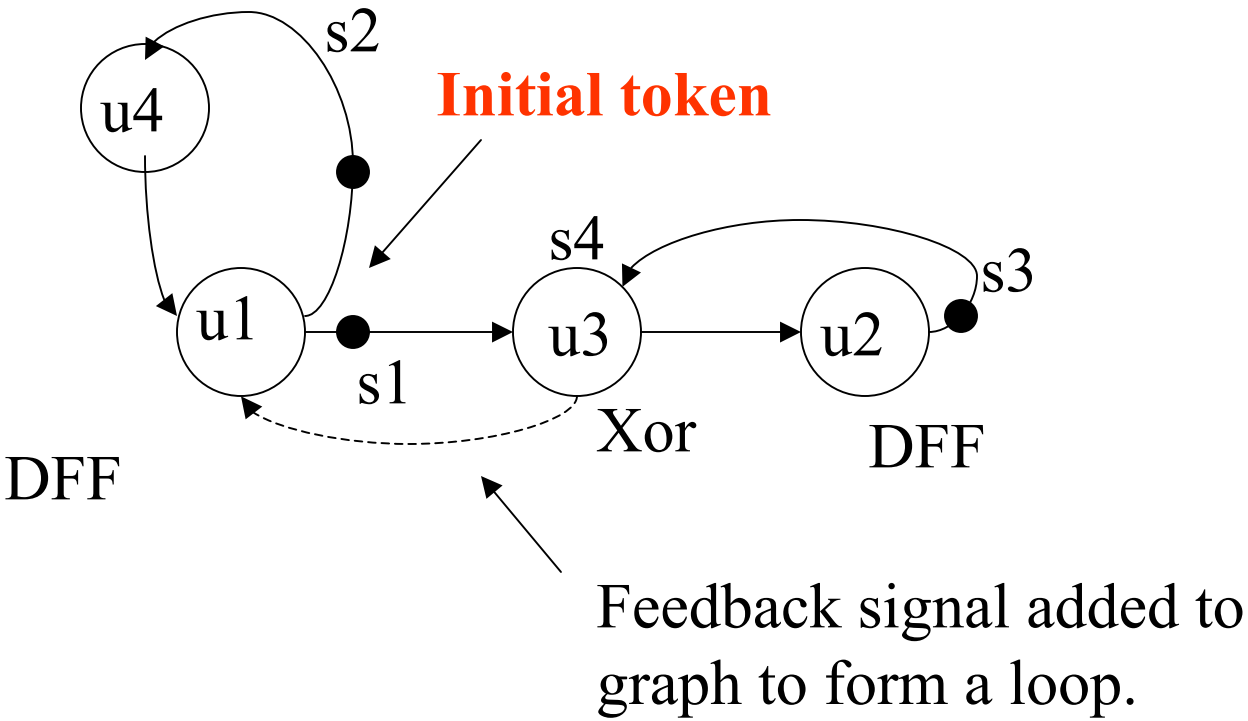
Verilog Transistor Level Simulation Assignment

Implement the logic network below as a dual-rail, dynamic, asynchronous logic circuit.



This is a two-bit counter; after reset simply counts up.

As a Marked Graph



All DFFs fanout signals have initial tokens on them.

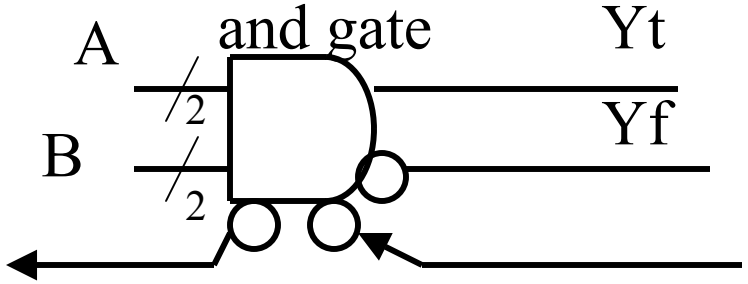
All signals must be part of a loop with only one token.

Mapping to a Delay Insensitive, Asynchronous Circuit

- Delay insensitive: circuit works regardless of wire/gate delays in circuit.
  - Have to detect the arrival of a signal at a gate
  - Typically, the receiving gate has to send an acknowledgement the signal arrival to the originating gate
  - Dual Rail encoding can be used to allow detection of signal arrival.
  - Acknowledgement signal (feedback) is single rail
- Asynchronous – no global clock signal
  - Global reset allowed, only asserted after power on

T/F Dual Rail Signaling, 4 Phase

Logic gate produces both true and complement version of output.



Yt = Yf = 0 is the null state. Asserted state is either Yt=1 or Yf = 1.

Event sequence (all signals null or negated)

a. Signals A, B arrive in some order.

b. After all inputs signals have arrived, Signal Y output is asserted (either Yt or Yf goes high).

c. This asserts Le indicating inputs have arrived and gate fired.

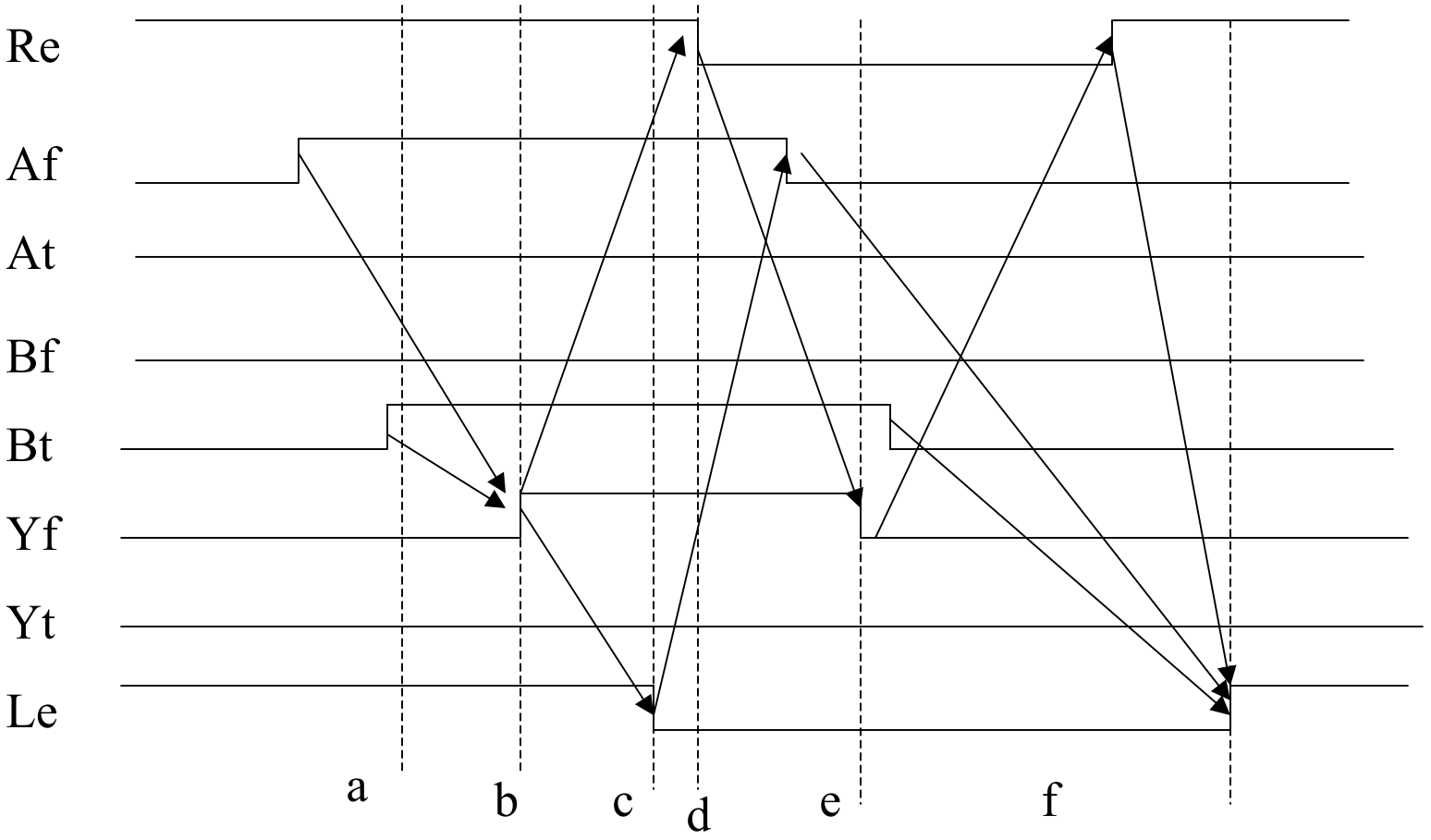
d. At some point, Re is asserted indicating output has arrived at destination gate and has been consumed.

e. Re assertion causes Y output to return to null.

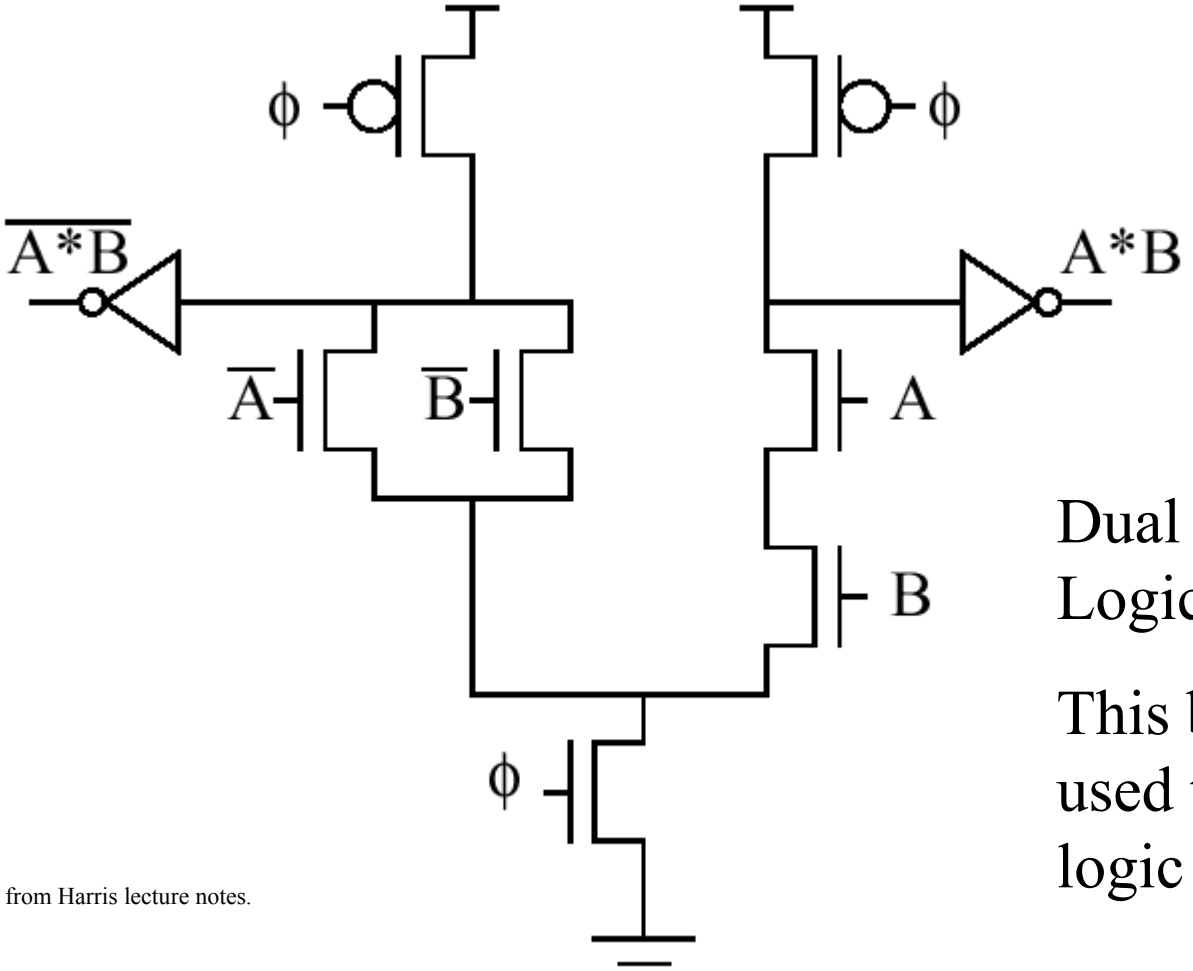
f. After A, B inputs return to null, and Re is negated indicating destination acknowledging Y return to null, then Le is negated.

4-Phase Signaling

All signals are in the negated (null) state prior to computation. During computation, signals are asserted. After computation, signals return to initial states. Re, Le are low true



Dual Rail Dynamic Logic

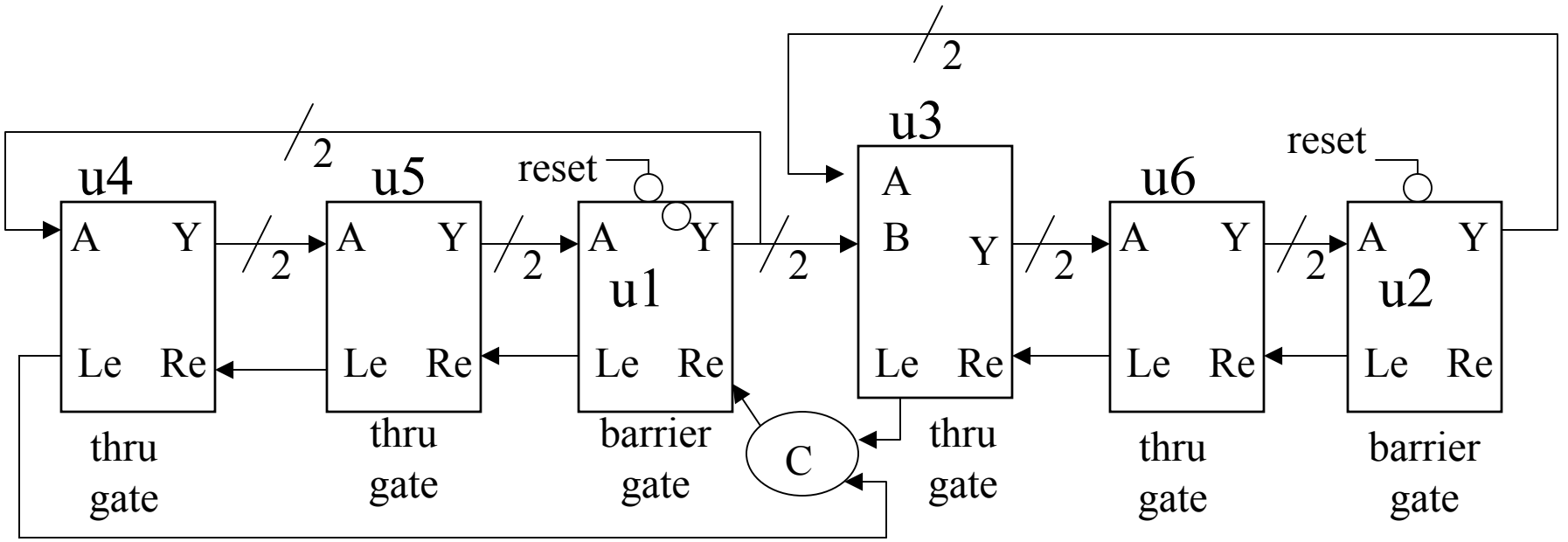


Pic from Harris lecture notes.

Dual Rail Domino Logic

This basic approach is used to implement the logic for computation.

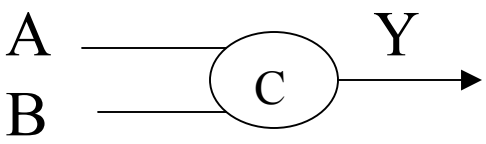
Asynchronous 2-bit Counter



barrier gate – DFFs in original design.
thru gate – combinational logic gates in original design.

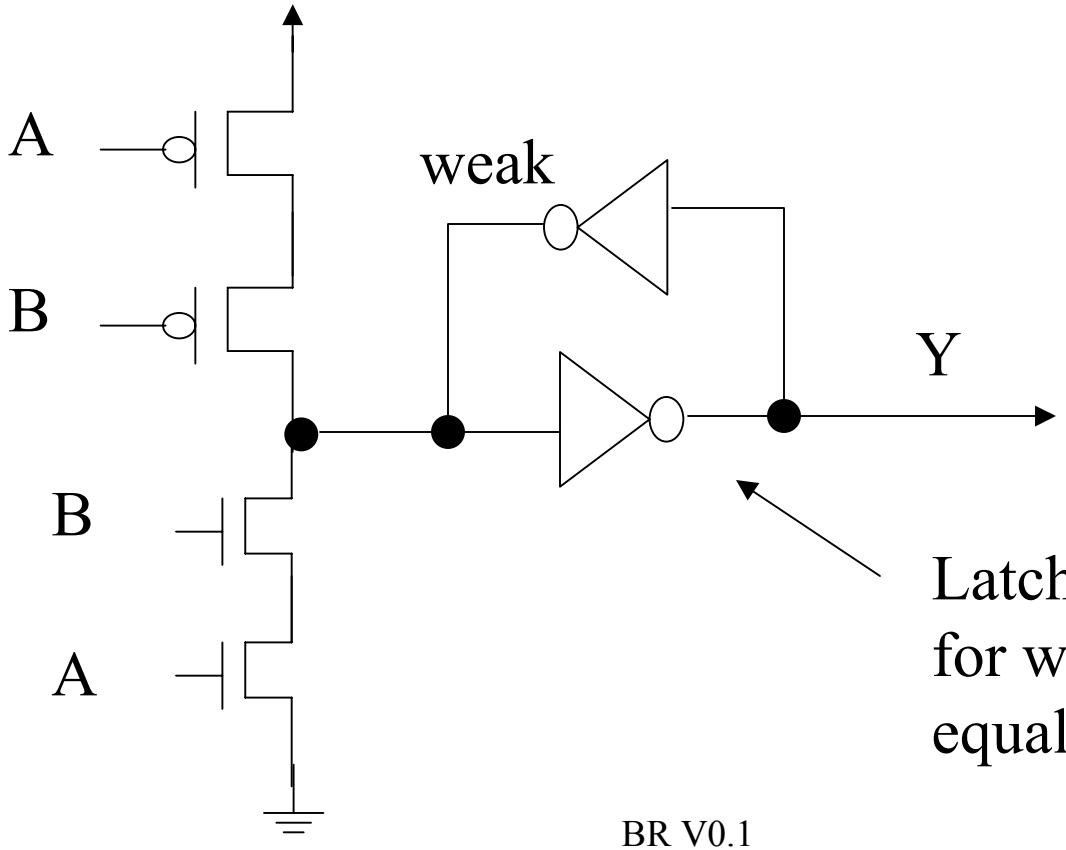
U4, U5, U6 are just buffer functions added to satisfy handshaking constraint that any loop must have THREE gates in it because of four phase handshaking.

Muller C-Element: Used to combine multiple acknowledgements into a single acknowledgements



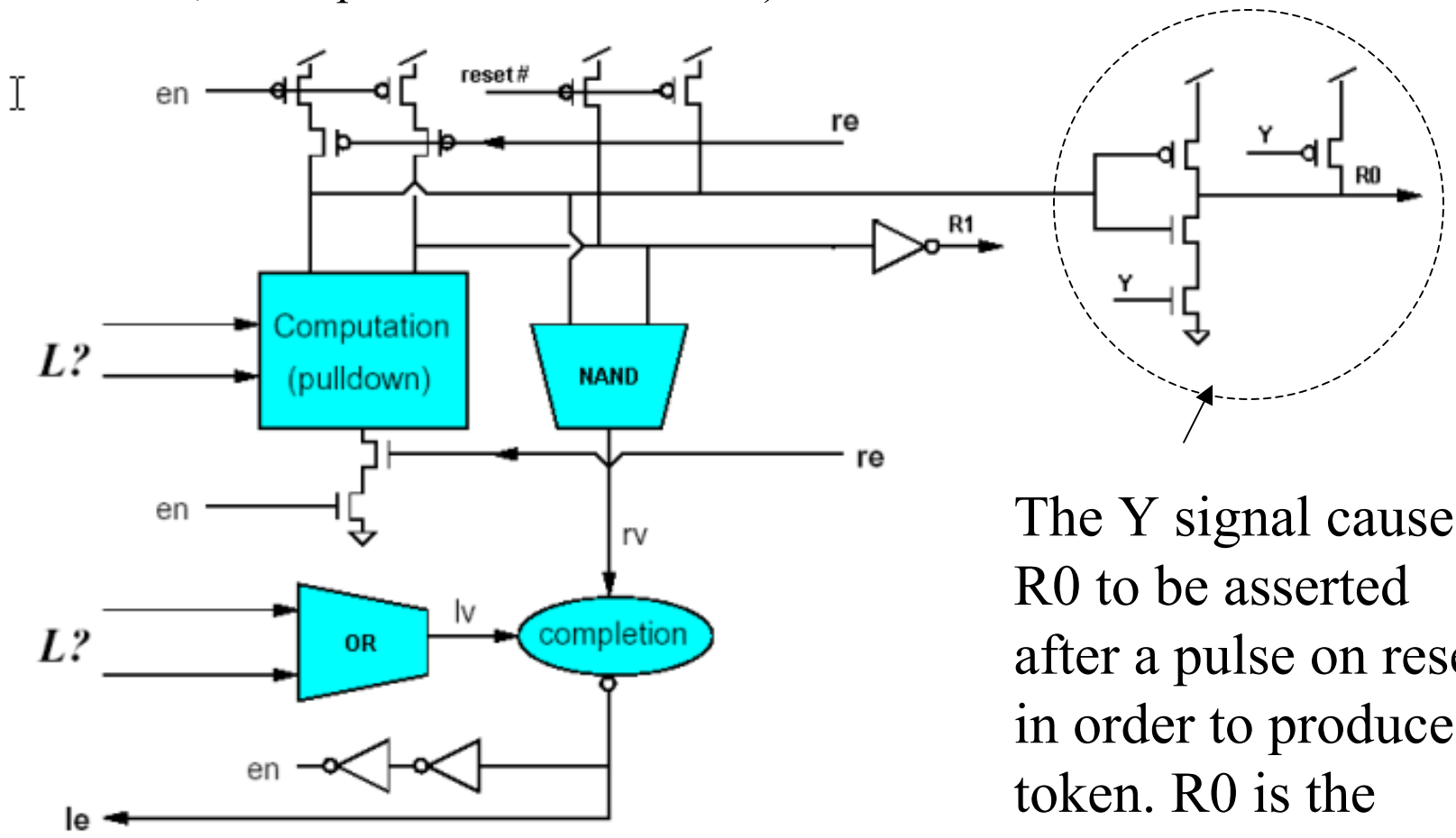
Output goes low only when all inputs are low.

Output goes high only when all inputs are high



Barrier Gate

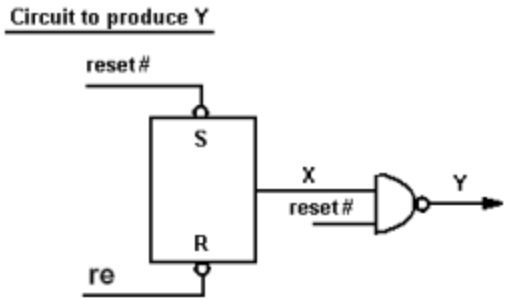
Must be able to produce initial token on output after reset (this means that after reset, an output has to be asserted!)

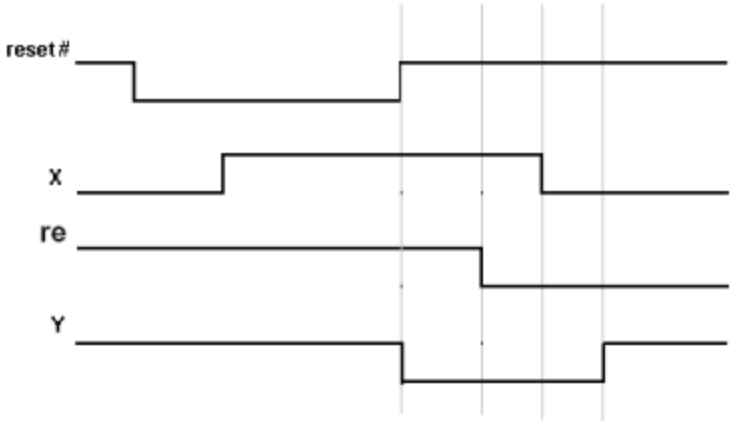


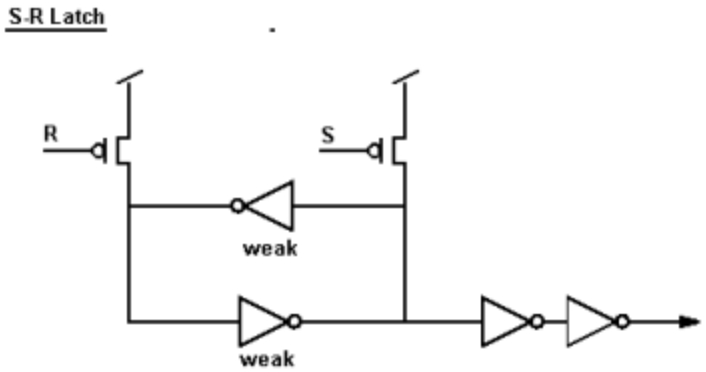
The Y signal causes R0 to be asserted after a pulse on reset in order to produce a token. R0 is the 'false' output.

Basic QDI gate design from Cal Tech

Token Generation



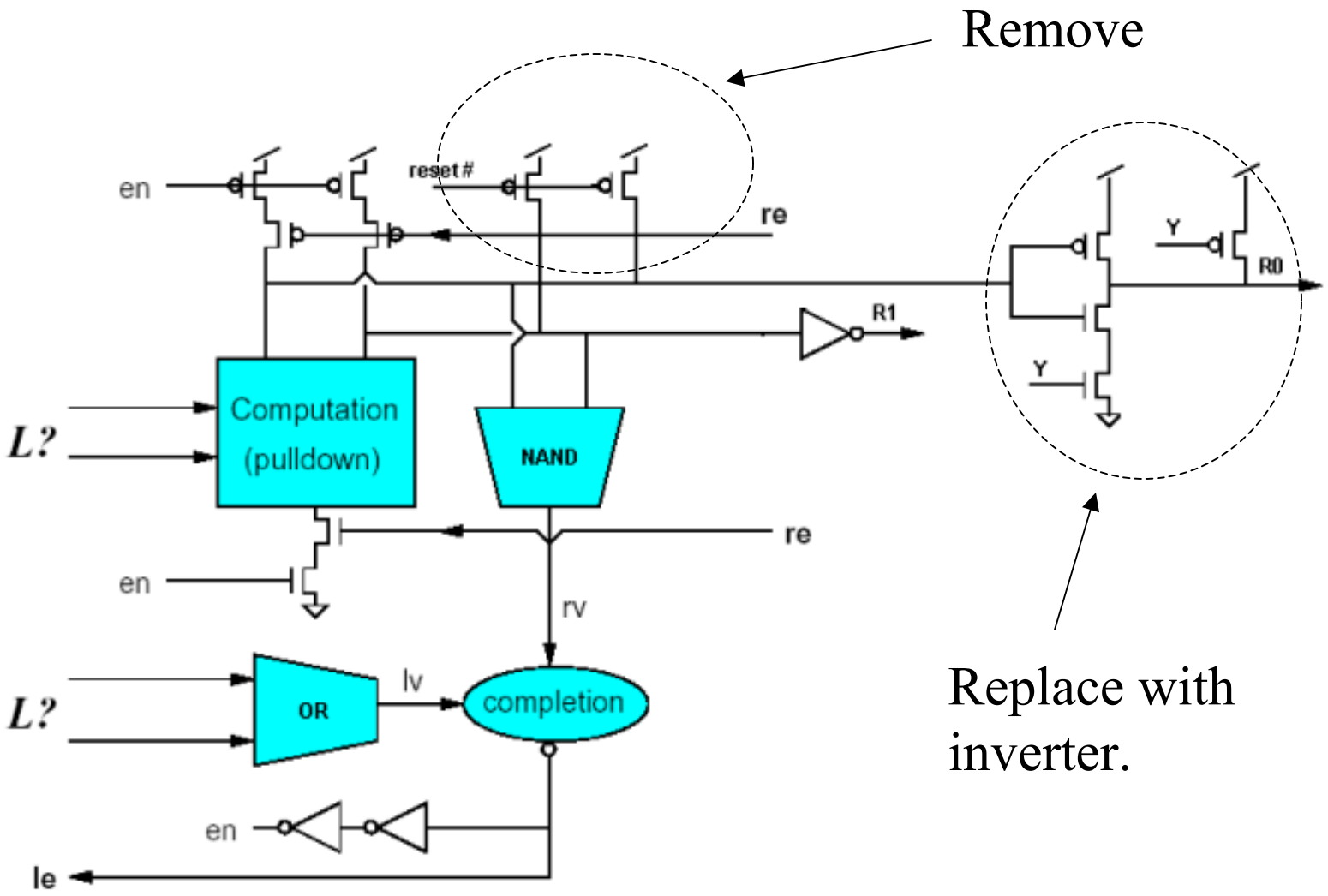




Circuit by
Lokesh Shivakumaraiah

Assertion, then negation of reset causes Y to be asserted, which puts token on gate output. Once Re is asserted acknowledging the token, the Y signal is negated. This 'kicks' the circuit into operation.

Through Gate



Assignment

- Write Verilog transistor level models for barrier, thru, and C-elements needed for 2-bit counter
- Write netlist and testbench that shows functionality.
- Output should be printed to console that shows reset assertion, followed by a trace of the counter output bits changing value.
- Top level testbench module must be called ‘tb’.
- Library name must be called ‘veri_async’.
- Use the makefile approach that has been used for previous labs.