Chart

Description automatically generatedH89/ESP32 Timing Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **uSec** |  |  |
| Cmd | 7e |  |  |  |
| Data | 7c |  |  |  |
| Status | 7d |  |  |  |
| H89 CPU clock | 8,000,000 | 0.125 |  |  |
| ESP32 CPU Clock | 240,000,000 | 0.004167 |  |  |
|  |  |  |  |  |
|  |  | **uSec** | **H89 Clk Cycles** | **ESP Clk Cycles** |
| H89 writes command port/ ESP32 Interrupt 7E |  | 0 |  |  |
| H89 Reads ESP32 status |  | 25 | 200 |  |
| H89 writes command data/ ESP32 Interrupt 7C |  | 116 | 928 |  |
| ESP Reads starts data Read |  | 141 |  | 6,000 |
| ESP Reads ends data read |  | 143 |  | 480 |
|  |  |  |  |  |
| ESP Time to read Parallel data |  | 2 |  |  |
| H89 time to write data |  | 0.3 |  |  |

Notes:

1. The H89 code is written in Turbo Pascal. Assembly language should speed this up considerably.
   1. Time from H89 start command process to first byte read by ESP32 is 141 usec
   2. Time from ESP32 data in interrupt to ESP read byte is 2 usec or 480 ESP32 clock cycles or 16 H89 clock cycles at 8 MHz.
2. The ESP32 is interrupt driven for read data transfers from the H89.
   1. H89 writes data to port 7C
      1. ESP32 gets interrupt, Sets ESP32\_BUSY flag
      2. ESP32 reads data in
      3. ESP32 checks if Command mode or Data
      4. ESP32 loads data either the Command or Data Buffer
      5. ESP32 sets status to H89\_WRITE\_OK or ESP\_BUSY if the data buffer is full
   2. The ESP32 data writes
      1. Data is written to the output latch
      2. Status is set to H89\_READ\_OK
   3. H89 reads data from port 7C
      1. ESP32 gets interrupt Read FC
         1. ESP32 decrements h89BytesToRead counter down to zero
            1. Initially planned to use as buffer pointer. Essentially, a flag now 1 or 0
         2. Sets status to ESP32\_BUSY
      2. ESP32 calling routine needs to clear flag ESP32\_BUSY to indicate read or write is ok.
3. Commands
   1. H89 writes to port 7E
   2. ESP32 checks variable cmdFlag
      1. If set to 1, then reboot (two successive writes to port 7E)
      2. Else sets cmdFlag to 1
         1. Successive data writes to 7C load command buffer, cmdData

A screenshot of a computer

Description automatically generated

6.5 us for Turbo Pascal to write successive ports, requires 54 clock cycles

H89 running at 8 MHz, 120 ns clock cycle