Project Part 2

Warning and alarm Scenarios

Scenario 1 (Gear Air Speed alarm & Air Brake Warning & Gear Override Warning & Gear-up Command)

Use (Gear Air Speed alarm) & (Air Brake warning) & (Gear Override warning)

= (Gear Down=Y) & (Speed >300) & (Speed >=250) & (t<60) & (Speed > 400)

= (Speed >400) & (Gear Down=Y) & (t<60)

Scenario 3 (Scenario 1 with !(Air Brake warning))

Use (Gear Air Speed alarm) & (!Air Brake warning) & (Gear Override warning)

= (Gear Down=Y) & (Speed >300) & ((Speed <250) + (t>=60)) & (Speed > 400)

= (Speed >400) & (Gear Down=Y) & (t>=60)

Scenario 2 logic: (Gear Not Down alarm & Air Brake Warning)

Use (Gear Not Down alarm & Air Brake Warning)

= (Gear Down=N) & ((t<=120) + (alt<1000)) & (Speed >= 250) & (t<60)

= (Gear Down=N) & (t<=120) & (Speed >= 250) & (t<60) +

(Gear Down=N) & (alt <1000) & (Speed >= 250) & (t<60)

= (Speed >= 250) & (Gear Down=N) & (t<60) +

(Speed >= 250) & (Gear Down=N) & (t<60) & (alt <1000)

= (Speed >= 250) & (Gear Down=N) & (t<60)

Scenario 4 logic: (Gear Not Down alarm & !Air Brake warning)

Use (Gear Not Down alarm & !Air Brake warning)

= (Gear Down=N) & ((t<=120) + (alt<1000)) & ((Speed <250) + (t>=60))

= ((Gear Down=N) & (t<=120) & (Speed <250)) + {!BD}

(Gear Down=N) & (t<=120) & (t>=60) + {!B!C}

(Gear Down=N) & (alt<1000) & (Speed <250) + {DE}

(Gear Down=N) & (alt<1000) & (t>=60)) {!CE}

Use a Karnaugh map with the following

A = (Gear Down=N)

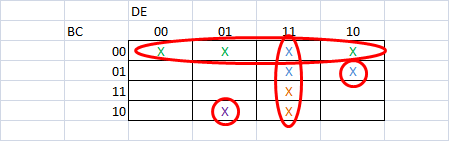
B = (t>120)

C = (t<60)

D = (Speed<250)

E = (Alt<1000)

A is always True, so I can use a 4-way Karnaugh map



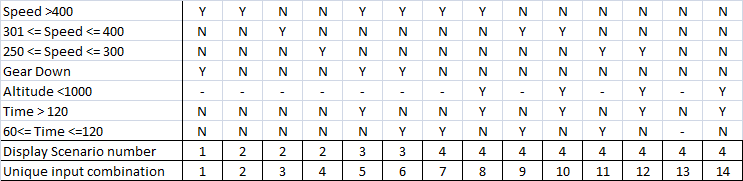
= B'C' + DE + B'CDE' + BC'D'E

= {Any Speed} & (Gear Down=N) & (60<=t<=120) +

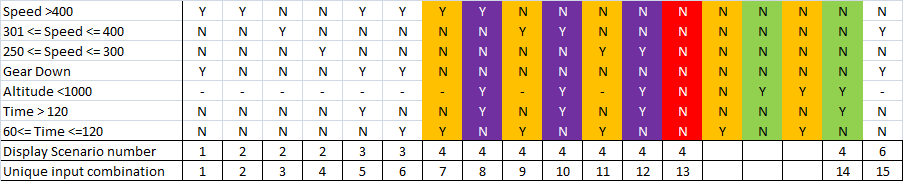
(Speed<250) & (Gear Down=N) & (alt<1000) +

(Speed<250) & (Gear Down=N) & (alt>=1000) & (t<60) +

(Speed>=250) & (Gear Down=N) & (alt<1000) & (t>120)



The following expands back out the Scenario 4 Speed<250 don't cares and maps out the logic (use the colors from the logical expressions above).



So we have demonstrated that the Scenario 4 is correctly captured in the answer (the table previous to the above expanded table).

Scenario 6 logic (Gear Air Speed alarm & Air Brake warning only)

Use (Gear Air Speed alarm & Air Brake warning & !Gear Override warning)

= (Gear Down=Y) & (Speed >300) & ((Speed >=250) & (t<60)) &

((Gear Down=N) + (Speed <= 400))

= (301<=Speed<=400) & (Gear Down=Y) & (t<60)

Scenario 7 logic (Gear Air Speed alarm only)

Use (Gear Air Speed alarm & !Air Brake warning & !Gear Override warning)

= (Gear Down=Y) & (Speed >300) & ((Speed <250) + (t>=60)) &

((Gear Down=N) + (Speed <= 400))

= (Gear Down=Y) & (Speed >300) & (t>=60) & (Speed <= 400)

= (301<=Speed<=400) & (Gear Down=Y) & (t>=60)

Scenario 8 logic (Air Brake warning only)

Use (Air Brake warning & !Gear Air Speed alarm & !Gear Not Down alarm)

= ((Speed >=250) & (t<60)) & ((Gear Down=N) + (Speed <=300)) &

((Gear Down=Y) + ((t<120)&(alt<1000)))

= (250<=Speed<=300) & (Gear Down=Y) & (t<60)

FAQs about the Unique Input Combinations

Q1. For the Gear Not Down alarm why can't I use don't cares in the speed ranges?

A1. Because the logic for the Gear Not Down alarm is based on the Gear Not Down alarm being True and the Air Brake Warning not being True (there are some combinations where I can get both and that is not this scenario). So, to get the Gear Not Down alarm and the !Air Brake Warning I need to look at Scenario 4 above.

Speed >=250 is a critical threshold. I need to have two sets of speeds: >=250 and <250. I can specify <250 as NNN, but I cannot use --- ("-"=don't care) for the speed>=250 condition because I get illegal combinations of speed (two or more Y's) - YYN. So, I have to use YNN,NYN, and YNN as the >250 case. In other words Speed>=250 is really represented as (250<=Speed<=300) + (301<=Speed<=400) + (Speed>400), with one of these conditions being True.

Q2. Am I testing all possible inputs conditions and all possible boundary conditions in these tests?

A2. No, we made this assumption when we developed the test cases by saying that we would only test one combination of the "No alarms and No warnings" scenario. When we made this simplification we purposely chose not to test the entire truth table. It would have taken another 15 unique input combinations with possibly up to 60 more test cases (depending upon the boundary conditions).

In making this assumption we are using a variation of happy path testing (or a variation of positive testing), where we are only testing the conditions that should assert a specified response (but we are not testing all combinations that should not produce a response). For non-security critical or non-safety critical software this can be an acceptable test approach because we are fully testing all specified responses and at each boundary condition.