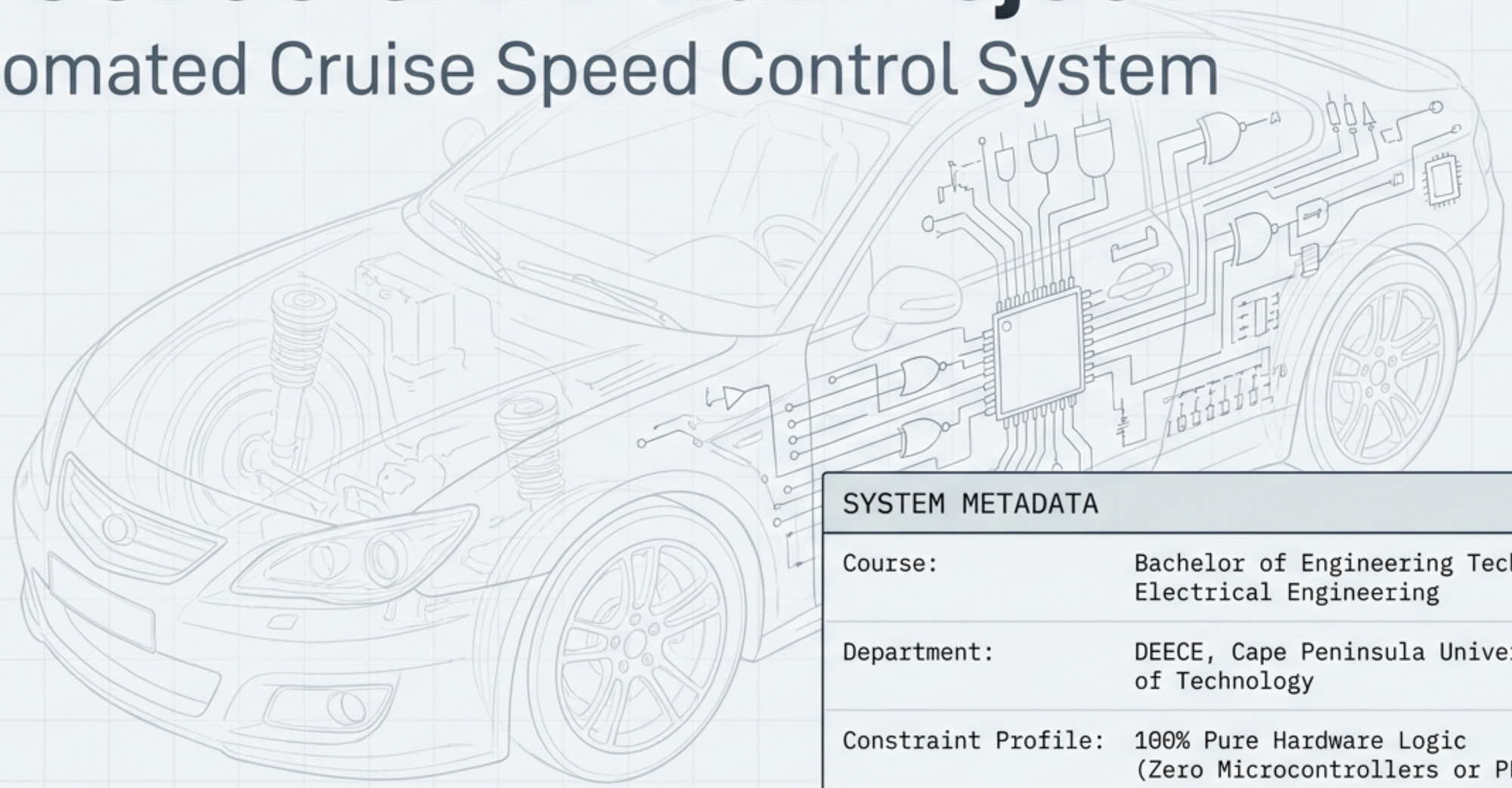
CSS370S GA1 Final Project

Automated Cruise Speed Control System

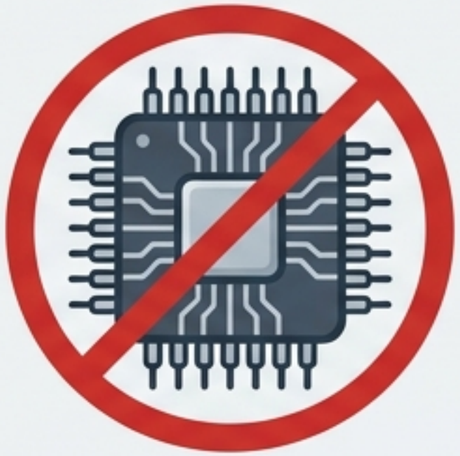


SYSTEM METADATA	
Course:	Bachelor of Engineering Technology: Electrical Engineering
Department:	DEECE, Cape Peninsula University of Technology
Constraint Profile:	100% Pure Hardware Logic (Zero Microcontrollers or PLCs)

THE OBJECTIVE

Design and implement a closed-loop automated cruise speed control system that senses, displays, evaluates, and reacts to real-time speed data.

THE CORE CONSTRAINT

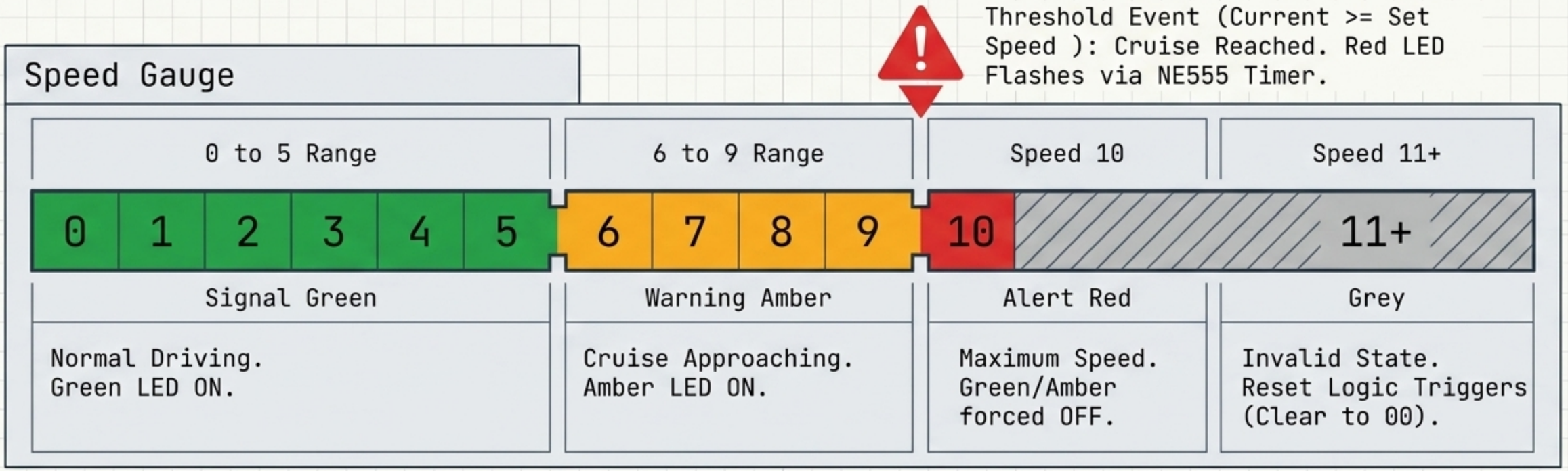


No Microcontrollers. No PLCs. The entire computational load must be handled strictly by analogue sensors, discrete digital logic gates, timers, and hardware comparators.

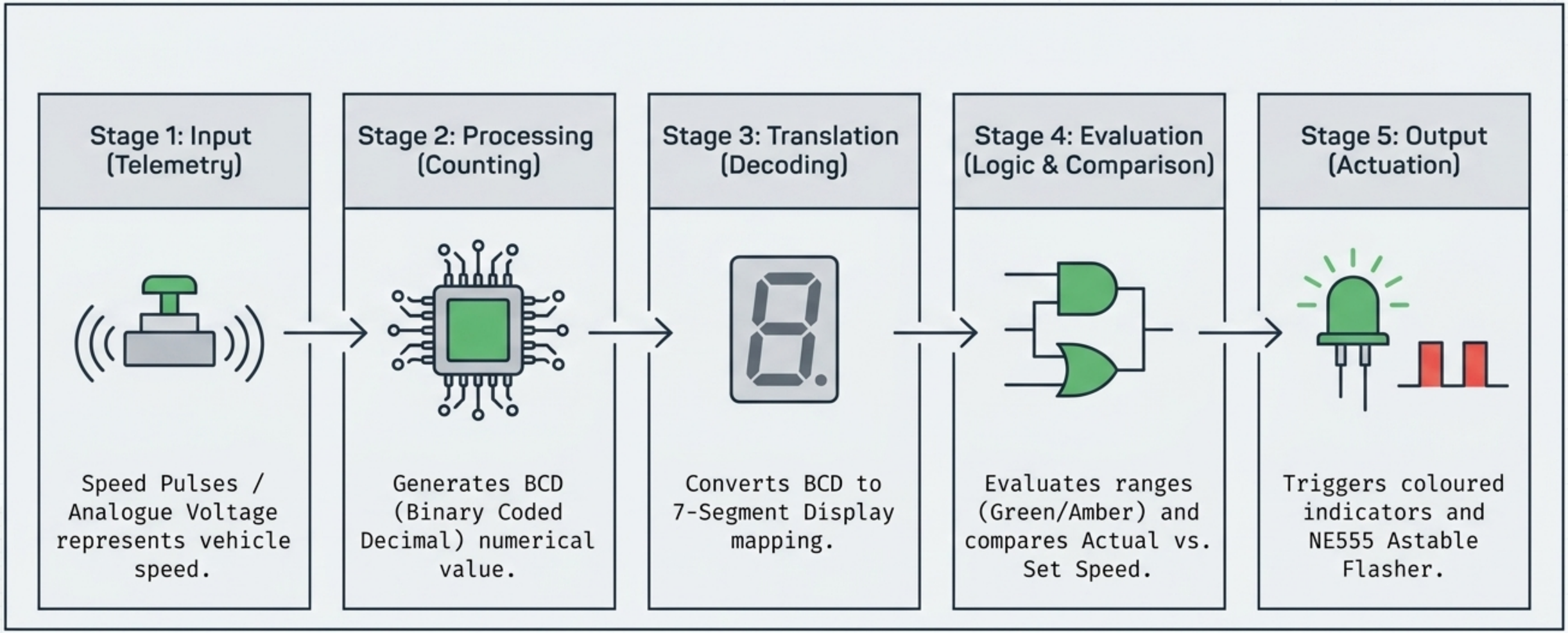
FUNCTIONAL REQUIREMENTS

- Display current speed continuously.
- Accept and store a user-selected set cruise speed.
- Trigger visual indicators based on precise speed ranges.
- Block or reset invalid state outputs (e.g., above max speed).

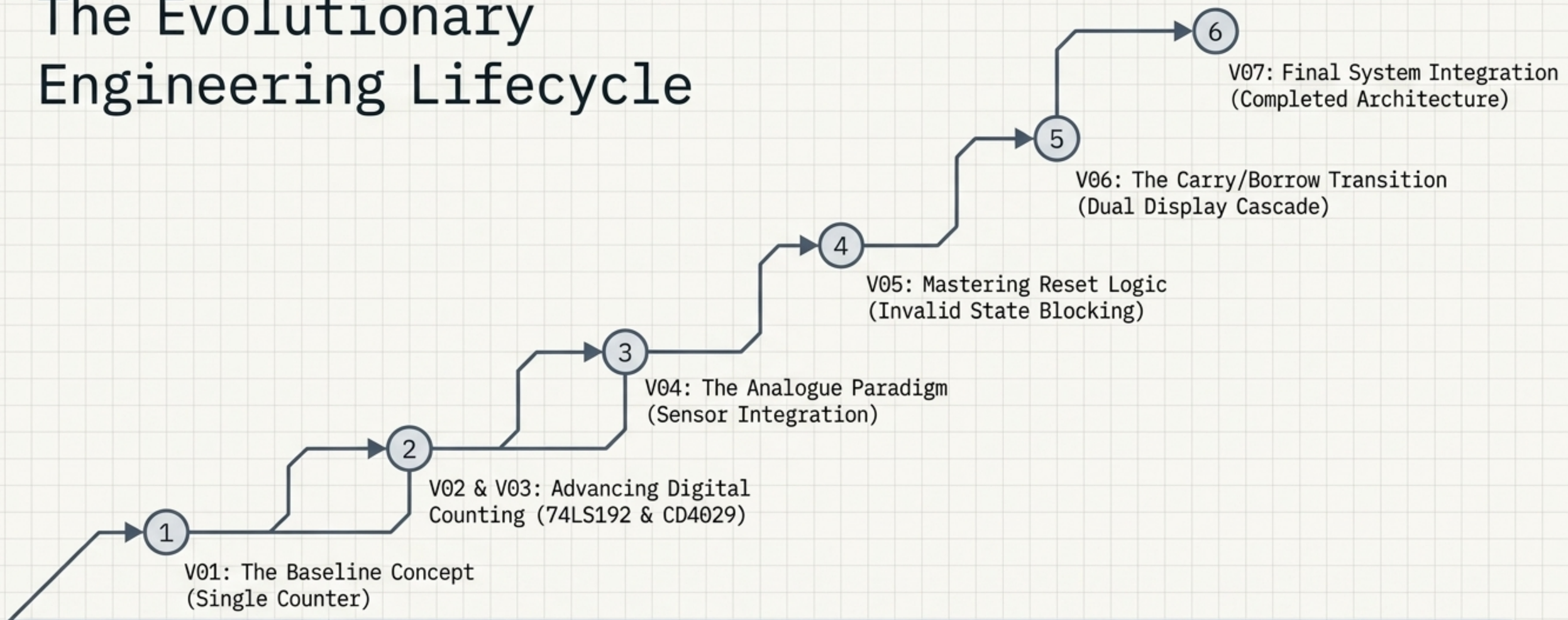
The Rules of the Road: System Logic Map



Universal System Architecture Blueprint

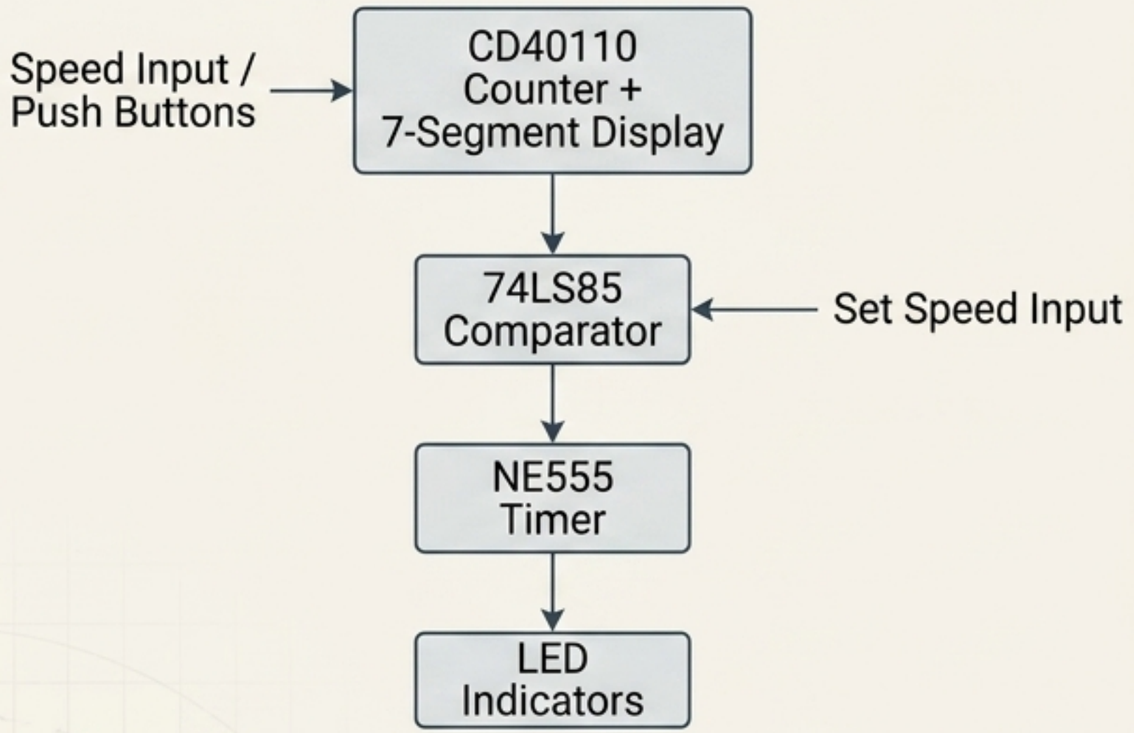


The Evolutionary Engineering Lifecycle

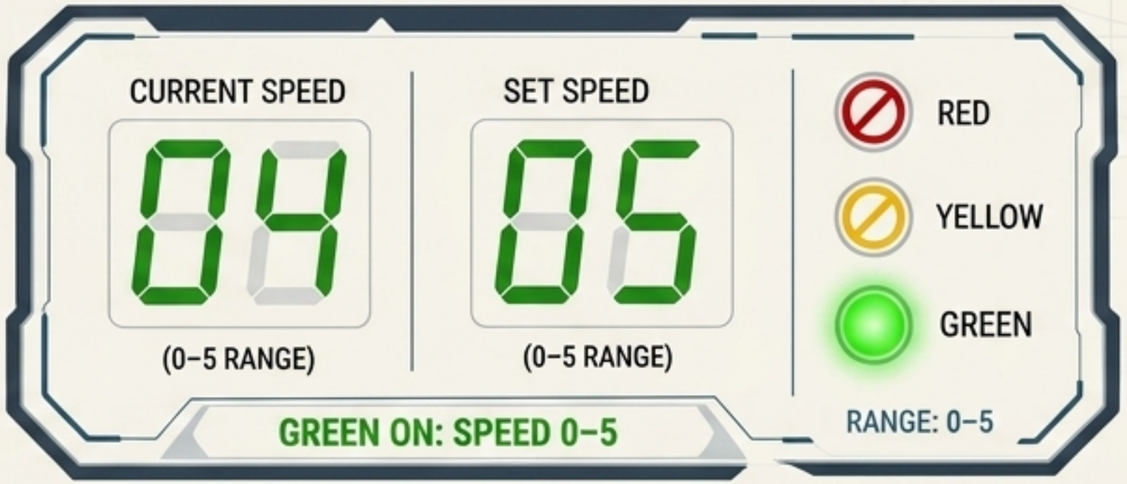


One connected journey: Each iteration resolves the hardware limitations of the previous stage, proving the same cruise-control principles across diverse IC families.

Stage 1: The Baseline System (V01)



Architecture Widget:
Counter/Display: CD40110 (Single IC handling both counting and 7-segment driving).
Evaluation: 74LS85 Comparator.
Output: NE555 Flasher.



Performance Note:
V01 establishes the foundational control logic before introducing the complexity of dual-displays and automated reset systems.

Stage 2: Advancing the Digital Counters (V02 & V03)

V02: 74LS192 System

- **Key Feature:** Synchronous Up/Down counting with 74LS48 display drivers.
- **Focus:** Independent UP/DOWN input pins requiring specific pulse management.

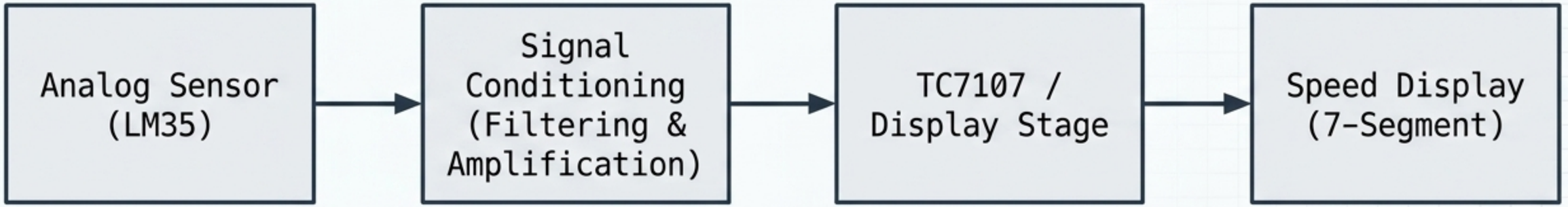
V03: CD4029 System

- **Key Feature:** Presettable Up/Down counter (BCD mode) with 74LS47 display drivers.
- **Focus:** Single clock input with a separate UP/DOWN control pin.

Both ICs successfully achieve the required BCD generation, proving the system's modularity.

Stage 3: The Analogue Paradigm (V04)

The Input Shift: Transitioning from discrete digital pulses to continuous environmental data.



Component Chain Widget		
LM35 Sensor: Senses temperature to simulate variable vehicle speed.	**Signal Conditioning:** Filters and amplifies the raw analogue signal.	**TC7107 Driver:** Directly converts the analogue input into a digital 7-segment display output.

Evaluation Note
Brings the project closer to real-world automotive telemetry by processing unrefined continuous data.

Deep Dive:
The ‘Speed 10’ Logic Trap

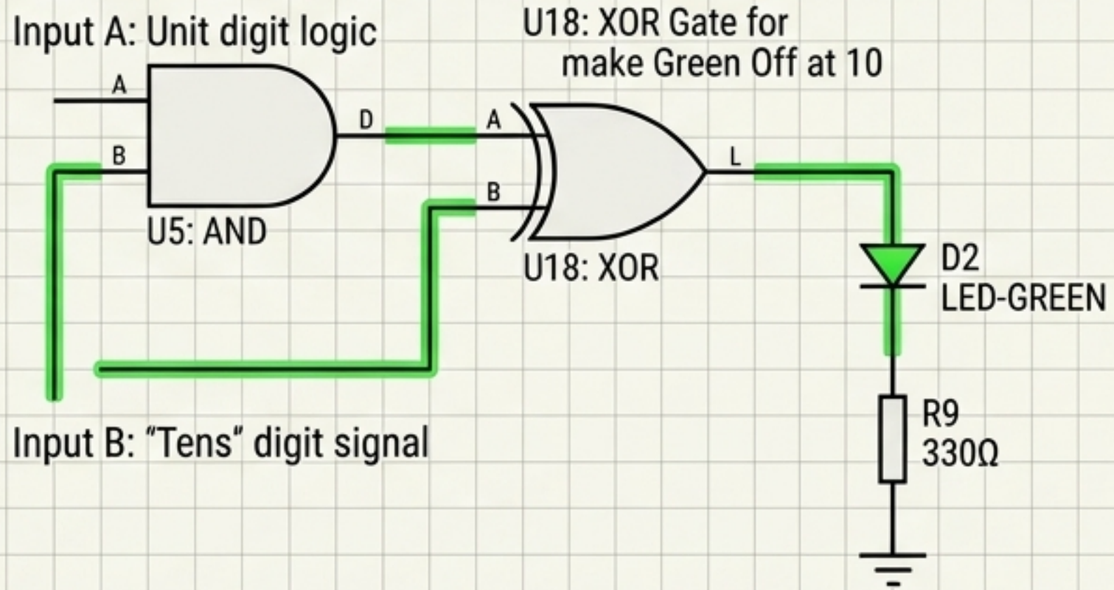
The Problem Widget

At speed 10, the ‘ones’ unit digit resets to 0. Basic logic interprets ‘0’ as falling within the 0–5 Green Range, incorrectly turning the Green LED ON.

The Hardware Solution

Input A: Unit digit logic (would trigger Green).
Input B: ‘Tens’ digit signal.

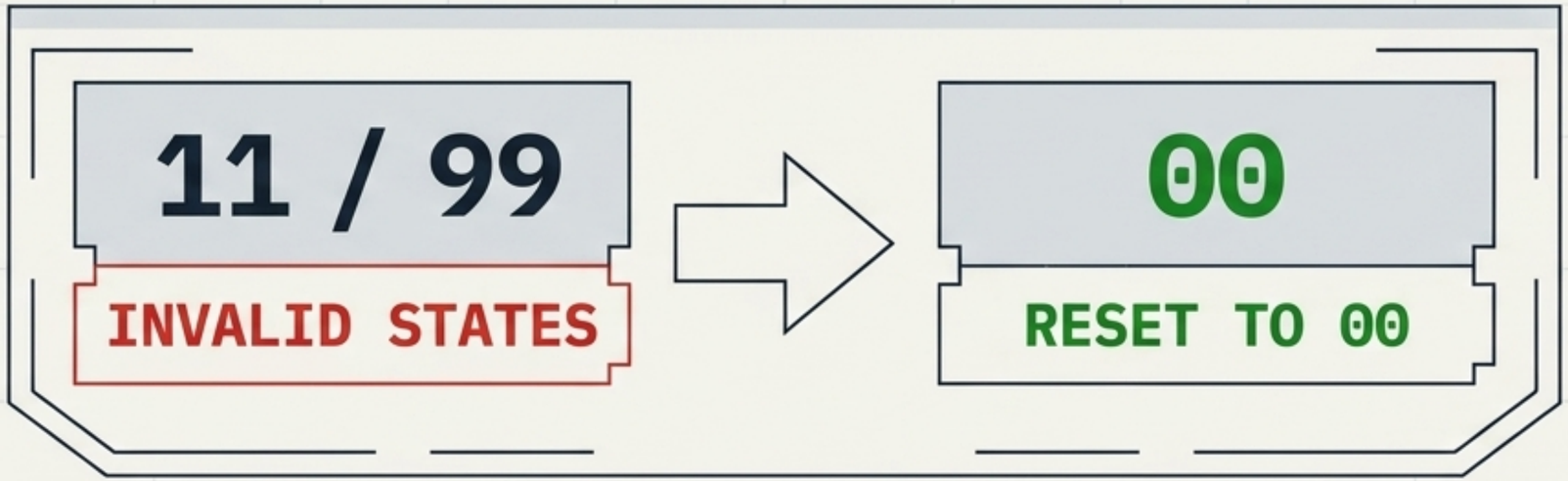
Result: The XOR correction actively uses the presence of the ‘tens’ signal to forcefully inhibit and kill the green light circuit, maintaining system integrity without code.



Stage 4: Mastering Reset Logic (V05)

Component: 74LS192 with active Reset Control.

The Challenge: A standard counter will continue past decimal 10 into invalid outputs (11 or 99).



The Hardware Intercept

Logic gates monitor the BCD output lines.

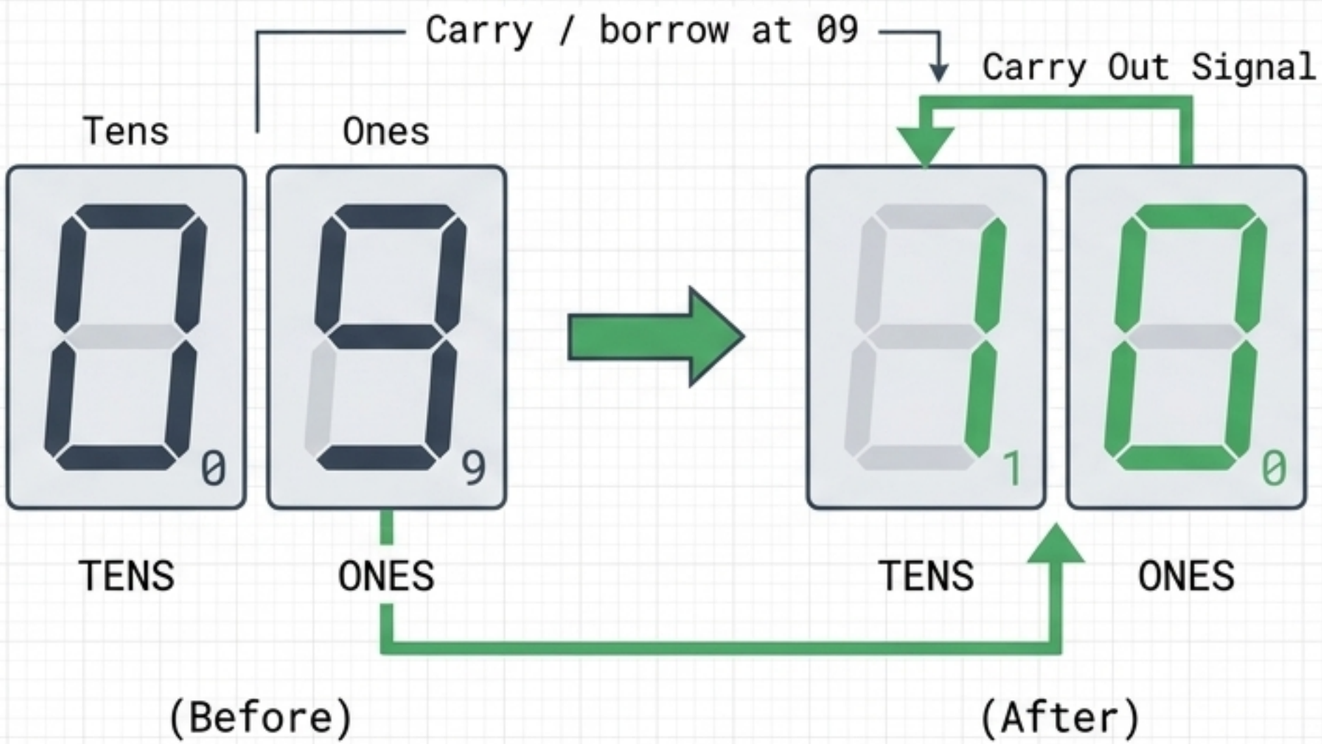
When the specific binary pattern for '11' (or a designated upper limit) is detected, the signal is routed to the clear/reset pin of the 74LS192.

Result: The system instantaneously resets to '00', simulating an automatic cruise disengagement upon critical failure.

Stage 5: The Carry/Borrow Chasm (V06)

Architecture: CD4029 Counters cascaded into CD4511 Common-Cathode Display Drivers.

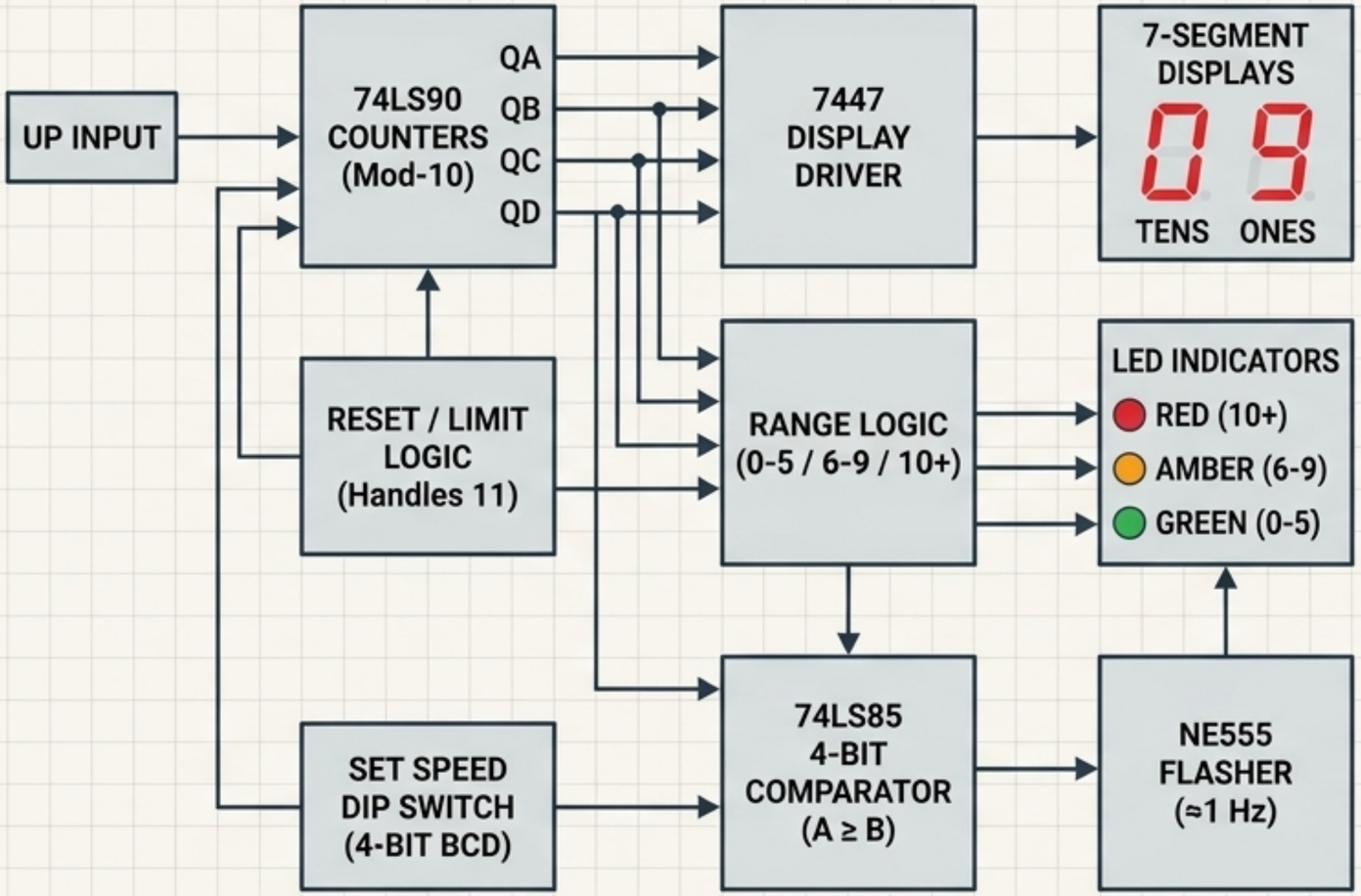
The Critical Threshold: The exact moment the vehicle accelerates from 09 to 10.



Hardware Execution Widget

1. The 'Ones' counter maxes out at 9.
2. Upon receiving the next pulse, it triggers a triggers a 'Carry Out' signal.
3. This signal physically flows into the 'Clock' input of the 'Tens' counter, incrementing it from 0 to 1, while the 'Ones' counter rolls back to 0.

The Final Integration (V07)



Core Logic Setup

74LS90 Counters + 7447 Display Drivers.

DIP Switches: Used to physically program the 4-bit BCD 'Set Speed'.

The 74LS85 Comparator Balance

Equation: RedEnable = (A > B) OR (A = B)

Triggers the ~1Hz NE555 Astable Flasher.

Status

Successfully merges three-stage range indication (Green/Amber), threshold warnings (Flashing Red), and Limit Logic (handling attempted 11s).

Hardware Evaluation Matrix

Version	Primary Counter	Display Driver	Input Paradigm	Crucial Innovation
V01	CD40110	Integrated	Push-button	Baseline 1-counter proof
V02	74LS192	74LS48	Push-button	Synchronous Up/Down
V03	CD4029	74LS47	Push-button	BCD mode selection
V04	LM35 (Sensor)	TC7107	Analogue	Environmental telemetry
V05	74LS192	74LS47	Push-button	Automated bounds reset
V06	CD4029	CD4511	Push-button	Carry/Borrow cascade handling
V07	74LS90	7447	DIP Switch	Full 4-bit comparison integration

Project Conclusion: GA1 Evaluation Alignment

Problem Definition (GA1 1 & 2)

Successfully conceptualised a microcontroller-free environment, bounding the logic strictly to discrete hardware capabilities.

Solution Strategy (GA1 3 & 4)

Iteratively deployed 7 distinct circuit architectures, systematically solving digit spillovers, false-positive logic (Speed 10 XOR), and analogue signal conversion.

Implementation & Evaluation (GA1 5 & 6)

Final closed-loop integration successfully displays telemetry, dictates colour-coded safety ranges, and triggers hardware-level automated resets, validating the engineering lifecycle.