

Nom : .....

Prénom: .....

---

# EXAMEN INTERMÉDIAIRE – ZWISCHENPRÜFUNG

## SOLUTION

*Information systems*

---

### **Anweisung / Consigne :**

Lesen Sie die Fragen gut durch und beantworten Sie diese **leserlich** auf den Aufgabenblättern. Alle Hilfsmittel (Dokumentation, Kurs, etc.) sind erlaubt, jedoch keine elektronischen Hilfen.

*Tipp: Verlieren Sie bei einzelnen Fragen nicht zu viel Zeit. Beantworten Sie zuerst die Fragen, die Ihnen keine Probleme stellen, und kommen Sie später auf die für Sie schwierigeren Fragen zurück. Die Skala ist unverbindlich.*

Lisez attentivement la donnée et répondez de manière **lisible** aux questions. Toute documentation est utilisable mais aucun moyen électronique ne l'est.

Un conseil : ne restez pas bloqués sur une question. Répondez tout d'abord aux questions avec lesquelles vous êtes à l'aise et revenez ensuite aux questions posant problème. Le barème indiqué est indicatif.

Question	Points	Score
Short questions	8	
ALU operations	6	
Input/Output	6	
Code comprehension	6	
Assembly programming	9	
Total:	35	

This exam has 5 questions, for a total of 35 points.

Rev 1.01

## Question 1 – Short questions (8 points)

*Diese Frage ist in unabhängige Aufgaben unterteilt. Die Punkte für jede Aufgabe sind links angegeben.*

*Cette question est séparée en plusieurs exercices indépendants. Le nombre de points pour chaque exercice est indiqué dans la marge.*

- [1 Pt] (a) *Warum ist der Literalwert, welcher in movlb gegeben ist, nur auf 4 Bits kodiert?*  
 Pourquoi le littéral donné dans l'instruction movlb est-il uniquement codé sur 4 bits ?

**Solution:** Car il y a 16 pages pour BSR (2 puissance 4)

- [1 Pt] (b) *Nehmen Sie an, dass ein spezieller PIC keinen Befehl GOTO hat. Wie wird das Ihren Assembler-Code beeinflussen (worauf müssen Sie aufpassen)?*  
 Sur un modèle particulier de PIC, l'instruction GOTO n'est pas disponible. Quel impact cela va-t-il avoir sur votre code assembleur (à quoi devrez-vous faire attention) ?

**Solution:** on ne pourra pas faire de sauts absolus dans tout l'espace de la mémoire programme. Il faudra faire des jumps relatifs uniquement (avec comme limite les 1023 du littéral)

- [1½ Pt] (c) *Was ist der Hauptunterschied zwischen einem GOTO und einem CALL?*  
 Quelle est la principale différence entre un GOTO et un CALL ?.

**Solution:** Un goto permet de sauter à un endroit en modifiant le PC. Un call permet en plus de revenir, à l'aide d'un return, de là où le call s'est effectué. C'est la stack enregistre l'endroit où l'on doit revenir.

- [1½ Pt] (d) *Erklären Sie warum der Befehl BC einen oder zwei Zyklen dauern kann.*  
 Expliquez pourquoi l'instruction BC peut prendre un ou deux cycles.

**Solution:** Cette instruction prend un cycle si le carry est 0 et deux autrement (si le saut est pris). Si l'on doit prendre le saut, il faut d'abord flusher le pipeline et c'est pourquoi il faut 2 cycles.

- [3 Pt] (e) *Wahr oder falsch? | Vrai ou faux ?*

*Es ist möglich, mit einem einzigen Befehl Werte zu addieren, die sich an zwei beliebigen Adressen befinden.*

Il est possible avec une seule instruction d'additionner deux valeurs contenues à des adresses quelconques.

True	False
<input type="checkbox"/>	<input checked="" type="checkbox"/>

*Einige Befehle des PIC18F sind auf 8 Bits kodiert*

Certaines instructions sur PIC18F sont codées sur 8 bits.

True	False
<input type="checkbox"/>	<input checked="" type="checkbox"/>

*Literalwerte in Assembler dürfen kein Vorzeichen haben.*

Les valeurs littérales en assembleur ne peuvent pas être signées.

True	False
<input type="checkbox"/>	<input checked="" type="checkbox"/>

*Literalwerte sind in die Befehle kodiert.*

Les valeurs littérales sont encodées dans les instructions.

True	False
<input checked="" type="checkbox"/>	<input type="checkbox"/>

*movf foo, bar kopiert den Inhalt vom foo zu bar.*

movf foo, bar permet de déplacer le contenu de foo dans bar.

True	False
<input type="checkbox"/>	<input checked="" type="checkbox"/>

*Das IR Register beinhaltet immer eine Kopie des Inhalts der Adresse PC-2.*

Le registre IR contient toujours une copie de ce qu'il y avait à l'adresse PC-2.

True	False
<input type="checkbox"/>	<input checked="" type="checkbox"/>

**Question 2 – ALU operations (6 points)**

- [4 Pt] (a) *Vervollständigen Sie den Zustand der ALU-bits nach jedes Befehls. X ist der Anfangszustand des Bits.*  
 Complétez l'état des bits de l'ALU après chaque instruction. X est l'état des bits au départ.

Instruction	C	DC	Z	OV	N
movlw 0	x	x	x	x	x
movwf STATUS	0	0	0	0	0
clrf LATH	0	0	1	0	0
decf LATH	0	0	0	0	1
decf LATH	1	1	0	0	1

- [1 Pt] (b) 1) *Was ist der Inhalt des Registers a nach folgendem Kode?*  
 Quel est le contenu du registre à l'adresse a après l'exécution du code suivant ?

```

1  a equ 0x21
2
3  movlw (0xa9 + 0x1)
4  movwf a
5  xorwf 0x21, F

```

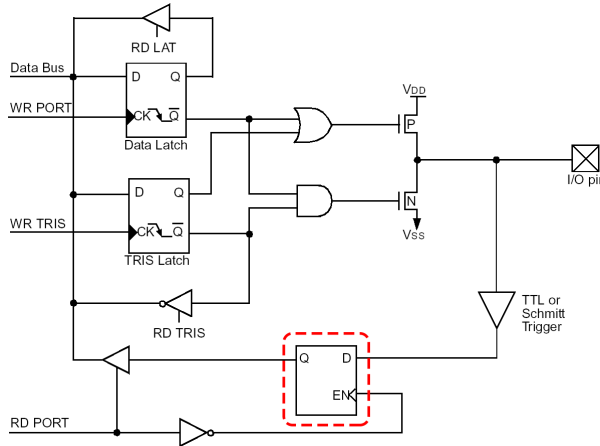
**Solution:** 0

- [1 Pt] 2) *Haben die Zeilen 3 oder 4 einen Einfluss auf das Resultat vom Befehl am Zeile 5? Erklären Sie warum.*  
 Le résultat de l'instruction à la ligne 5 est-il influencé par les lignes 3 ou 4? Expliquez pourquoi.

**Solution:** Yes, the instruction at line 4 copies the content of WREG to the address 0x21. As a result, whatever there is in register at location 0x21, the result is always 0 (a xor a is always 0, for every a).

### Question 3 – Input/Output (6 points)

Gegeben sei das folgende Schema eines Pins: | Soit le schéma de la pin suivante :



[2 Pt] (a) Erklären Sie, wozu das eingekreiste Flipflop dient. Expliquez à quoi sert la bascule entourée.

**Solution:** Elle sert à latcher l'entrée de la pin de manière synchrone afin d'éviter qu'elle bouge lorsque le processeur la lit. Cela évite les problèmes de metastabilité et assure que les setup/hold time de l'entrée soient corrects en sortie.

[4 Pt] (b) Der Pin RB0 des PICs ist über einen Taster mit der Masse verbunden. Der Pin RB1 ist über eine LED mit V<sub>DD</sub> verbunden. Schreiben Sie den kompletten Assemblercode um den Zustand der LED jedesmal zu verändern, wenn auf den Taster gedrückt wird.

La pin RB0 du PIC est connectée à la masse à travers un bouton. La pin RB1 est connectée à V<sub>DD</sub> à travers une LED. Écrivez le code assembleur complet permettant de changer l'état de la led à chaque fois que l'on presse le bouton.

**Solution:**

```

1  movlw 0xFD          ; important is 'xxxxxx01
2  movwf TRISB        ; port b direction
3  bcf INTCON2,RBPU   ; activate pull-up on port b (optional)
4
5  loop:
6  btfsc PORTB,0      ; wait button pressed
7  goto loop
8
9  bt_pressed:
10 btg LATB,1         ; toggle led
11
12 wait_released:
13 btfss PORTB,0      ; wait button released
14 goto wait_released
15 goto loop          ; do it again
    
```

### Question 4 – Code comprehension (6 points)

[2 Pt] (a) Vervollständigen Sie folgende Tabelle nach der Ausführung jedes Befehls. Compléter le tableau ci-dessous après l'exécution de chaque instruction

[2 Pt] (b) Wie sind die zwei ersten Befehle kodiert? (die Adresse von WREG ist 0xFE8) Donnez l'encodage des deux premières instructions (l'adresse de WREG est 0xFE8).

Instruction	WREG	RAM 0X00	RAM 0X01
	0xaa	0x12	0xb7
movf 0X01, 0	0xb7	0x12	0xb7
movff 0, WREG	0x12	0x12	0xb7
bsf 0, 3	0x12	0x1A	0xb7
movlw 0x01	0x01	0x1A	0xb7
movff WREG, 1	0x01	0x1A	0x01
addwf 0x01, 0	0x02	0x1A	0x01

**Solution:** 0x5001  
0xC000 FFE8

- [2 Pt] (c) Man möchte den Inhalt von Adresse 0x340 zur Adresse 0x441 kopieren. Schlagen Sie zwei Möglichkeiten vor, einmal mit BSR und einmal ohne.  
On désire déplacer le contenu de l'adresse 0x340 à l'adresse 0x441. Proposez deux manières de le faire, une fois sans BSR et une fois avec.

**Solution:**

```

1 ; First solution
2 movlb 3
3 movf 0x40, W, BANKED
4 movlb 4
5 movwf 0x41, F, BANKED
6
7 ; Other solution
8 movff 0x34, 0x441
    
```

### Question 5 – Assembly programming (9 points)

- (a) Ein 7-Segment-Display besteht aus unabhängigen LEDs, welche man mit einem uC kontrollieren kann. Man möchte die untenstehende Sequenz auf einem solchen Display anzeigen, wobei der Übergang von einem Zustand zum nächsten per Knopfdruck ausgelöst wird.

Un display 7 segments est composé de leds indépendantes que l'on peut contrôler à l'aide d'un uC. On désire afficher la séquence ci-dessous sur un tel display, le passage d'un état à l'autre se faisant à chaque appui sur un bouton.

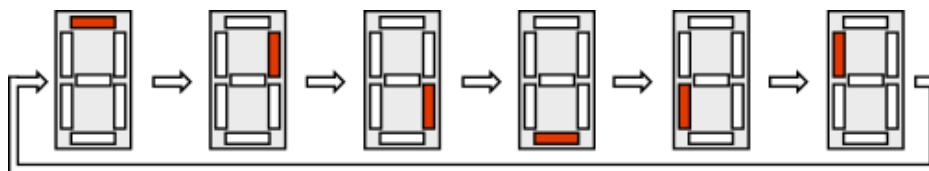
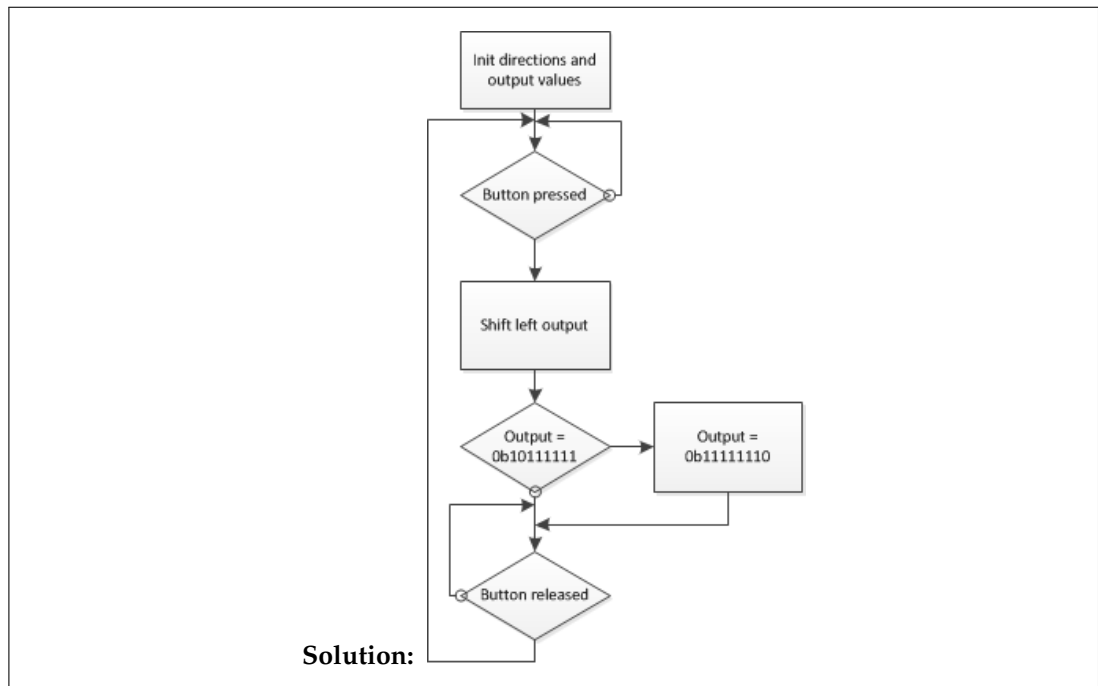
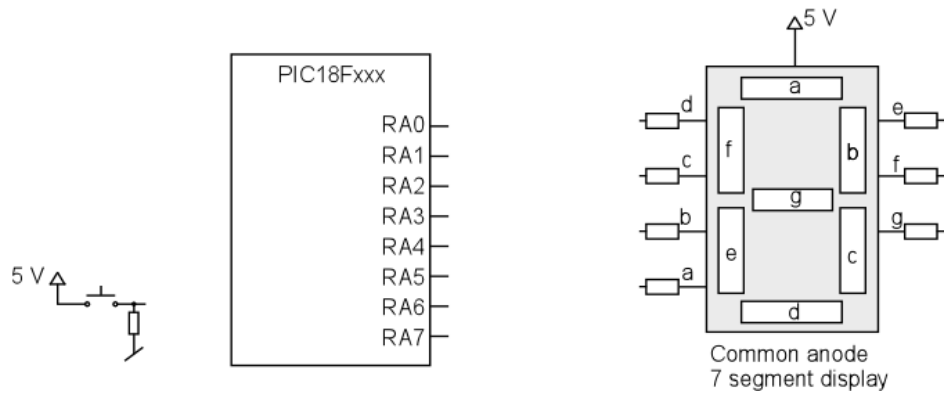


Figure 1 – 7 segments sequence

- [2 Pt] 1) Vervollständigen Sie das untenstehende Elektronikschemata um den PIC mit dem Display zu verbinden.  
Complétez le schéma électrique ci-dessous pour connecter le PIC avec le display.
- [2 Pt] 2) Zeichnen Sie das Flussdiagramm des Programms, welches die geforderte Funktion ausführt.  
Dessinez le diagramme de flux du programme effectuant la fonction demandée



[5 Pt]

3) Schreiben Sie den entsprechenden Assemblercode. Écrivez le code assembleur complet correspondant.

**Solution:**

```
1      MOVLW   B'10000000'  
2      MOVWF   TRISA  
3      MOVLW   B'11111110'  
4      MOVWF   LATA  
5  BtPress:  
6      BTFSS   PORTA,7  
7      GOTO    BtPress  
8      RLNCF   LATA  
9      BTFSC   LATA,6  
10     GOTO    BtRelease  
11     MOVLW   B'11111110'  
12     MOVWF   LATA  
13  BtRelease:  
14     BTFSC   PORTA,7  
15     GOTO    BtRelease  
16     GOTO    BtPress
```

---

*Fin*

---