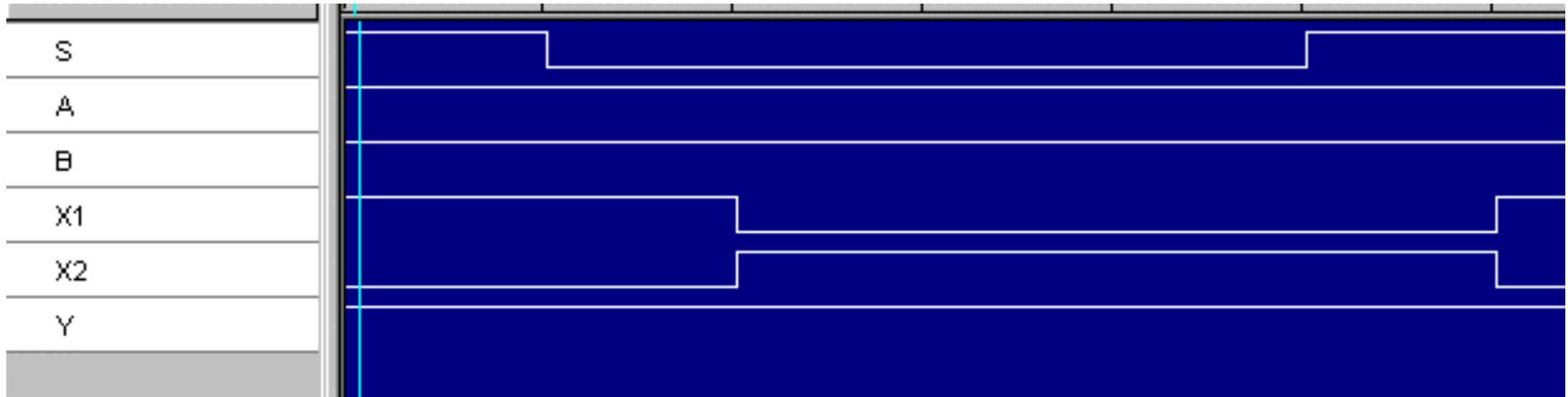
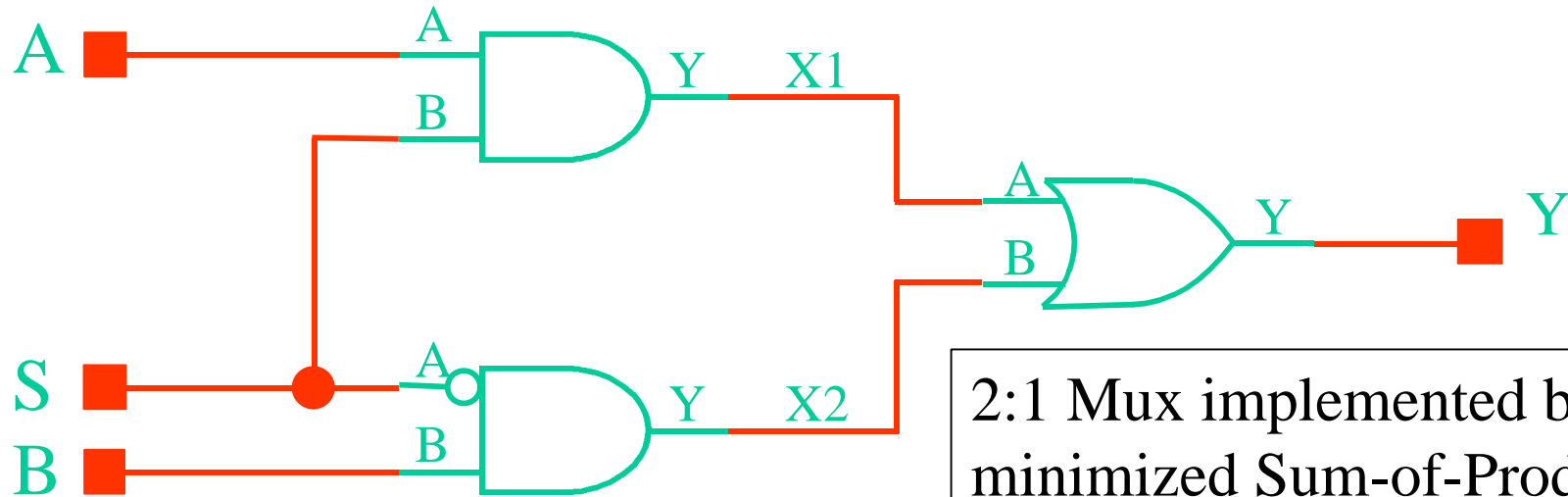


# Static Hazards

# Definitions

- If the change of a *single* variable causes a momentary change in other variables, which should not occur, then a *static hazard* is said to exist.
- If, after switching an input, the output has multiple transitions for a short time, then a *dynamic hazard* exists. For example
  - S/B:    0 → 1
  - IS:     0 → 1 → 0 → 1

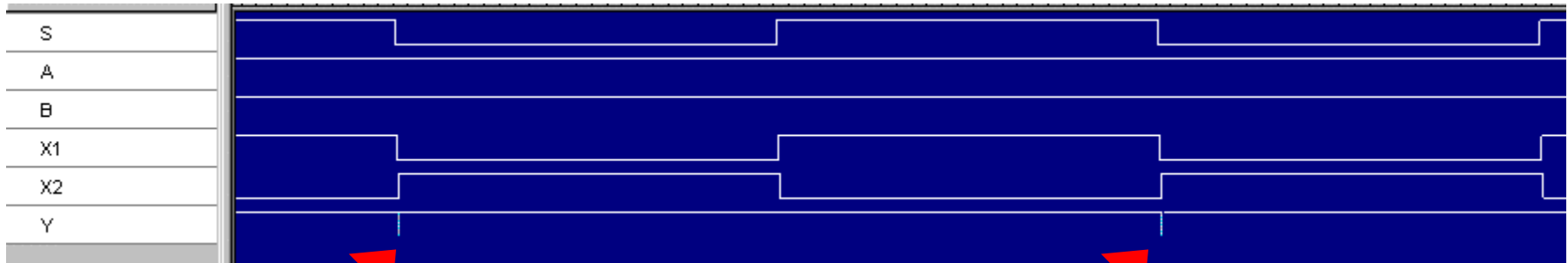
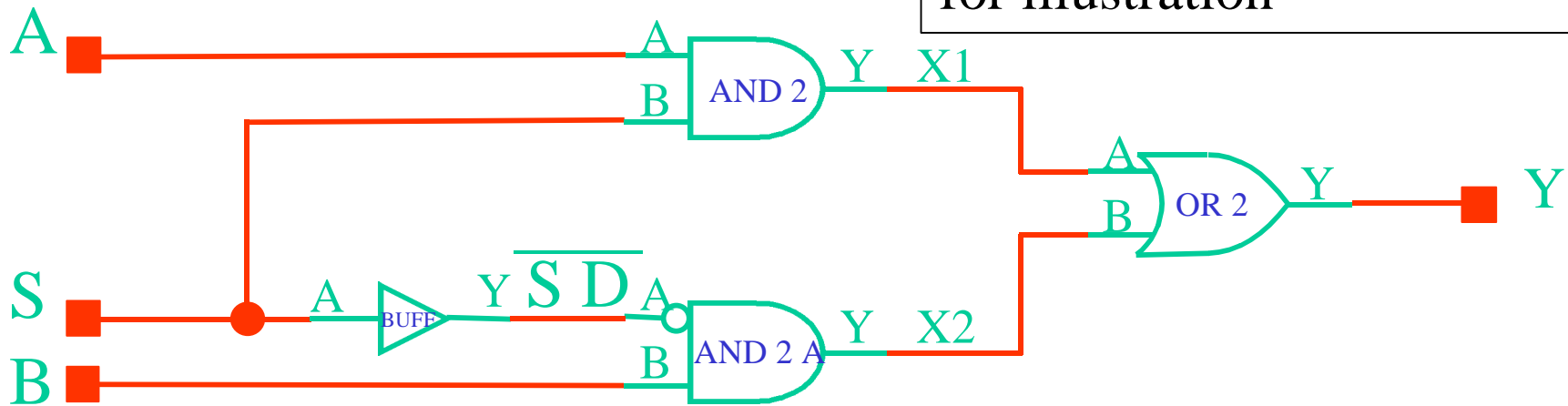
# Static Hazard



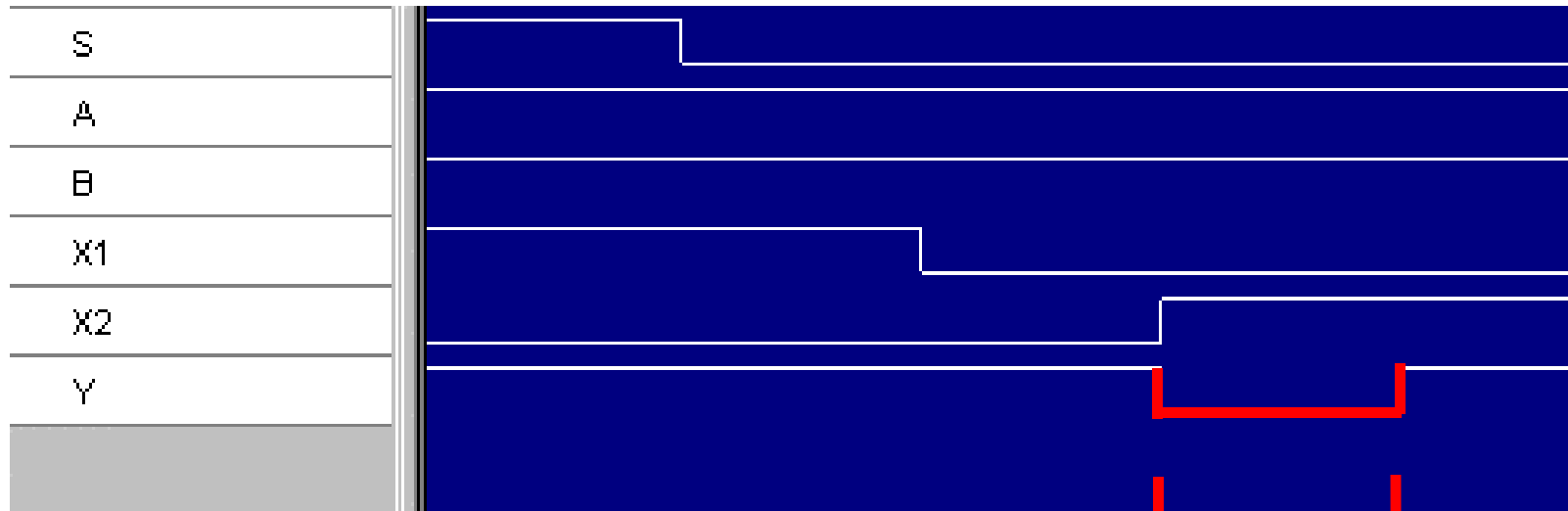
Idealized matched delays

# Static Hazard

In real circuits, delays don't exactly match; Added delay for illustration



# Static Hazard



We now have a "glitch."

Same waveform, zoomed in.

# Static Hazard

|     |  | A B |     |     |     |
|-----|--|-----|-----|-----|-----|
|     |  | 0 0 | 0 1 | 1 1 | 1 0 |
| S=0 |  | 0   | 1   | 1   | 0   |
| S=1 |  | 0   | 0   | 1   | 1   |

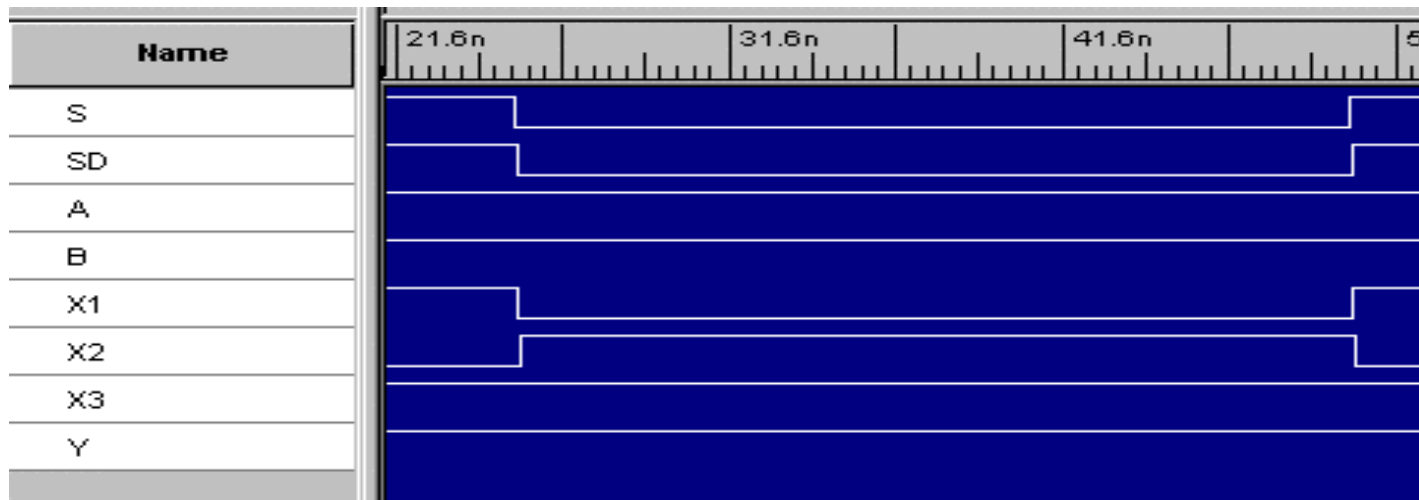
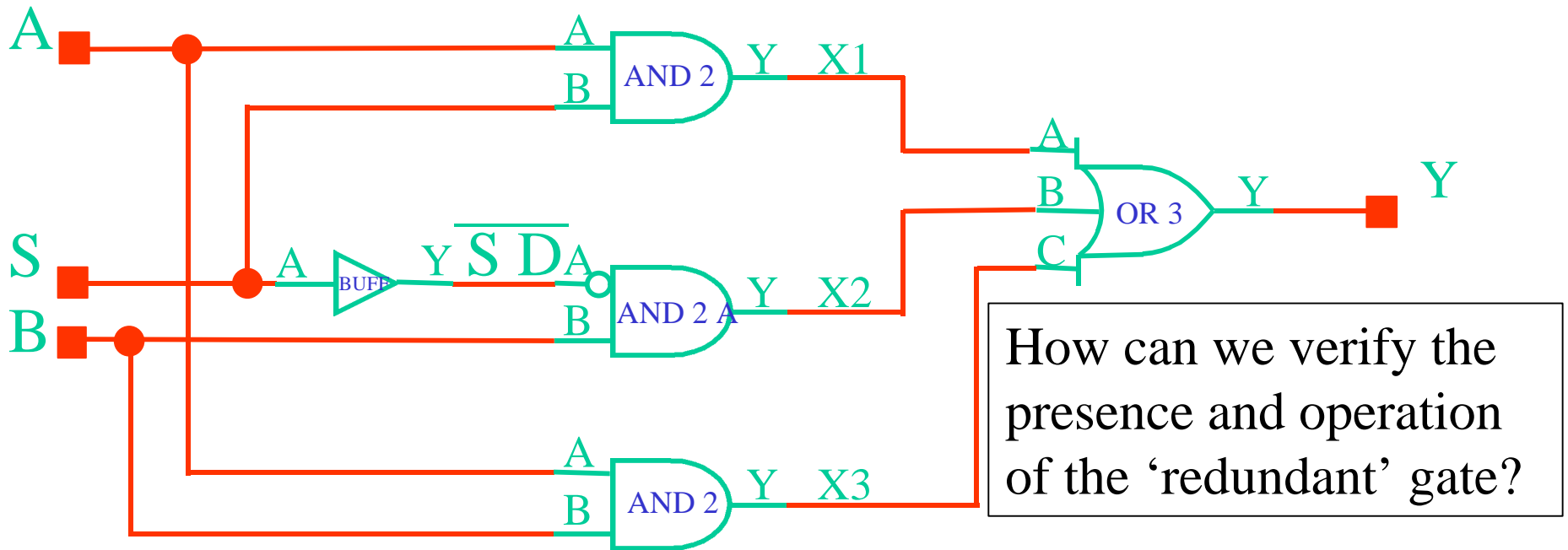
Illustrating the minimized function on a Karnaugh map.  
Only two 2-input AND gates are needed for the product terms

# Static Hazard

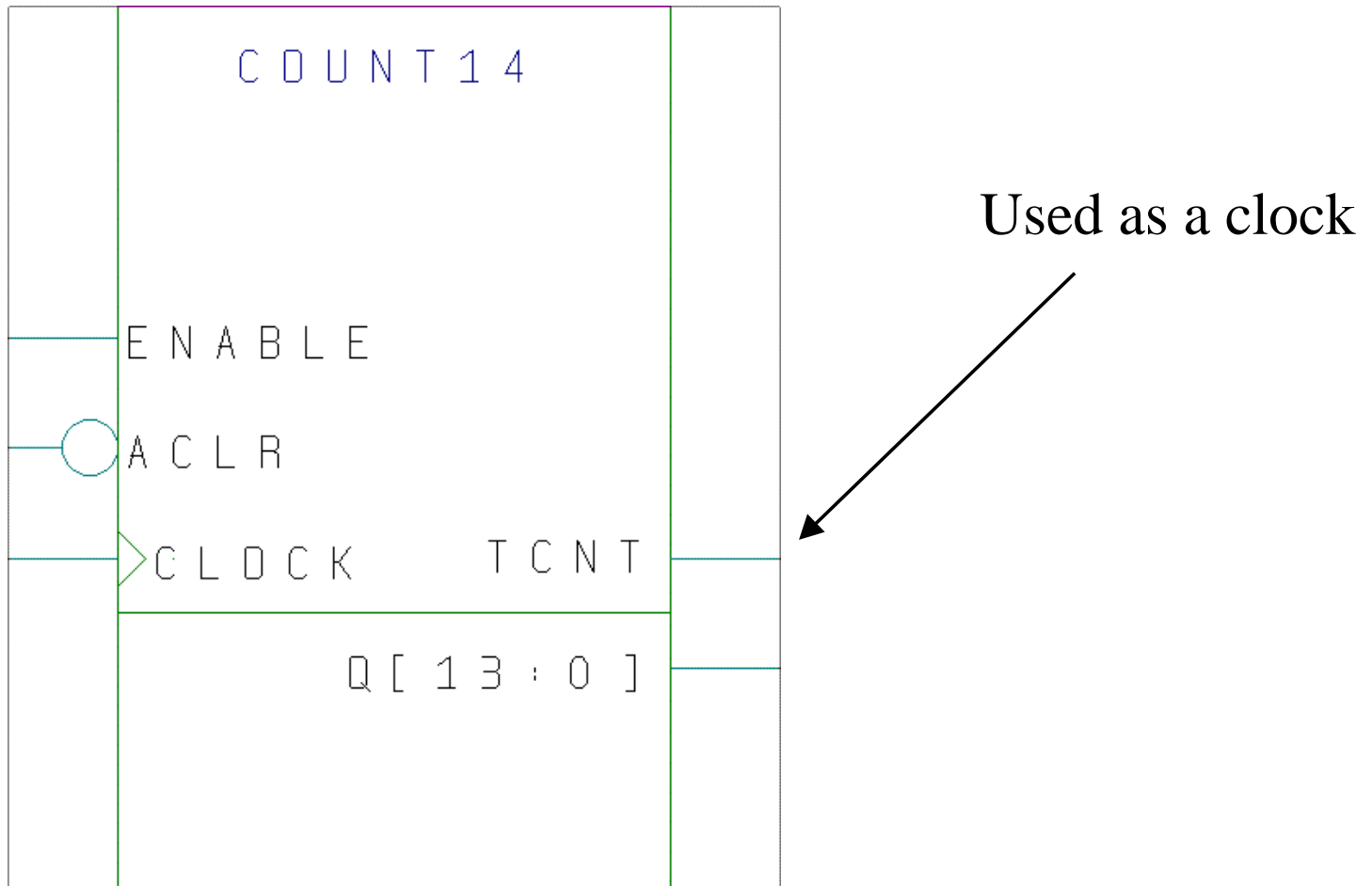
|     |  | A B |     |     |     |
|-----|--|-----|-----|-----|-----|
|     |  | 0 0 | 0 1 | 1 1 | 1 0 |
| S=0 |  | 0   | 1   | 1   | 0   |
| S=1 |  | 0   | 0   | 1   | 1   |

The blue oval shows the redundant term used to cover the transition between product terms.

# Static Hazard

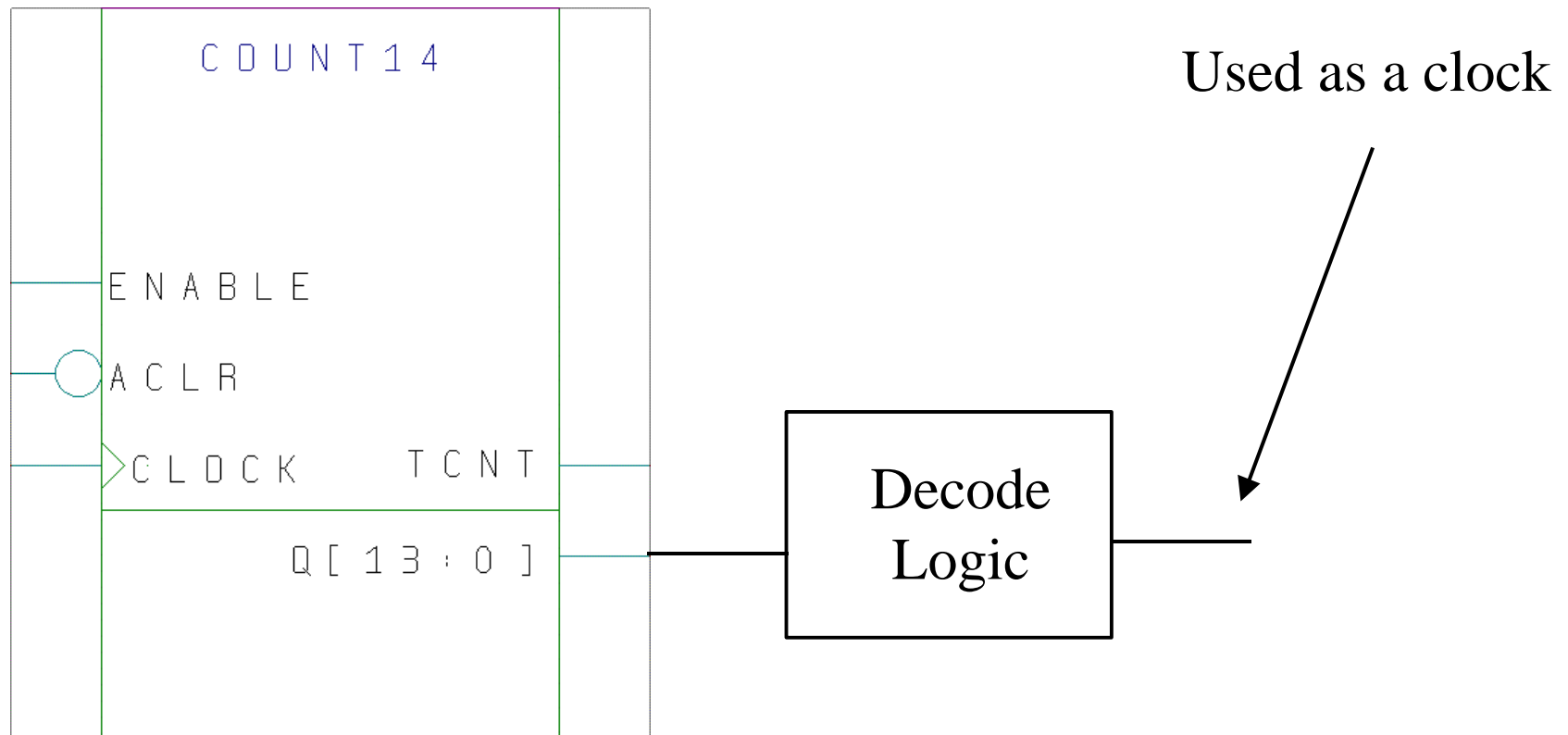


# Asynchronous Decoding High Level

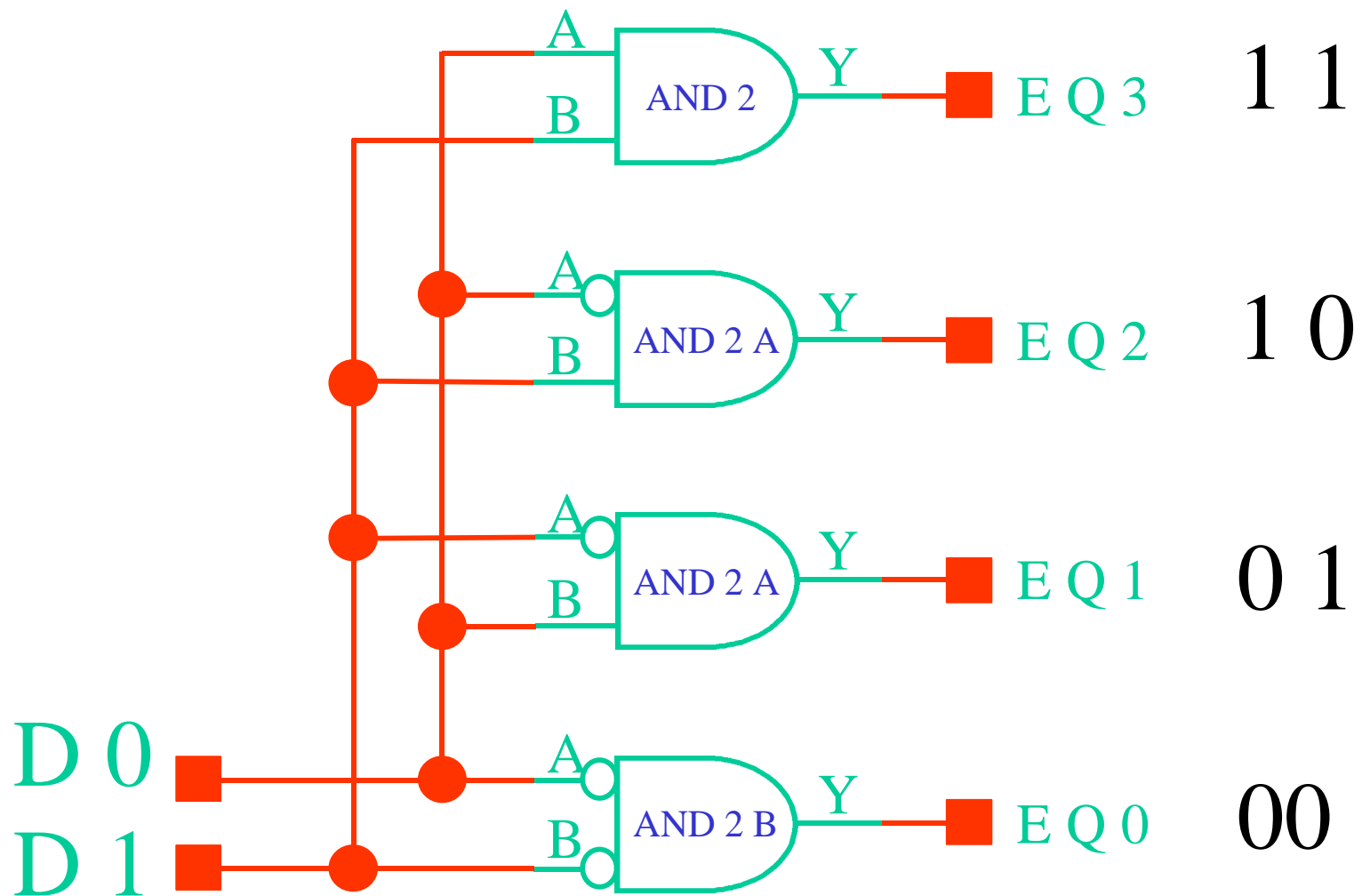


# Asynchronous Decoding

## High Level - Another Form

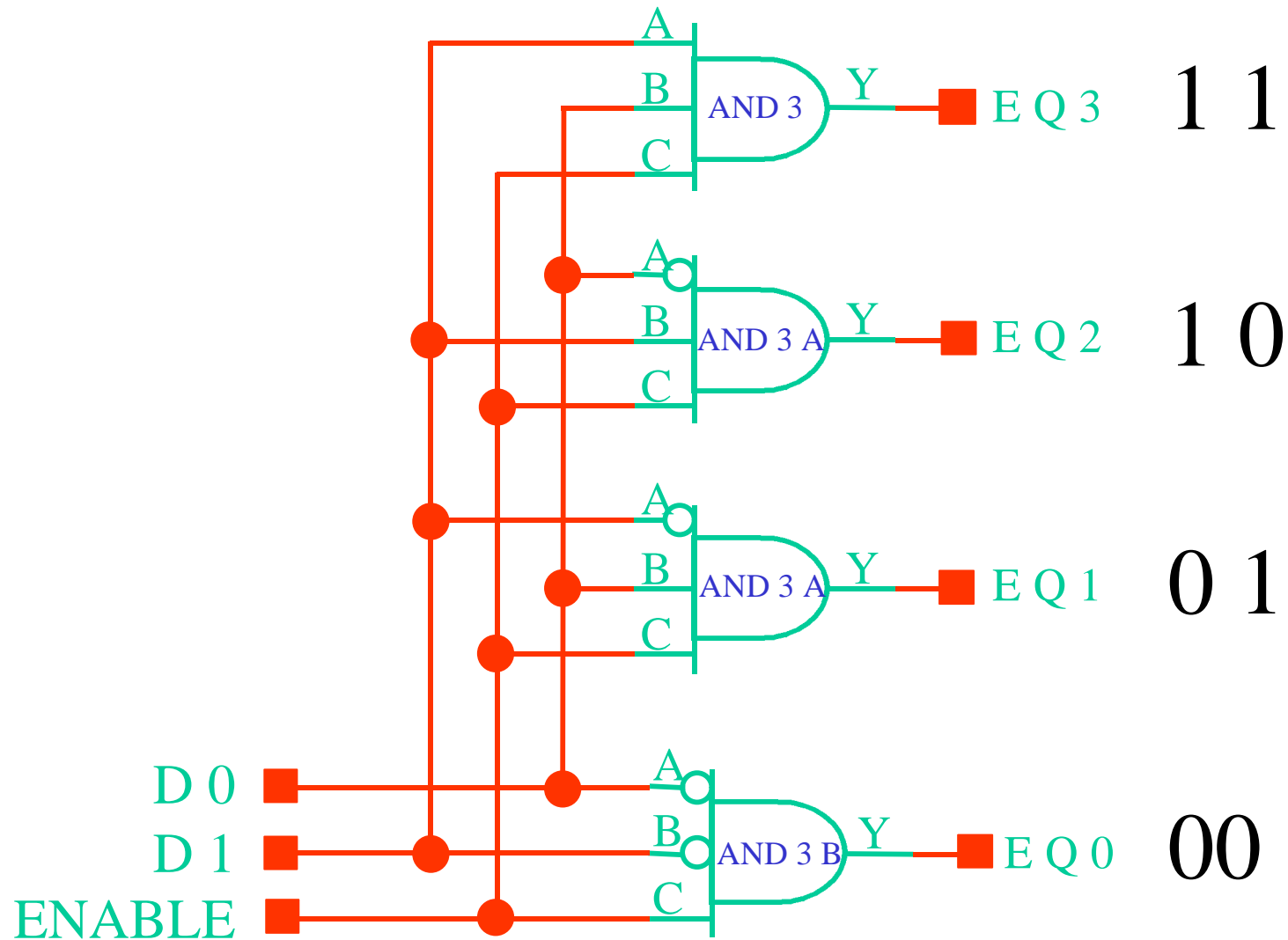


# 2:4 Decoder

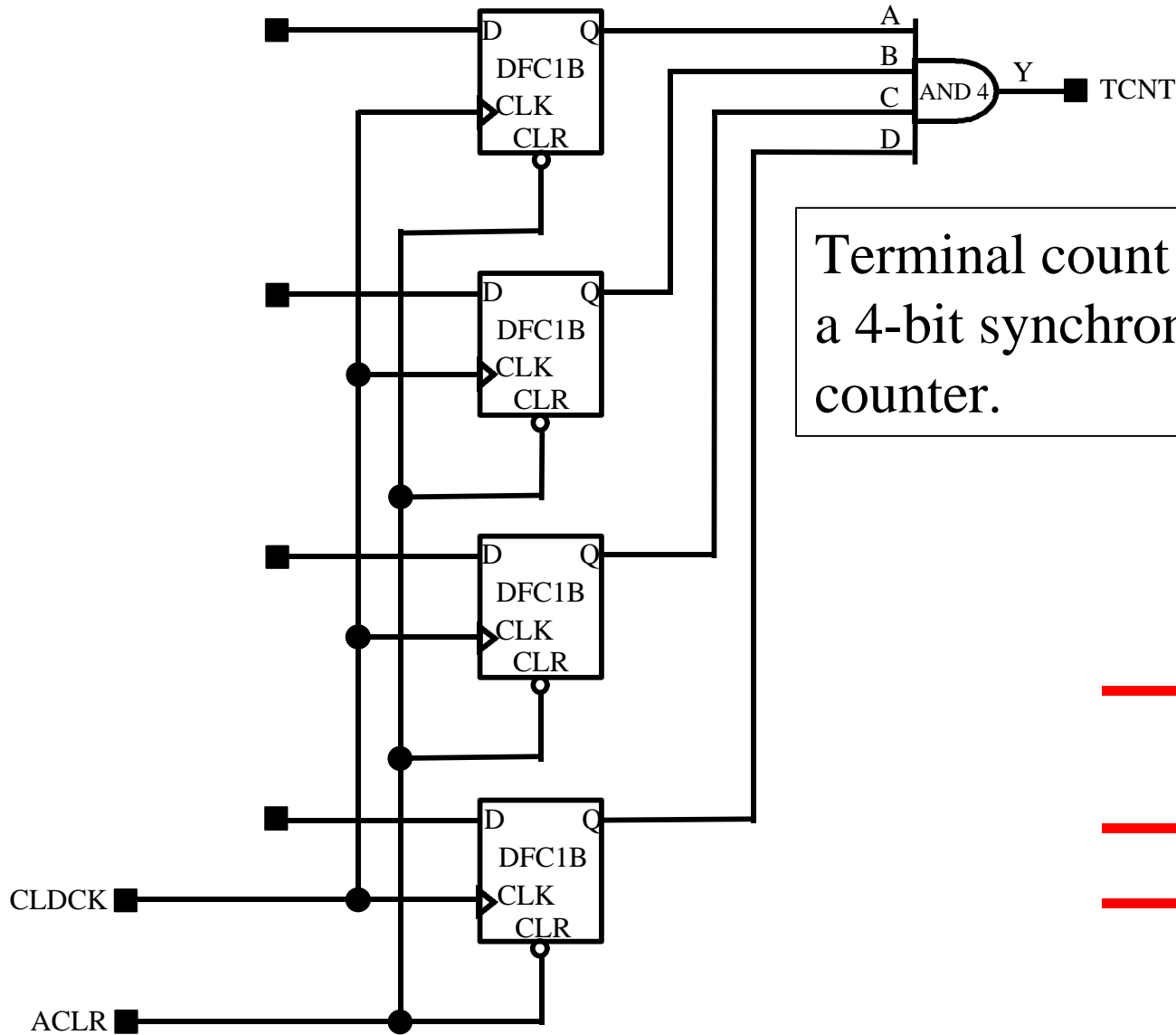


What happens when the inputs goes from 01 to 10?

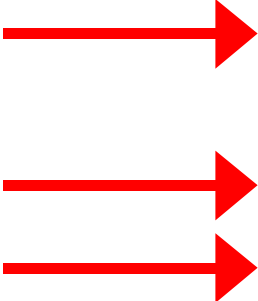
# 2:4 Decoder with Enable



# Implementation Level



Terminal count of a 4-bit synchronous counter.



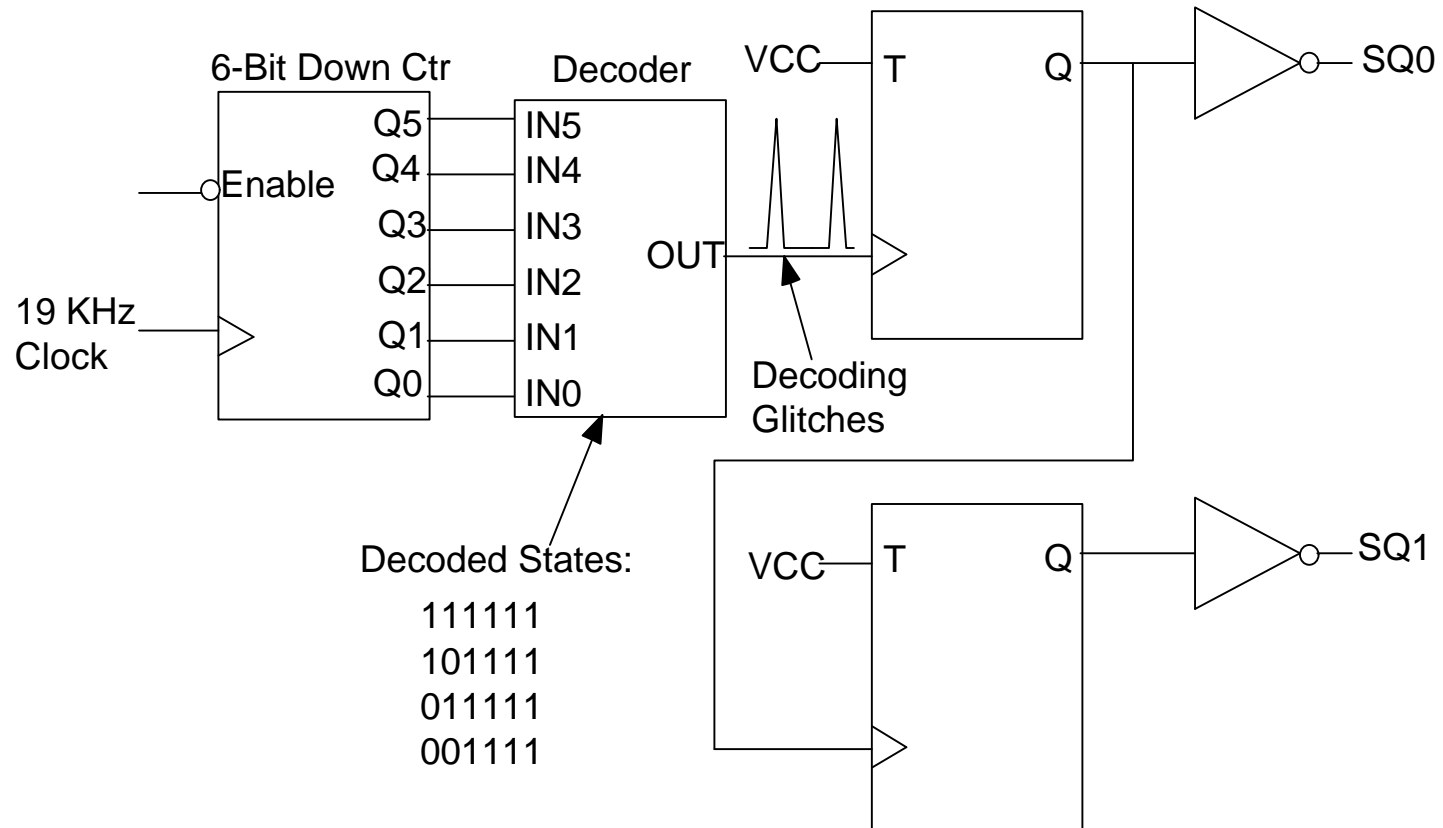
|      |    |
|------|----|
| 0000 | 0  |
| 0001 | 1  |
| 0010 | 2  |
| 0011 | 3  |
| 0100 | 4  |
| 0101 | 5  |
| 0110 | 6  |
| 0111 | 7  |
| 1000 | 8  |
| 1001 | 9  |
| 1010 | 10 |
| 1011 | 11 |
| 1100 | 12 |
| 1101 | 13 |
| 1110 | 14 |
| 1111 | 15 |
| 0000 | 16 |

# Asynchronous Decoding Glitch Generation

→ 0111111111111111  
1000000000000000  
...  
→ 1111111110111111  
1111111111000000

Because of unequal propagation delays, the sequence can momentarily go through state 1111111111111111 generating a glitch.

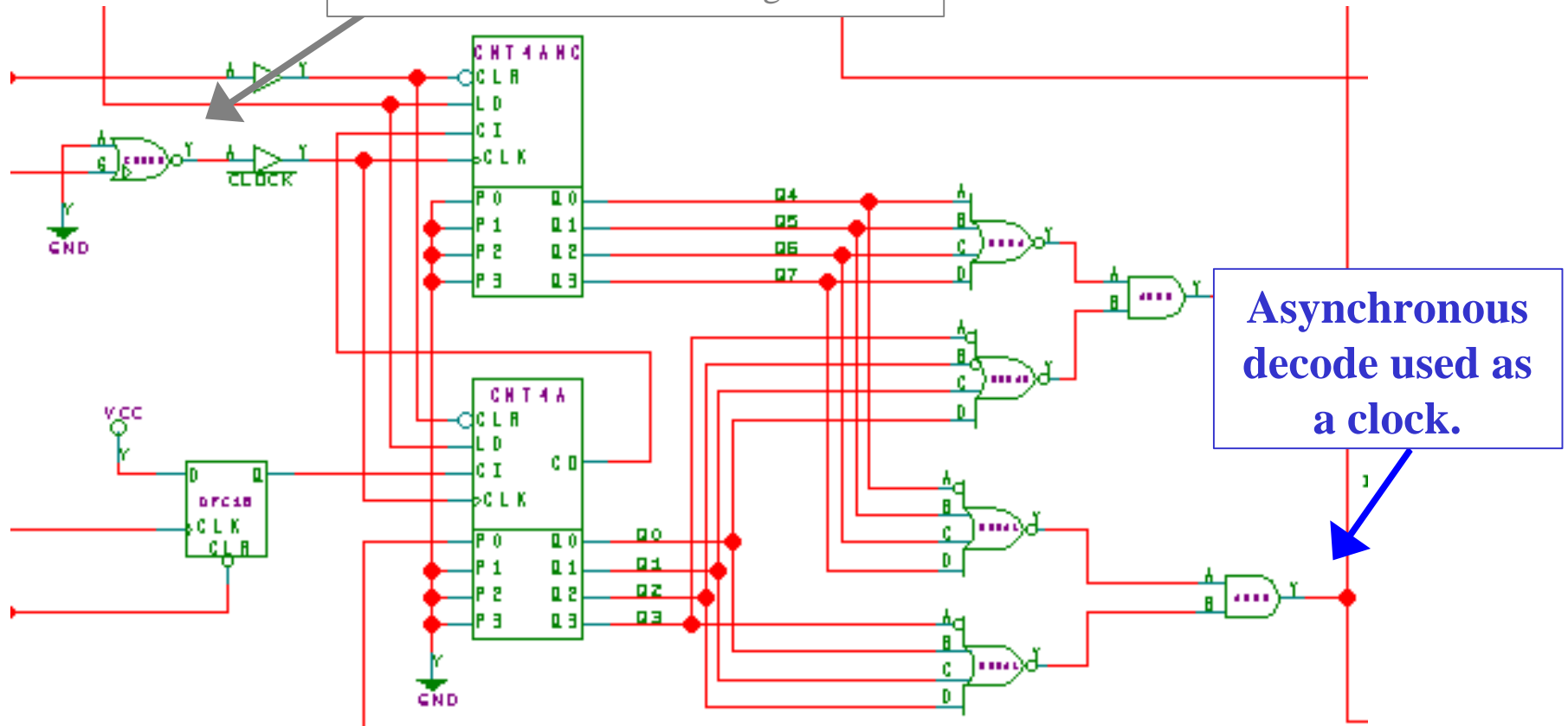
# Decoder Output Used As Clock



From Erickson, MAPLD 2000

# Logic Design

Global Clock converted to local clock and fed to drive shift register.

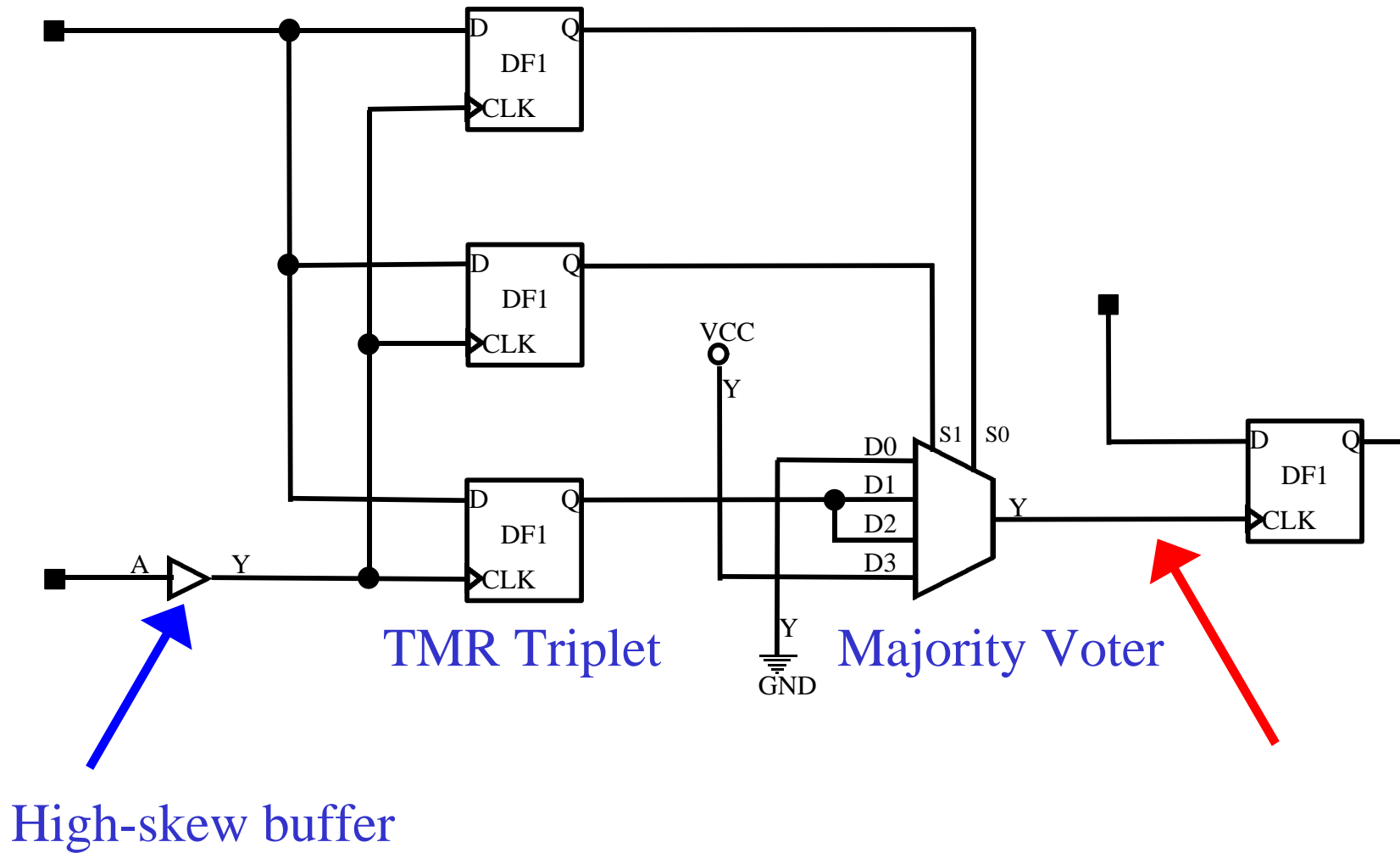


Asynchronous decode used as a clock.

Designer unaware that a parallel asynchronous decode may glitch. Relied on back-annotated logic simulation. This construct appears repeatedly.

# Static Hazard

## Flight Design Example



# Static Hazard

## Flight Design Example

Care is needed when using TMR circuits. First, the output of the voter may be susceptible to a logic hazard “glitch.” This is not a problem if the TMR is feeding the input of another synchronous input. However, the TMR output should never feed asynchronous inputs such as flip-flop clocks, clears, sets, read/write inputs, etc.

“Design Techniques for Radiation-Hardened FPGAs”

Actel Corporation, September 1997

-- based on “SEU Hardening of Field Programmable Gate Arrays (FPGAs) for Space Applications and Device Characterization,” R. Katz, R. Barto, et. al., IEEE Transactions on Nuclear Science, Dec. 1994.

# Dynamic Hazards

We have covered static hazards. There are also dynamic hazards. An example of a dynamic hazard would be when a circuit is supposed to switch as follows:

$$0 \rightarrow 1$$

But instead switches:

$$0 \rightarrow 1 \rightarrow 0 \rightarrow 1$$

Any circuit that is static hazard free is also dynamic hazard free.