Step 1:

Step 1 and 2-State the Problem (Analyse) and Inputs, Outputs, Assumptions (Organise)

Design a combinational alarm that sounds whenever the car is started, and any occupied front seat has its seatbelt unfastened.

Inputs (active-HIGH unless noted):

DRIV: driver present (1 = seated)

PASS: passenger present (1 = seated)

BELTD^: active-LOW driver belt (0 = unfastened, 1 = fastened)

BELTP^: active-LOW passenger belt (0 = unfastened, 1 = fastened)

IGN: ignition (1 = ON) — in your slide’s table this is always 1

Output:

ALARM^: active-LOW (0 = alarm ON, 1 = alarm OFF)

When the ignition is ON, turn the alarm ON (LOW) if driver is seated and their belt is unfastened, or passenger is seated and their belt is unfastened. Otherwise keep the alarm OFF (HIGH).

binary logic

we explicitly consider every input combo when IGN=1

Step3-Design the Solution

Final output equation:

ALARM\_ON = IGN · ( DRIV·¬BELTD^ + PASS·¬BELTP^ )  
  
Truth Table:

This truth table matches the lecture slide exactly (IGN fixed at 1). BELTD^ and BELTP^ are active-LOW inputs (0 = unfastened, 1 = fastened). ALARM^ is active-LOW output (0 = alarm ON, 1 = alarm OFF).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DRIV | PASS | BELTD^ | BELTP^ | IGN | ALARM^ |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Pseudocode:

read DRIV, PASS, BELTD^, BELTP^, IGN

NOT BELTD^

NOT BELTP^

IF IGN = 1 AND ( (DRIV AND D\_unf) OR (PASS AND P\_unf) ) THEN

ALARM^ ← 0 // sound alarm

ELSE

ALARM^ ← 1 // silence

ENDIF

Flow chart:





