Perfektní, jdeme na to! Začneme pěkně od začátku – krok po kroku. Níže najdeš **kompletní řešení praktické maturitní zkoušky pro mikrokontrolér 8051 v assembleru** podle zadaní z fotky.

**✅ ÚKOL Č. 1 – Návrh zapojení MCU 8051 (2x LED, 1x 7segment, 2x tlačítko)**

**1.1 Blokové schéma**

lua

ZkopírovatUpravit

+------------------------+

| MCU 8051 |

| |

| P1.0 ----> LED1 |

| P1.1 ----> LED2 |

| P1.2 ----> Tlačítko1 |

| P1.3 ----> Tlačítko2 |

| P2.0-P2.6 -> 7-seg A-G|

| P2.7 ----> 7seg DP |

+------------------------+

**1.2 Schéma zapojení (textová forma)**

* **LED1**: Anoda na P1.0 přes rezistor (např. 330 Ω), katoda na GND
* **LED2**: Anoda na P1.1 přes rezistor, katoda na GND
* **Tlačítko1**: Jeden kontakt na P1.2, druhý na GND (s pull-up odporem nebo vnitřní)
* **Tlačítko2**: Jeden kontakt na P1.3, druhý na GND
* **7-segmentovka (common cathode)**:
  + Segmenty A-G + DP připojeny na P2.0 až P2.7 přes rezistory (např. 220 Ω)
  + Společná katoda na GND

**✅ ÚKOL Č. 2 – Programování periferií v assembleru**

**2.1 LED střídavě blikají**

asm

ZkopírovatUpravit

LED\_BLINK:

MOV P1, #00000001B ; LED1 on

ACALL DELAY

MOV P1, #00000010B ; LED2 on

ACALL DELAY

SJMP LED\_BLINK

DELAY:

MOV R2, #100

L2: MOV R1, #255

L1: DJNZ R1, L1

DJNZ R2, L2

RET

**2.2 LED blikají s různou frekvencí**

Použijeme dva různé delay podprogramy.

asm

ZkopírovatUpravit

LED\_BLINK\_FREQ:

SETB P1.0

ACALL DELAY\_SHORT

CLR P1.0

ACALL DELAY\_SHORT

SETB P1.1

ACALL DELAY\_LONG

CLR P1.1

ACALL DELAY\_LONG

SJMP LED\_BLINK\_FREQ

DELAY\_SHORT:

MOV R2, #50

; jako výše

RET

DELAY\_LONG:

MOV R2, #150

; jako výše

RET

**2.3 Zobrazení 0–9 na 7segmentovce**

Používáme pole (lookup tabulku) pro segmentové hodnoty.

asm

ZkopírovatUpravit

MOV DPTR, #SEGMENT\_TABLE

MOV R0, #0

DISPLAY\_LOOP:

MOV A, R0

MOVC A, @A+DPTR

MOV P2, A

ACALL DELAY

INC R0

CJNE R0, #10, DISPLAY\_LOOP

MOV R0, #0

SJMP DISPLAY\_LOOP

SEGMENT\_TABLE:

DB 3FH ; 0

DB 06H ; 1

DB 5BH ; 2

DB 4FH ; 3

DB 66H ; 4

DB 6DH ; 5

DB 7DH ; 6

DB 07H ; 7

DB 7FH ; 8

DB 6FH ; 9

**2.4 Logické funkce NOT, AND, OR s tlačítky a LED**

asm

ZkopírovatUpravit

LOGIC:

MOV A, P1

; A = xxxx xxBA (B = btn1, A = btn2)

; NOT tlačítko1 -> LED1

MOV C, ACC.2

CPL C

MOV P1.0, C

; AND btn1 & btn2 -> LED2

MOV C, ACC.2

ANL C, ACC.3

MOV P1.1, C

; OR btn1 | btn2 -> LED1

; MOV C, ACC.2

; ORL C, ACC.3

; MOV P1.0, C

SJMP LOGIC

**2.5 Funkce XOR s tlačítky a LED**

asm

ZkopírovatUpravit

LOGIC\_XOR:

MOV A, P1

MOV C, ACC.2

MOV B, ACC.3

MOV A, C

XRL A, B

MOV P1.0, A

SJMP LOGIC\_XOR

**✅ ÚKOL Č. 3 – Časové zpoždění**

**3.1 Zpoždění pomocí DJNZ – již výše uvedeno**

**3.2 Zpoždění pomocí časovače (např. Timer0)**

asm

ZkopírovatUpravit

DELAY\_TIMER:

MOV TMOD, #01H ; Timer0 mode1

MOV TH0, #0FCH ; vyšší byte

MOV TL0, #66H ; nižší byte (~50ms při 12MHz)

SETB TR0 ; start

WAIT\_T0:

JNB TF0, WAIT\_T0

CLR TR0

CLR TF0

RET

**✅ ÚKOL Č. 4 – Otestujte programy na UNI HW**

*Zapsat během zkoušky – kontrola funkčnosti na desce s MCU 8051.*

**✅ ÚKOL Č. 5 – Zpracování dokumentace**

**Obsah dokumentace:**

1. Název práce a zadání
2. Popis zapojení
3. Seznam použitých součástek
4. Zdrojové kódy
5. Popis funkce jednotlivých částí
6. Fotodokumentace testování (pokud je vyžadováno)
7. Závěr