This ladder logic design is for a **PLC (Programmable Logic Controller)** project involving **timers, counters, limit testing, and digital inputs/outputs**. Here's a breakdown of what each part (rung) does:

**Rung-by-Rung Explanation**

**Rung 0: Start/Stop Logic**

* **Inputs:**
  + Stop button → <Local:1:I.Data.0>
  + Start button → <Local:1:I.Data.1>
* **Output:**
  + Run signal is latched (Set/Reset logic)

**Purpose:** Basic motor or system run control using Start/Stop push buttons.

**Rung 1: Retentive Timer RTO1**

* **Timer:** RTO1
  + Preset: 99,999,999
  + Accumulated: 27,878
* **Trigger:** Run signal
* **Behavior:** Retentive timer that runs while system is active.

**Purpose:** Keep time when the system is running, does not reset on stop.

**Rung 2: Free-Running Timer (TON)**

* **Timer:** FreeRunning
  + Preset: 5000
  + Accumulated: 0 (initial)
* **Trigger:** Run signal
* **Reset:** When FreeRunning.DN is true

**Purpose:** Creates a looping timer that counts repeatedly while the system is running.

**Rung 3: Greater Than Comparison**

* **Condition:** If FreeRunning.ACC > 4000
* **Output:** Turns on Out\_1 (<Local:1:O.Data.0>)

**Purpose:** Activates an output when timer accumulation exceeds 4000.

**Rung 4: Push Button and Output Control**

* **Input:** PB\_1 (<Local:1:I.Data.2>)
* **Output:** Out\_3 (<Local:1:O.Data.3>)

**Purpose:** Turns on an output when PB\_1 is pressed.

**Rung 5: Counter CTU1**

* **Trigger:** Rising edge from Out\_3
* **Counts Up To:** 5
* **Reset:** When CTU1.DN is reached

**Purpose:** Counts how many times PB\_1 is pressed (or Out\_3 is activated), then resets.

**Rung 6: Equals Comparison**

* **Condition:** CTU1.ACC == 4
* **Output:** Out\_4 (<Local:1:O.Data.4>)

**Purpose:** Activates output when counter reaches 4.

**Rungs 7–11: Range Checks (LIM instructions)**

Use LIM (limit) blocks on FreeRunning.ACC:

* **Out\_5 (100–200)**
* **Out\_6 (300–400)**
* **Out\_7 (500–600)**
* **Out\_8 (700–800)**
* **Out\_9 (900–1000)**

**Purpose:** These outputs activate based on the timer value falling within specific ranges.

**Rung 12: One-Shot Logic**

* **Trigger:** FreeRunning.ACC > 4000
* **With:** One-shot rising (ONS) contact Bit1
* **Output:** Out\_10 (<Local:1:O.Data.10>)

**Purpose:** Sends a **single pulse** output when the timer exceeds 4000 — avoids continuous activation.

**Rung 13: Output Reset with Input**

* **Input:** <Local:1:I.Data.4>
* **Resets:** RTO1 timer

**Purpose:** Manual reset for the retentive timer.

**Summary: What This PLC Ladder Logic Does**

This project simulates a **simple control system** with the following features:

* Basic **Start/Stop control**
* **Retentive and free-running timers**
* **Output triggers** based on time ranges
* **Counter** for button press events
* **One-shot pulse** output
* **Manual reset** for the RTO time