



ECE 3060

VLSI and Advanced Digital Design

Takeaways

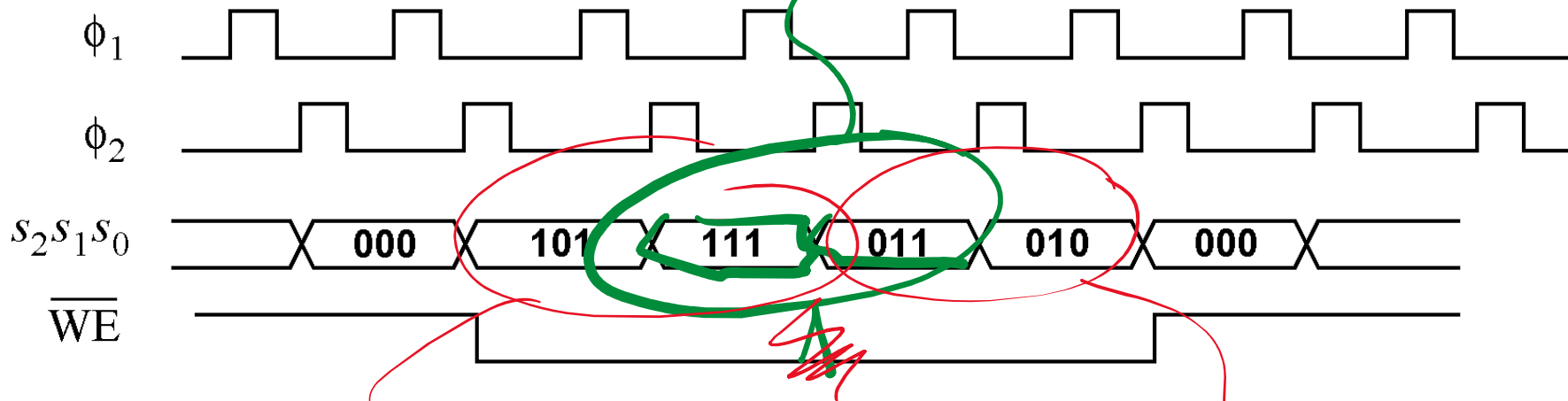
- (1) Standard digital synthesis tools generate logic with hazards
- (2) I know approaches to syn. + logic design to gen. hazard-free ^{comb.} logic ~~gate~~ glitch-free

Motivation

Suppose we wish a logic signal (say a DRAM \overline{WE} signal) to be glitch free *for 4 clock cycles*

Suppose the signal is generated by four states of a state machine as shown

$$s_2 s_1 s_0 + s_2 s_1 \overline{s_0} = s_2 s_0$$

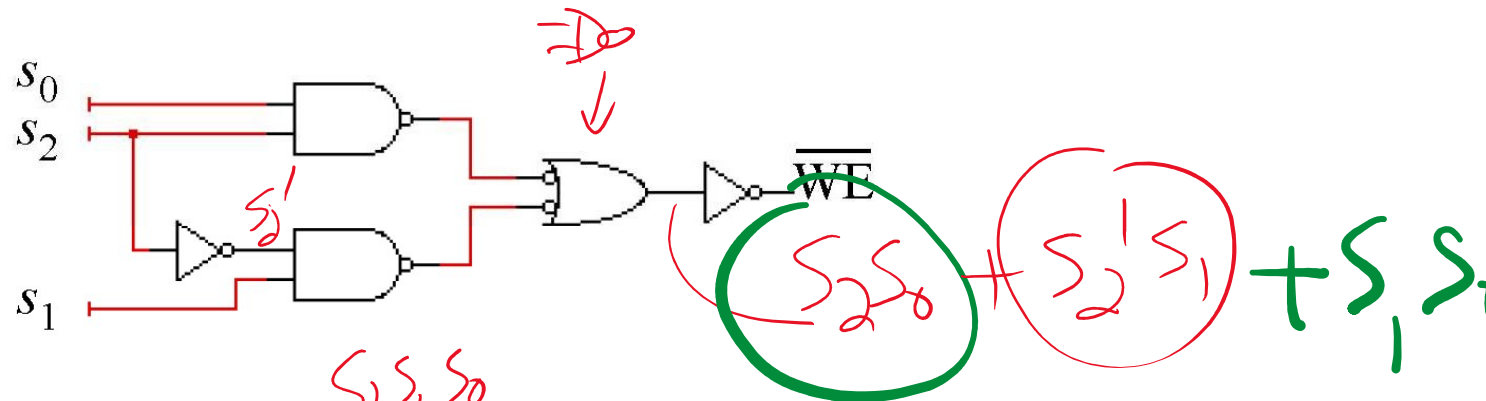


Minimization yields $\overline{WE} = (s_2 s_0 + \overline{s_2} s_1)$

$s_2 s_0$

$\overline{s_2} s_1$

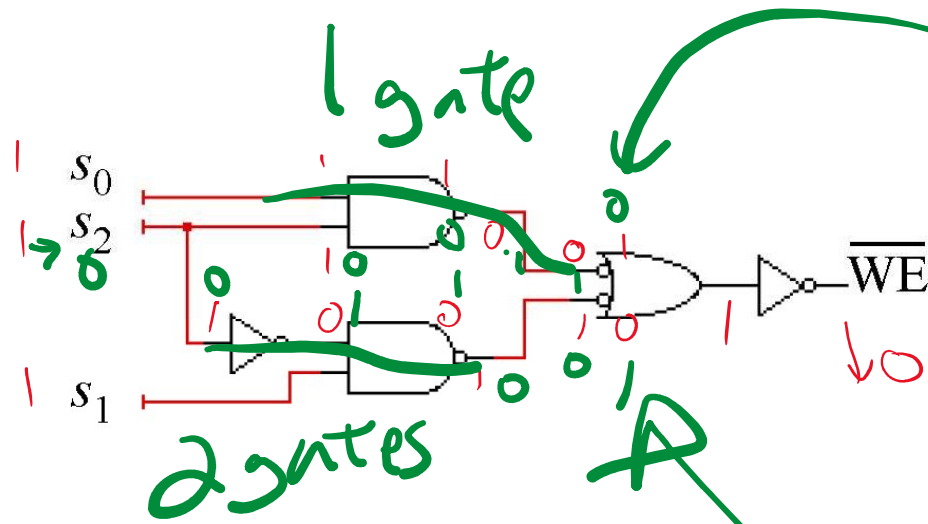
Implementation



Consider transition $S = 111$ to $S = 011$

This cause of this glitch is a static 1 hazard in the implementation.

A static 1 hazard is present whenever adjacent 1 nodes on the cube are not covered by the same implicant



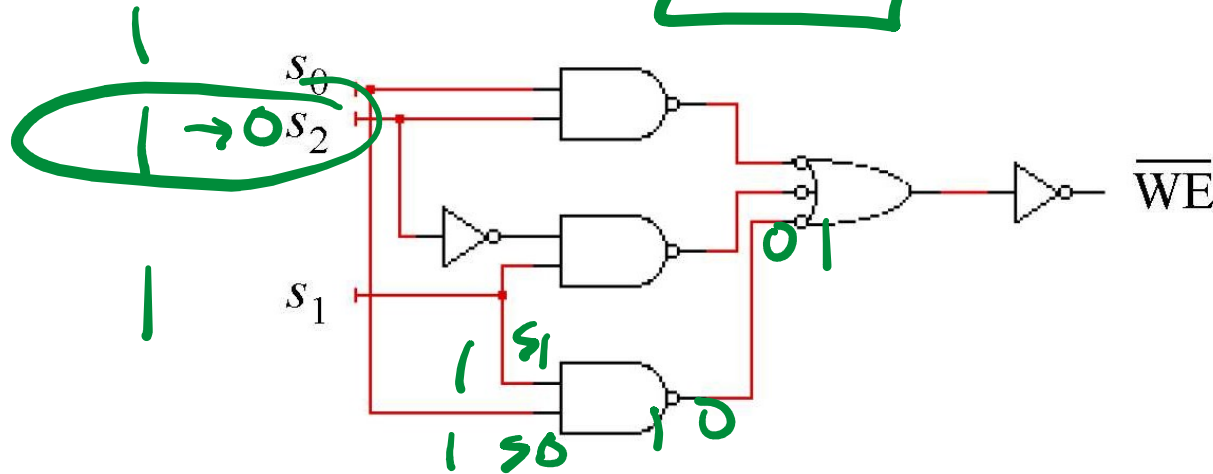
when does arrive compared to

if 0 arrives first, then have 0 0 0

Solution

Add the consensus term

$$s_1 s_0$$



An analogous case exists for a static 0 hazard in a product of sums expression

Dynamic hazards are much more difficult to analyze, and occur in multi-level logic and where multiple inputs may change concurrently.