



## Stepwise Guide: Q4 DELAY Calculation

Let's walk through how to calculate the TOTAL DELAY for the given DELAY subroutine step-by-step, as asked in Q4, with a clock frequency of **2 GHz** (2,000,000,000 Hz):

### 1. Understand the Code Structure

```
LDI R16, 0xFF      ; Outer loop init (AGAIN3)
LDI R17, 0xFF      ; Middle loop init (AGAIN2)
LDI R18, 0x05      ; Inner loop init (AGAIN1)
AGAIN1:
    DEC R18
    BRNE AGAIN1
    DEC R17
    BRNE AGAIN2
    DEC R16
    BRNE AGAIN3
RET
```

- There are **three nested loops**: R16 (outer), R17 (middle), R18 (inner).
- Each loop counts down from initial value to zero (inclusive).

### 2. Instruction Cycle Times (from question)

- **LDI**: 1 cycle
- **DEC**: 2 cycles
- **BRNE**: 4 cycles
- **RET**: Not needed for large delay, but is 4 cycles

### 3. Calculate the Total Number of Loop Iterations

- Outer loop: R16 from 0xFF (255) down to 0 → 255 iterations (AGAIN3)
- Middle loop: R17 from 0xFF (255) down to 0 → 255 iterations (AGAIN2)
- Inner loop: R18 from 0x05 (5) down to 0 → 5 iterations (AGAIN1)
- So, **total loop executions** =  $255 \times 255 \times 5 = 325,125$

But, we need to account for each instruction's placement in the loops, since each gets executed a different number of times!

## 4. Breakdown: Instruction Execution Counts

### a) LDI instructions

- Each LDI (R16, R17, R18) executes once: **3 instructions × 1 cycle = 3 cycles**

### b) Inner Loop (**AGAIN1: DEC R18 + BRNE AGAIN1**)

- For every run through inner loop (for each R18 value):
  - **DEC R18:** Runs once per inner loop iteration (so 5 times per middle-outer).
  - **BRNE AGAIN1:** Runs:
    - 4 times (branch taken: for R18 = 4,3,2,1)
    - Once (not taken: for R18 = 0)
- This pattern repeats for every run of the outer and middle loops (so  $255 \times 255$  times).

### c) Middle Loop (**DEC R17 + BRNE AGAIN2**)

- After inner loop ends, DEC R17 runs once, BRNE AGAIN2 runs once (either taken, or not).
- This is repeated for all  $255 \times 255$  executions (one per outer loop).

### d) Outer Loop (**DEC R16 + BRNE AGAIN3**)

- After middle loop ends, DEC R16 runs once, BRNE AGAIN3 runs once (same logic).
- This is repeated for all 255 executions.

### e) RET at the end

- Runs once, after all loops end.

Let's organize per loop level.

## 5. Detailed Instruction Count Table

Assume:

- Outer loop (R16): 255 times
- Middle loop (R17): 255 times per outer
- Inner loop (R18): 5 times per middle per outer

So:

- **Inner Loop Executions:**  $255 \times 255 \times 5 = 325,125$
- **Middle Loop Executions:**  $255 \times 255 = 65,025$
- **Outer Loop Executions:** 255

### a) Inner Loop detailed

- **DEC R18:**  $5 \times (255 \times 255) = 325,125$
- **BRNE AGAIN1:** For each inner loop:
  - Branch taken for R18 = 4,3,2,1  $\rightarrow$  4 times per loop  $\times$  65,025 loops = 260,100
  - Not taken for R18 = 0  $\rightarrow$  1 time per loop  $\times$  65,025 loops = 65,025
- **Total BRNE AGAIN1 executions:** 325,125

### b) Middle Loop

- **DEC R17:** Once per completion of an inner loop, so 65,025 times
- **BRNE AGAIN2:** same (either taken or not given the counter) = 65,025 times

### c) Outer Loop

- **DEC R16:** 255 times
- **BRNE AGAIN3:** 255 times

### d) LDI instructions

- 3 times (once for each register)

### e) RET

- Once

## 6. Add Up Total Machine Cycles

**LDI:**  $3 \times 1 = 3$  cycles

#### Inner Loop:

- DEC R18:  $325,125 \times 2$  cycles = 650,250
- BRNE AGAIN1:  $325,125 \times 4$  cycles = 1,300,500

#### Middle Loop:

- DEC R17:  $65,025 \times 2$  cycles = 130,050
- BRNE AGAIN2:  $65,025 \times 4$  cycles = 260,100

#### Outer Loop:

- DEC R16:  $255 \times 2$  cycles = 510
- BRNE AGAIN3:  $255 \times 4$  cycles = 1,020

**RET:** 4 cycles

**TOTAL CYCLES:**

- LDI: 3
- Dec R18: 650,250
- Brne Again1: 1,300,500
- Dec R17: 130,050
- Brne Again2: 260,100
- Dec R16: 510
- Brne Again3: 1,020
- Ret: 4
- **Sum = 2,342,437 cycles**

## 7. Convert Cycles to Real Delay

At **2 GHz clock**:

- Cycle time =  $1 / 2,000,000,000$  seconds = 0.5 ns (nanoseconds) per cycle
- Total time =  $2,342,437 \times 0.5$  ns = 1,171,218.5 ns
- **Convert to milliseconds:**
  - 1 ms = 1,000,000 ns
  - Time = 1,171,218.5 ns = about **1.17 ms**

## Final Answer Summary

- The DELAY subroutine with these loop counts and instruction cycle times, at a 2 GHz clock, will create a total delay of about **1.17 milliseconds**.

If you want to learn how to adjust this delay (make it longer/shorter), you can change loop start values or instruction sequence. Need a walkthrough for the 7-segment section, or more on delay logic? Let me know!

✱

1. WhatsApp-Image-2025-11-08-at-12.54.35-AM.jpg