Hazards and Glitches

Prof. James L. Frankel
Harvard University

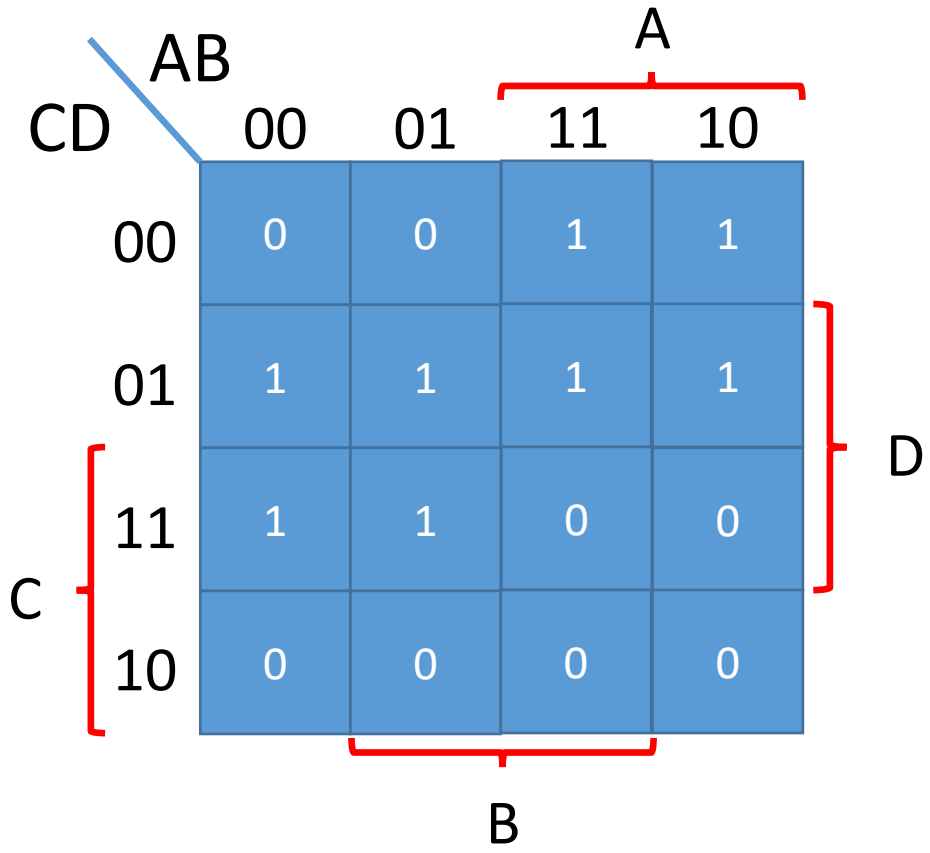
Version of 10:29 AM 2-Dec-2021

Hazards and Glitches

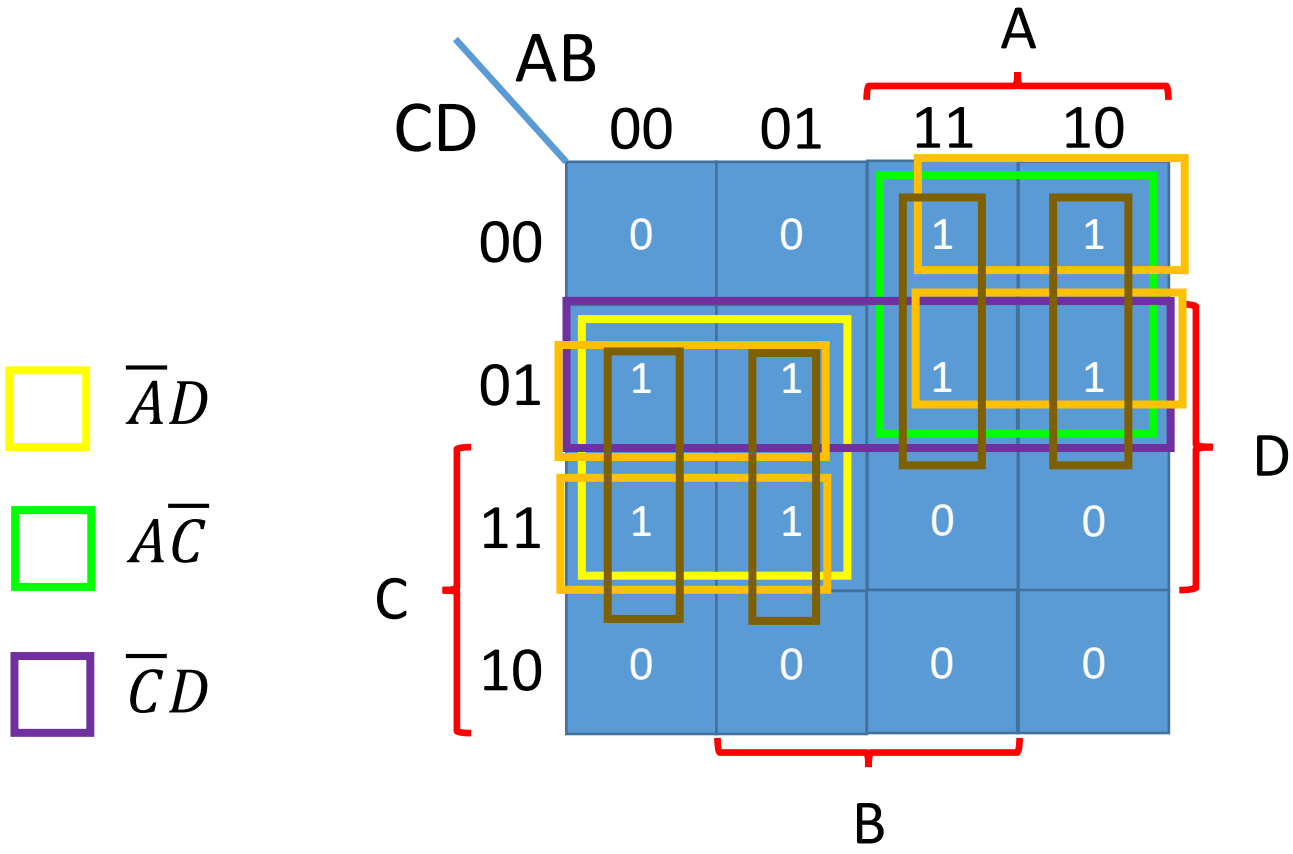
- Glitches
  - An *unwanted pulse* on the output of a combinational logic network
  - A momentary change where the signal should have remained unchanged
- Hazards
  - A circuit that has *the potential to exhibit a glitch* is said to have a hazard

Example Function/Circuit

- F(A, B, C, D) = ∑ m(1, 3, 5, 7, 8, 9, 12, 13)



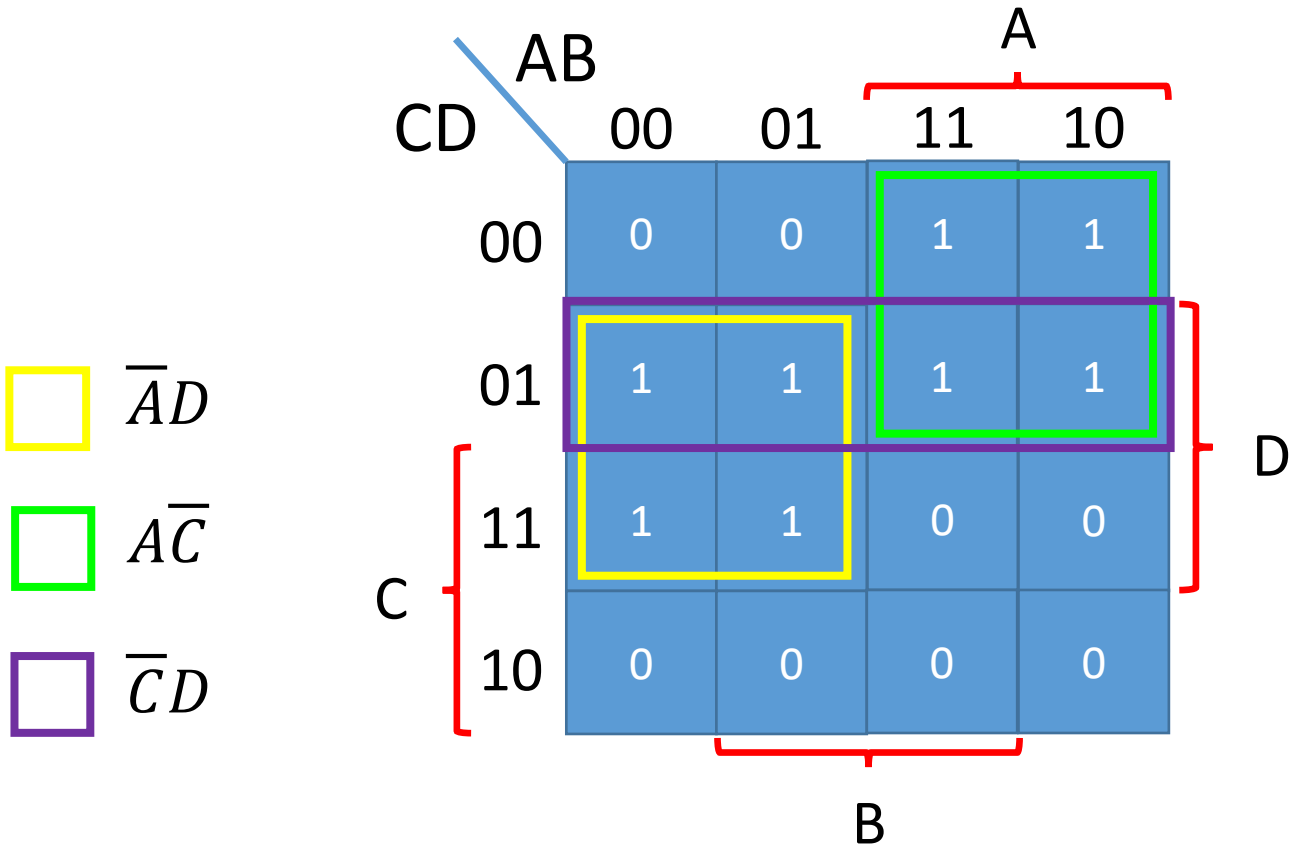
Identify Covering Rectangles



Implicants and Prime Implicants

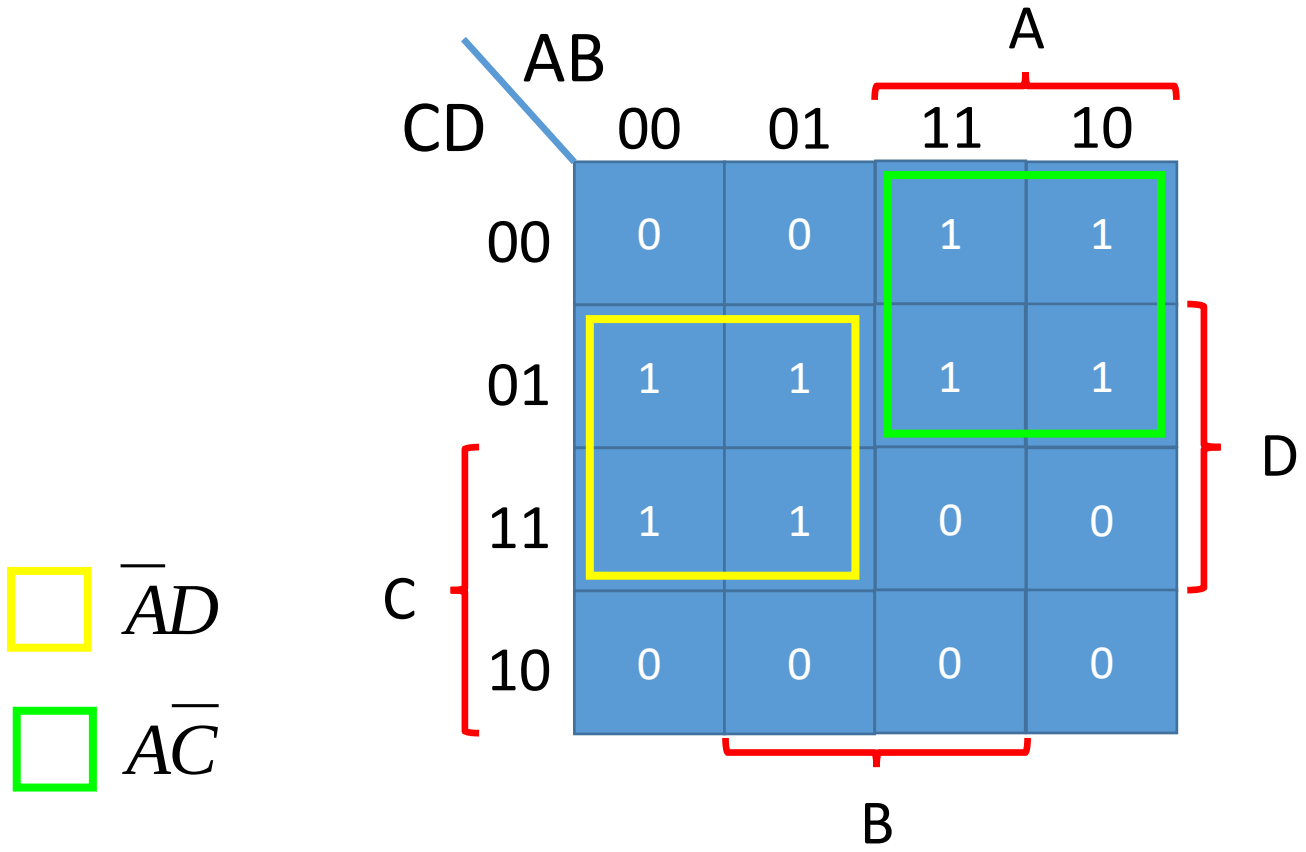
- An ***implicant*** of a function F is any group of elements that can be combined together in a K-map
- A ***prime implicant*** is an implicant that cannot be combined with another implicant to eliminate a literal
  - Grow implicants as much as possible to find prime implicants
- Each prime implicant corresponds to a product term in the minimum sum-of-products expression for function F
- After prime implicants are identified, then a Minimum Cover needs to be determined
  - The smallest set of prime implicants are the ***essential prime implicants***

Identify Prime Implicants



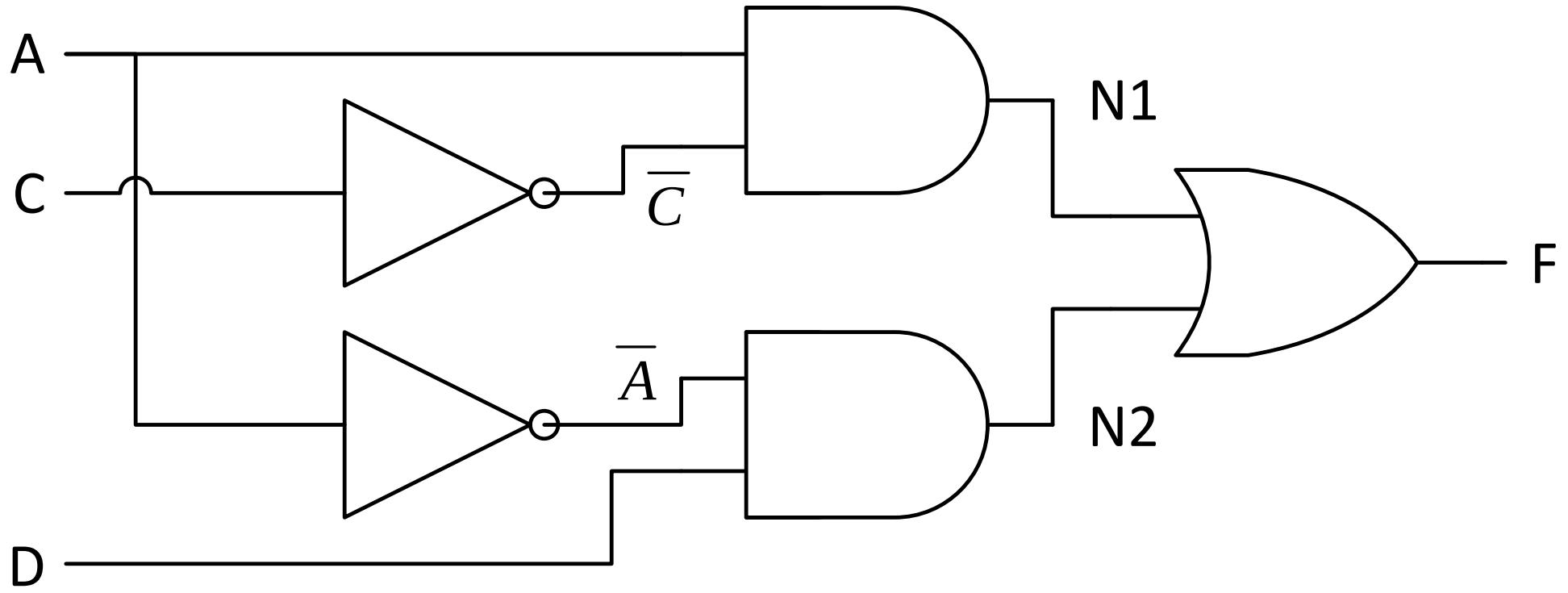
Create Simplified Expression by Finding Essential Prime Implicants in Sum-of-Products Form

- F(A, B, C, D) = $A\overline{C} + \overline{A}D$

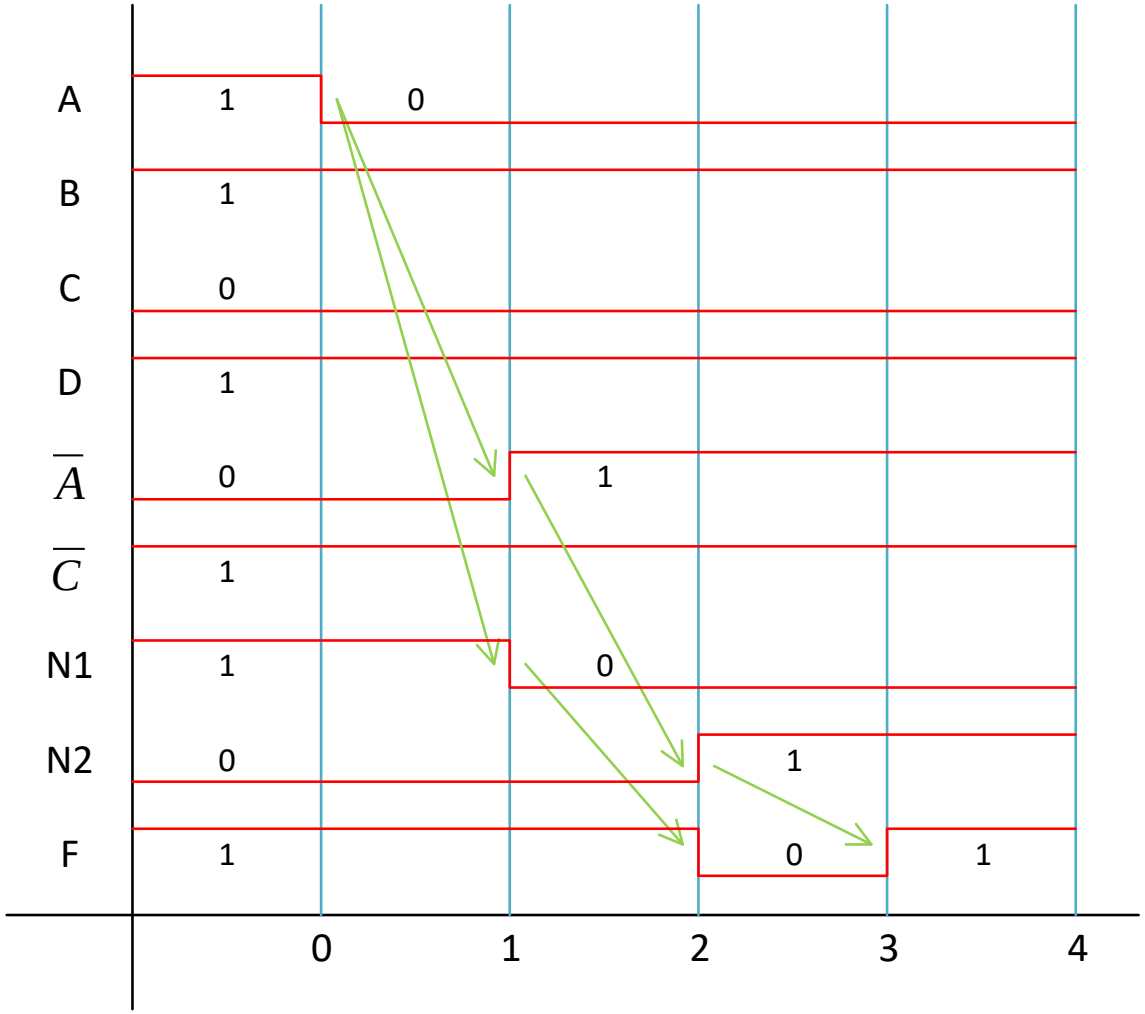


Draw the Corresponding Circuit

- F(A, B, C, D) = $A\overline{C} + \overline{A}D$

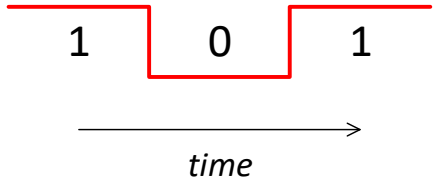


Create the Timing Waveform Diagram

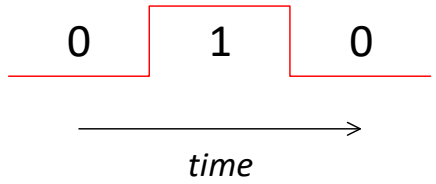


Kinds of Hazards

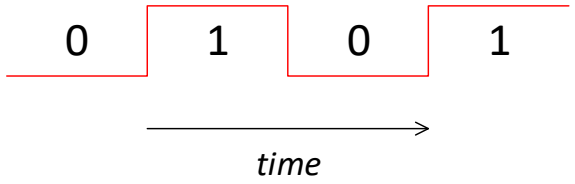
- Static 1-Hazard:

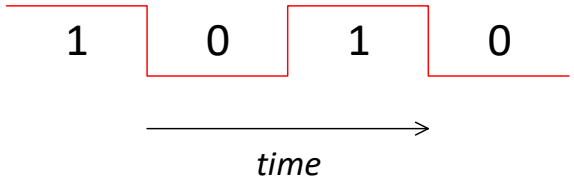


- Static 0-Hazard:



- Dynamic Hazards:





How Our Static 1-Hazard Occurred

- The change in input spanned from one prime implicant to another and that can cause a static 1-glitch
- Such a change in inputs moves from the coverage of one term to the coverage of a different term
  - The output is exposed to possibly exhibit a glitch when not under the coverage of a single term

How to Remove Glitches Caused by Single-bit Input Changes

- If we add redundant prime implicants such that one prime implicant will cover all possible single-bit input change transitions, then there can be no glitches caused by single-bit input changes
- Static 1-Hazard Elimination
  - All static 1-hazards caused by single-bit input change transitions can be found by guaranteeing that each adjacent 1-position (*i.e.,* adjacent on-set elements) in the K-map is covered by a single prime implicant
- Static 0-Hazard Elimination
  - All static 0-hazards caused by single-bit input change transitions can be found by guaranteeing that each adjacent 0-position (*i.e.,* adjacent off-set elements) in the K-map is covered by a single prime implicant

Hazard Elimination

- Adding prime implicants to eliminate static 1-hazards is equivalent to adding terms in the sum-of-products representation

- Adding prime implicants to eliminate static 0-hazards is equivalent to adding factors in the product-of-sums representation

Possible Static Hazards

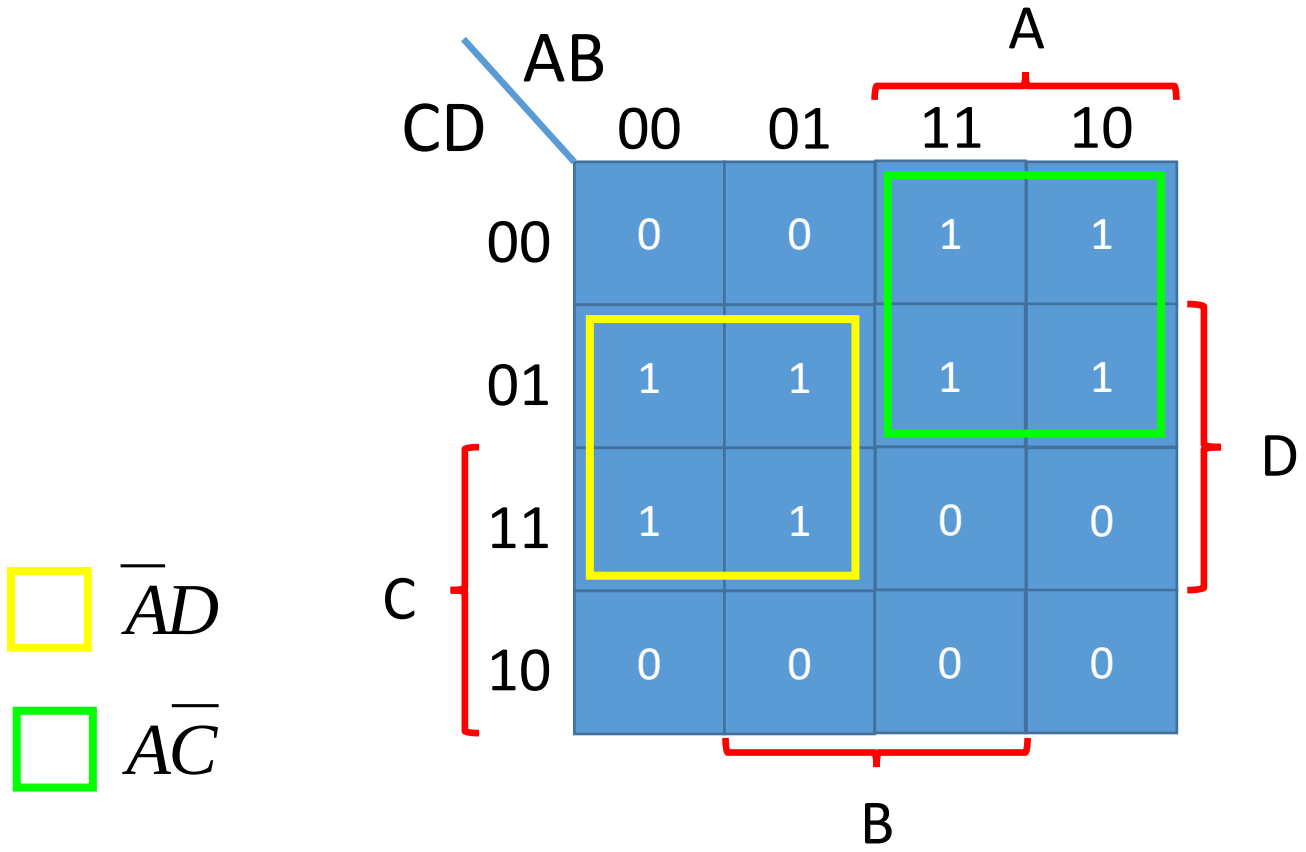
- A two-level expression or circuit (*i.e.,* either sum-of-products or product-of-sums) can never exhibit both static 1-hazards and static 0-hazards from a single-bit input change
  - This is true even if the inputs go through inverters before the two-level logic
- For single-bit input changes,
  - a sum-of-products expression or circuit can exhibit only static 1-hazards
    - Sum-of-products can never exhibit static 0-hazards
  - a product-of-sums expression or circuit can exhibit only static 0-hazards
    - Product-of-sums can never exhibit static 1-hazards

Static Hazards Removal

- Therefore, for single-bit input changes in two-level logic, it is necessary to only remove
  - static 1-hazards in sum-of-products expressions or circuits
  - static 0-hazards in product-of-sums expressions or circuits

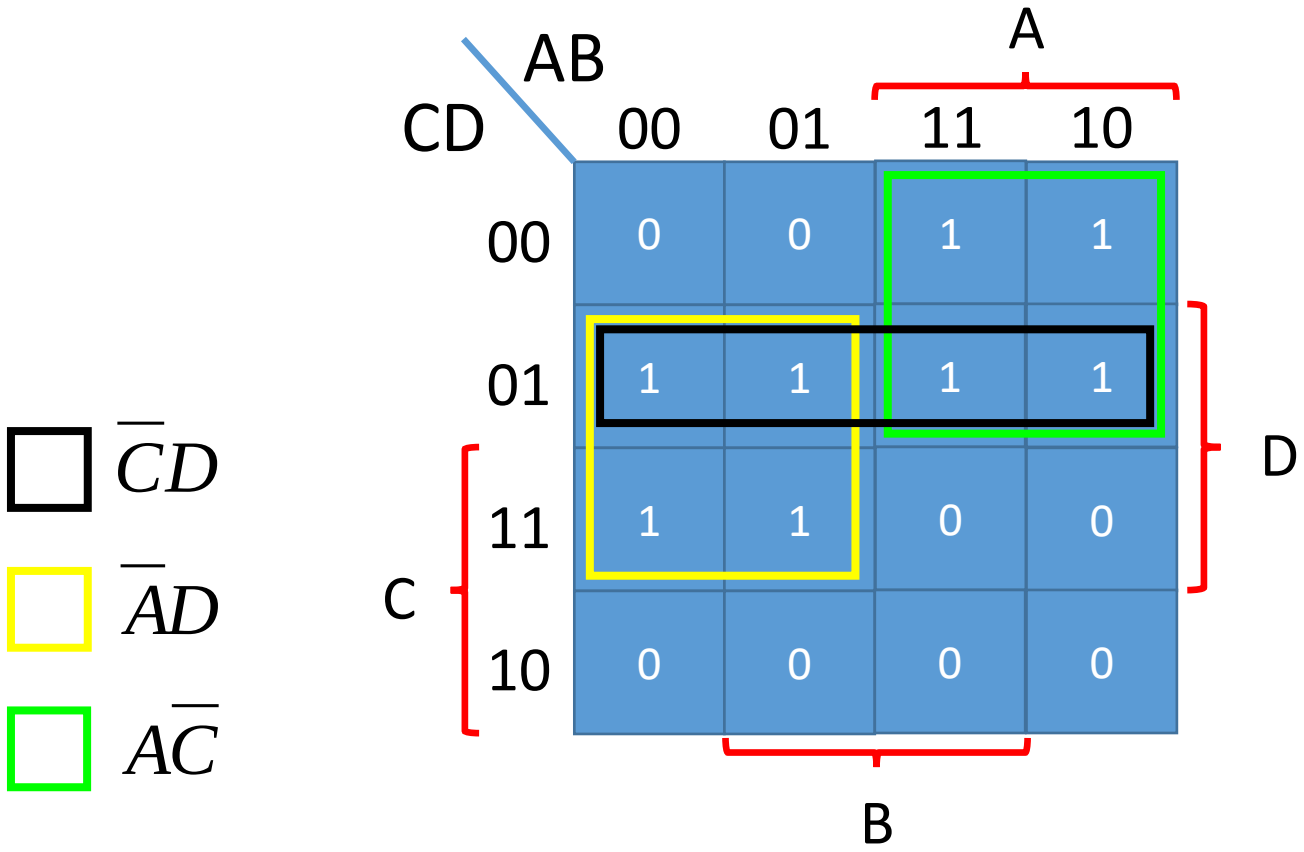
So, Starting with Our Simplified Expression in Sum-of-Products Form

- F(A, B, C, D) = $A\overline{C} + \overline{A}D$



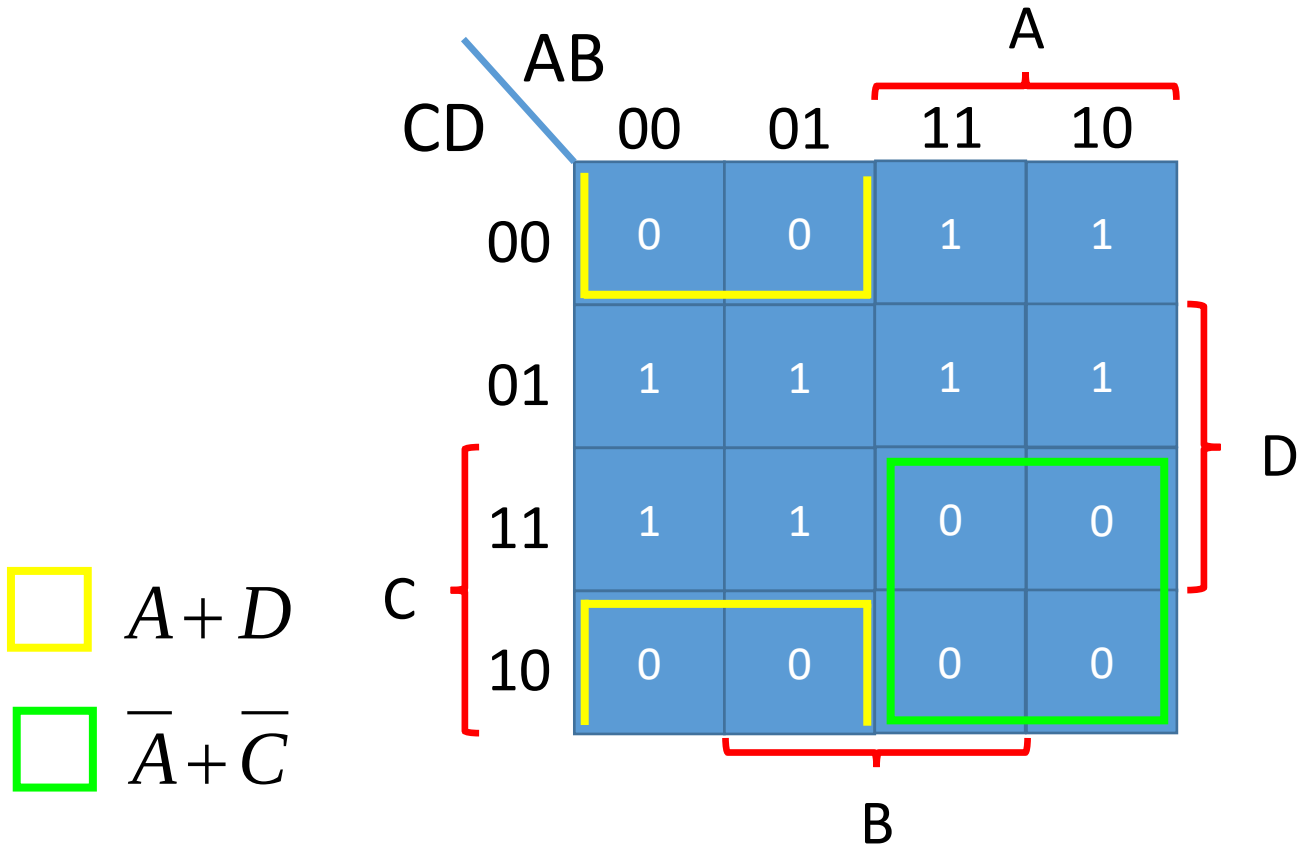
Add Redundant Prime Implicants to Eliminate Static 1-Hazards in Sum-of-Products

- F(A, B, C, D) = $A\overline{C} + \overline{A}D + \overline{C}D$



If *Instead* Creating a *Product-of-Sums* Expression, Find Essential Prime Implicants

- F(A, B, C, D) = $(\overline{A}+\overline{C})(A+D)$



Then, Add Redundant Prime Implicants to Eliminate Static 0-Hazards in *Product-of-Sums*

- F(A, B, C, D) = $(\overline{A}+\overline{C})(A+D)(\overline{C}+D)$