

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	
Arithmetic operations	4	
Code comprehension	4	
IO pins	4	
Writing assembly code	6	
Everything together	6	
Total:	32	

This exam has 6 questions, for a total of 32 points.

Rev 1.00

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) Was ist der Inhalt von W nach dem folgenden Code (in der Adresse 0x20 ist der Wert 3 gespeichert).
Quel est le contenu de W après le code suivant, sachant l'adresse 0x20 contient 3.

```
movlw 08
movwf 20
subwf 20, w
```

Solution: 0

- [1 Pt] (b) Warum gibt es 3 bits für b in der Kodierung des Befehls bcf?
Pourquoi dans l'encodage de l'instruction bcf il y a-t-il 3 bits pour b ?

Solution: 0

- [1 Pt] (c) Ist der Befehl addlw 384 gültig? Begründen Sie Ihre Antwort!
L'instruction addlw 384 est-elle valide ? Justifiez votre réponse en détail!

Solution: No, it's not doable. The literal value here is more than 8 bits wide, therefore it cannot be stored into an instruction like this.

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

PIC ist ein Harvard Prozessor.

Le PIC est un processeur de type Harvard.

True	False
☒	☐

Die PIC-Pipeline erlaubt es, mehr Befehle pro Sekunde auszuführen.

Le pipeline du PIC lui permet d'exécuter plus d'instructions par seconde.

True	False
☒	☐

Die PIC-Pipeline verlangt es, Sprung-Befehle mit zwei Instruktionen zu kodieren.

Le pipeline du PIC oblige à coder les sauts sur deux instructions.

True	False
☐	☒

Es gibt die selbe Anzahl Speicherstellen für das Programm wie für die Daten.

Il y a le même nombre de cases mémoire pour stocker le programme que pour stocker les données.

True	False
☐	☒

Zum Zeitpunkt des Zyklus t führt der Prozessor immer den Befehl aus, welcher zum Zeitpunkt des Zyklus $t - 1$ gefetched wurde.

Au temps de cycle t , le processeur exécute toujours l'instruction fetchée au temps de cycle $t - 1$

True	False
☐	☒

Der movwf Befehl braucht das d Bit nicht, weil das Ziel dieses Befehls immer das WREG Register ist.

L'instruction movwf n'a pas besoin du bit d car la destination de cette instruction est toujours le registre WREG.

True	False
☐	☒

- [1 Pt] (e) Warum ist es mit einem PIC nicht möglich, den Inhalt von zwei Speicherposition in einem Zyklus zu addieren? Begründen Sie Ihre Antwort.
Pourquoi n'est-il pas possible d'additionner en un cycle le contenu de deux positions mémoire sur PIC ? Justifiez votre réponse.

Solution: Because one of the inputs is directly connected to W

- [1 Pt] (f) Was ist das Ziel des Parity-bits im seriellen Protokoll (RS232), welches wir im Labor benutzt haben?
A quoi sert le bit de parité dans le protocole série sur RS232 vu au laboratoire ?

Solution: To check if there was an error during transmission.

Question 2 – Arithmetic operations (4 points)

Das Register W ist -1 und die Adresse 0x33 enthält den Wert 127. Geben Sie den Wert der Flags Z, C, OV, N nach der Ausführung der folgenden Operationen (jede Operation ist unabhängig zu betrachten).

Le registre W contient la valeur -1 et l'adresse mémoire 0x33 contient la valeur 127. Donnez les valeurs de Z, C, OV, N après exécution des instructions suivantes (chaque opération est indépendante).

- (a) (W) + (0x33) (a) **Z=0, C=1, OV=0, N=0**
- (b) (0x33) - (W) (b) **Z=0, C=0, N=1, OV=1**
- (c) (W) + 1 (c) **Z=1, C=1, N=0, OV=0**
- (d) (W) AND (0x33) (d) **All 0**

Question 3 – Code comprehension (4 points)

Gegeben sei die ASCII-Tabelle: | Soit la table ASCII :

00h	␣ (nul)	10h	► (dle)	20h	□	30h	0	40h	@	50h	P	60h	‘	70h	p
01h	Ⓢ (soh)	11h	◄ (dc1)	21h	!	31h	1	41h	A	51h	Q	61h	a	71h	q
02h	Ⓢ (stx)	12h	↑ (dc2)	22h	"	32h	2	42h	B	52h	R	62h	b	72h	r
03h	♥ (etx)	13h	!! (dc3)	23h	#	33h	3	43h	C	53h	S	63h	c	73h	s
04h	♦ (eot)	14h	¶ (dc4)	24h	\$	34h	4	44h	D	54h	T	64h	d	74h	t
05h	♣ (enq)	15h	§ (nak)	25h	%	35h	5	45h	E	55h	U	65h	e	75h	u
06h	♣ (ack)	16h	— (syn)	26h	&	36h	6	46h	F	56h	V	66h	f	76h	v
07h	• (bel)	17h	‡ (etb)	27h	'	37h	7	47h	G	57h	W	67h	g	77h	w
08h	■ (bs)	18h	↑ (can)	28h	(	38h	8	48h	H	58h	X	68h	h	78h	x
09h	␣ (tab)	19h	↓ (em)	29h	)	39h	9	49h	I	59h	Y	69h	i	79h	y
0Ah	☐ (lf)	1Ah	(eof)	2Ah	*	3Ah	:	4Ah	J	5Ah	Z	6Ah	j	7Ah	z
0Bh	♂ (vt)	1Bh	← (esc)	2Bh	+	3Bh	;	4Bh	K	5Bh	[	6Bh	k	7Bh	{
0Ch	Ⓢ (np)	1Ch	↳ (fs)	2Ch	,	3Ch	<	4Ch	L	5Ch	\	6Ch	l	7Ch	
0Dh	♂ (cr)	1Dh	↔ (gs)	2Dh	-	3Dh	=	4Dh	M	5Dh	]	6Dh	m	7Dh	}
0Eh	♂ (so)	1Eh	▲ (rs)	2Eh	.	3Eh	>	4Eh	N	5Eh	^	6Eh	n	7Eh	~
0Fh	✱ (si)	1Fh	▼ (us)	2Fh	/	3Fh	?	4Fh	O	5Fh	_	6Fh	o	7Fh	△

- [3 Pt] (a) Was zeigt der folgenden Code auf dem PC-Terminal, welches mit dem seriellen Port zum PIC verbunden ist? Man nimmt an, dass der Port richtig initialisiert ist.
Qu'affiche le code suivant dans le terminal du PC relié au port série ? On considère que les initialisations du port sont effectuées correctement.

```
1    movlw .103
2    call  send
3    movlw 0x6f
```

Listing continues on next page →

...continuing from previous page

```

4      call    send
5      call    send
6      andlw   0xf4
7      call    send
8
9      send:
10     btfss   PIR1, TXIF
11     goto    send
12     movff   WREG, TXREG
13     return

```

Solution: good

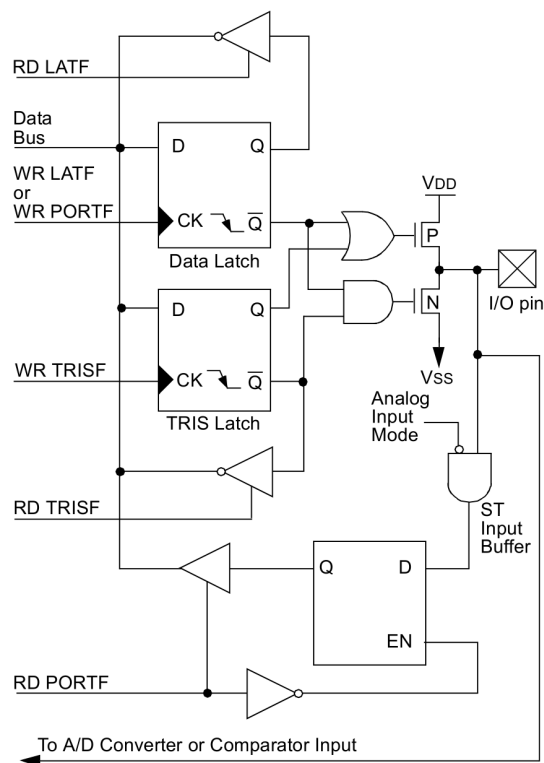
[1 Pt]

- (b) *Erklären Sie die Bedeutung der Zeile 10.*
Expliquez à quoi sert la ligne 10 du code.

Solution: To wait until the transmit register is free for a new data send.

Question 4 – IO pins (4 points)

Gegeben sei das Schema von einem IO-Pin des Ports F. | Soit le schéma d'une pin d'entrée-sortie du port F.

 $[^{1/2}\text{Pt}]$

- (a) Was ist die Quelle der Signale RD LATF und RD PORTF?
Quelle est la source des signaux RD LATF et RD PORTF?

Solution: The decode unit of the CPU

 $[^{1/2} \text{Pt}]$

- (b) Was ist das Ziel des Flipflops, welches mit dem Signal ST Input Buffer verbunden ist?
À quoi sert la bascule D qui est connecté au signal ST Input Buffer?

Solution:

- [1 Pt] (c) Was liest man auf *PORTF* wenn der Port im Analogmodus konfiguriert ist?
Que lit-on sur le *PORTF* si le port est configuré en mode analogique?

Solution: Only zeroes.

- [2 Pt] (d) Erklären Sie (mit einem Beispiel) wozu die Signale *WR LATF*, *WR PORTF*, *RD LATF* und *RD PORTF* dienen.
Expliquez à l'aide d'un exemple à quoi servent les signaux *WR LATF*, *WR PORTF*, *RD LATF* et *RD PORTF*.

Solution:

Question 5 – Writing assembly code (6 points)

- [2 Pt] (a) Man möchte den Inhalt des Registers *LATB*, welches die Adresse *0xF8A* hat, in *W* kopieren. Schreiben Sie den Kode, um diese Operation zu realisieren, ohne den Befehl *movff* oder den Modus *access bank* zu benutzen.
On désire copier le contenu du registre *LATB*, qui se trouve à l'adresse *0xF8A*, dans *W*. Donnez le code nécessaire pour réaliser cette opération sans utiliser l'instruction *movff* ni le mode *access bank*.

Solution:

```
movlw 0xaa
movff WREG, LATB
movlb 0xf
movf 0x8A, W, BANKED
```

- [4 Pt] (b) Man nehme an, dass der Port B als Ausgang konfiguriert ist. Das Register *W* enthält die Bits *0b000000abc* (wobei *a*, *b*, *c* Bits repräsentieren). Schreiben Sie ein Programm, welches die drei niedrigstwertigen Bits (LSB) in *W* zum *PORTB* kopiert (ohne dabei die anderen Bits von *PORTB* zu verändern). Hinweis: Benutzen sie bitweise Operatoren (AND, OR, ...).

On suppose le port B configuré en sortie. Le registre *W* contient les bits *0b000000abc* (avec *a*, *b*, *c* des bits). Écrivez un programme permettant de recopier le contenu des 3 bits de poids faible de *W* vers *PORTB* sans changer les autres bits de *PORTB*. Conseil: Utilisez judicieusement les instructions sur les bits (AND, OR, ...).

Solution: For instance, a possible solution is:

```
comf WREG, W
andwf 0x00, F
comf WREG, W
iorwf 0x00, F
movff 0x00, WREG
```

Question 6 – Everything together (6 points)

Gegeben sei folgender Code (Ausgabe des Assemblers) | Soit le code suivant produit par l'assembleur:

```
0000 goto    0x26
...
0026 clrf    PORTB
0028 movlw   0x10
002a movwf   TRISB, 0
002c bsf     PORTB, 1, 0
002e call    0x39
```

- [1 Pt] (a) Wenn Leds mit den vier LSBs verbunden sind (zwischen dem Pin und 3V), welche davon sind nach der Ausführung des obenstehenden Codes eingeschaltet?
Si des leds sont connectées sur les 4 bits de poids faible (entre la pin et 3V), lesquelles seront allumées après l'exécution du code ci-dessus?

Solution: Leds on pin 0, 2 and 3 are on.

- [1 Pt] (b) Das Register *PORTB* hat die Adresse *0xF81*. Schreiben den Maschinencode für die Befehle bei den Adressen *0x26* und *0x28*.
Le registre *PORTB* se trouve à l'adresse *0xF81*. Donnez l'encodage en code machine des instructions se trouvant aux adresses *0x26* et *0x28*.

Solution: 0x6A81
0x0E10

- [4 Pt] (c) Beschreiben Sie den Zustand der CPU nach der Ausführung der Adressen *0x2a*, *0x2c* und *0x2e*. Benutzen Sie dazu die folgenden Diagramme (eines pro Instruktion). Vergessen Sie die Werte auf den Bussen nicht.
Compléter l'état du CPU après exécution des instructions aux adresses *0x2a*, *0x2c* et *0x2e* sur les figures suivantes (une figure par instruction). N'oubliez pas de mettre les valeurs sur les bus.

△ Turn page →

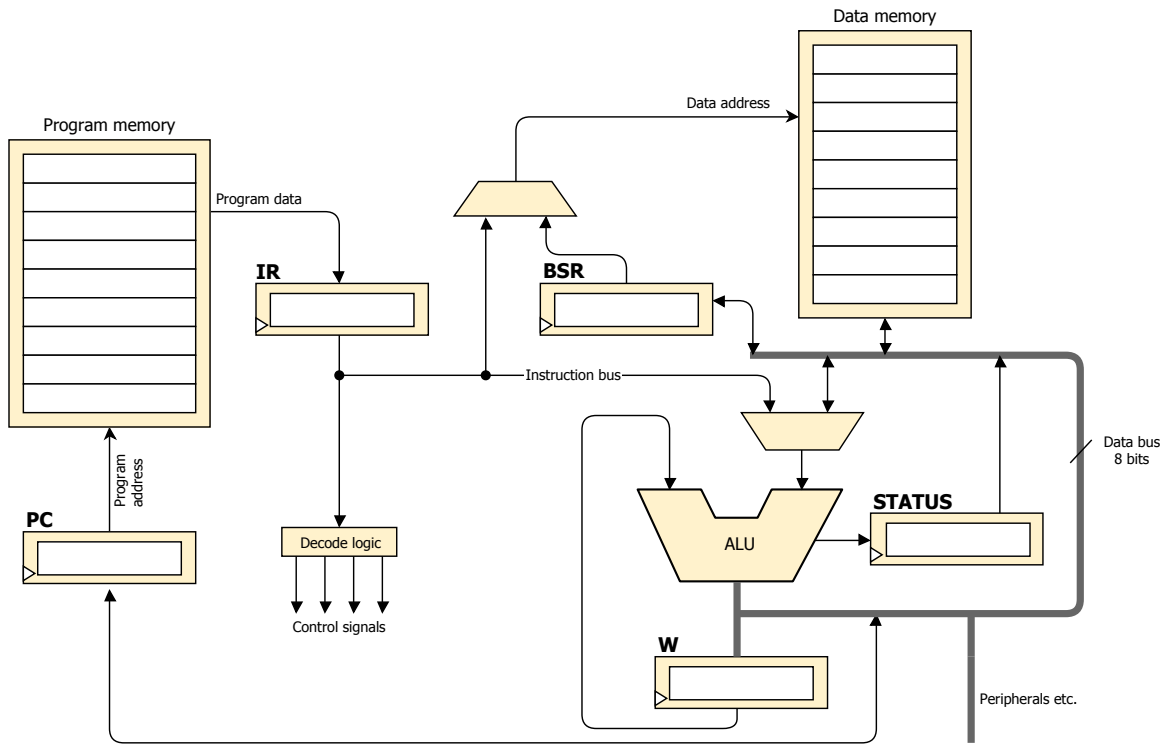


Figure 1 – Status after executing `movwf TRISB, 0`

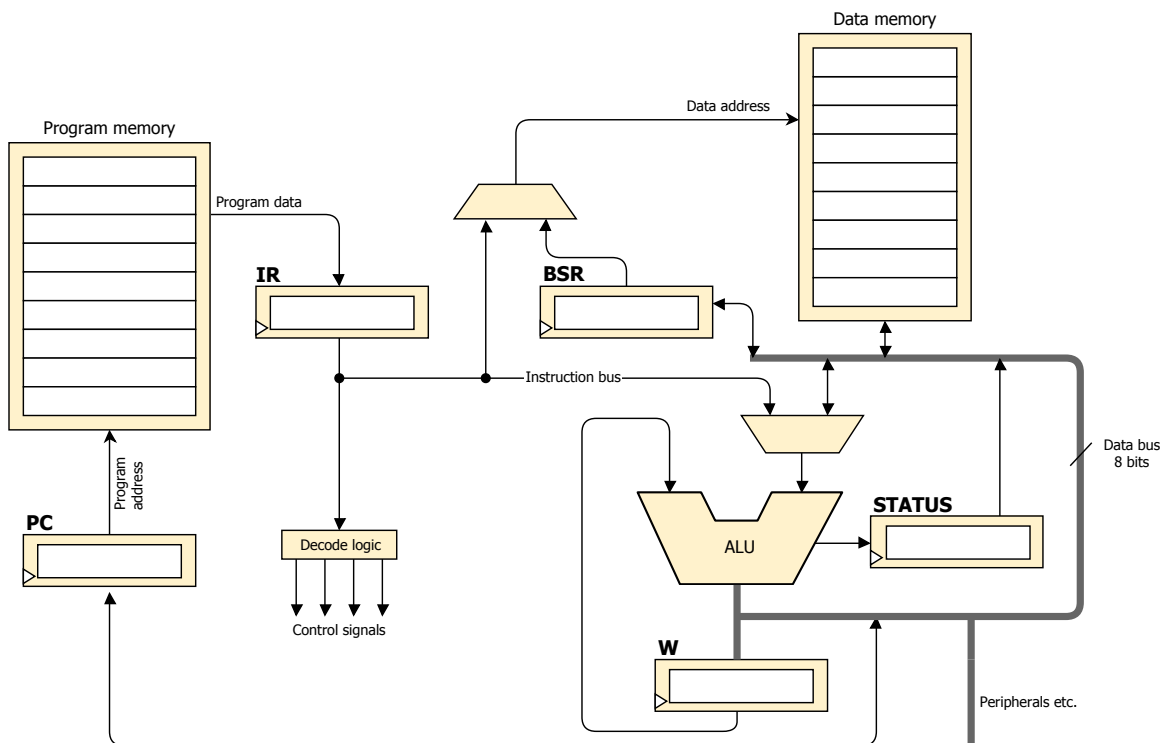


Figure 2 – Status after executing `bsf PORTB, 1, 0`

Fin

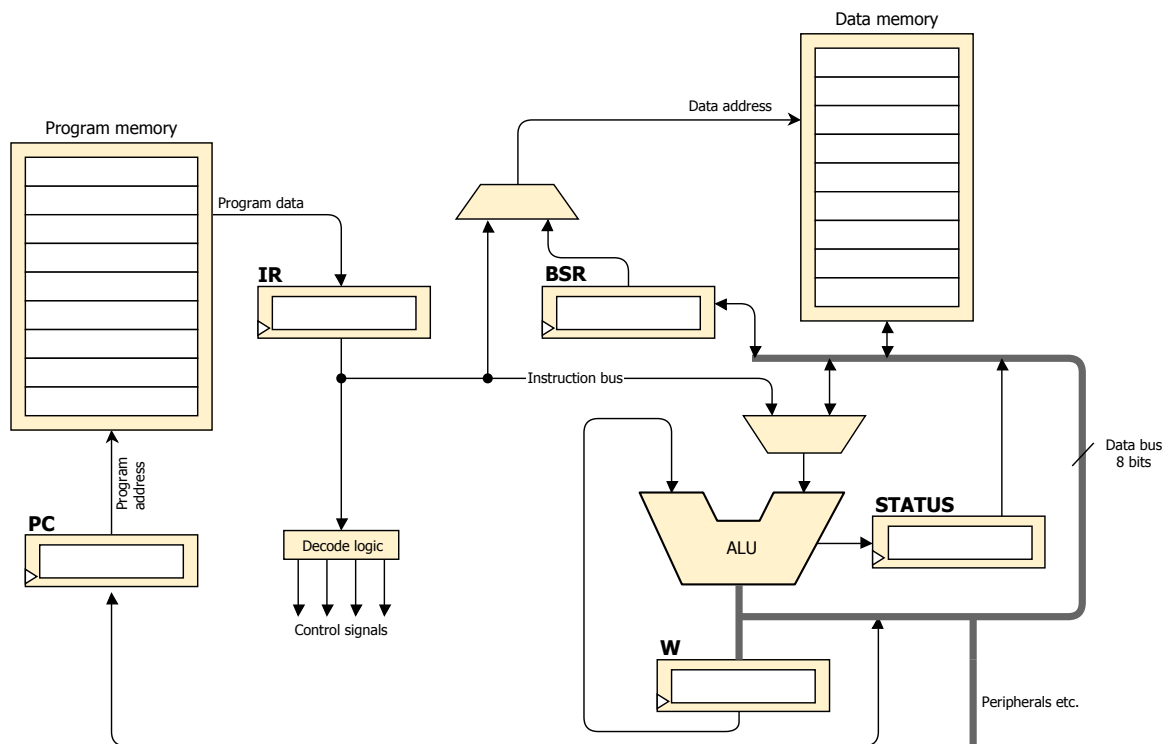


Figure 3 – Status after executing call 0x39

