Rechnerarchitektur, Kapitel 1.4

2. Lauflicht

1. Dieses Programm zeigt ein Lauflicht am Ausgabe-Port (OUT):
Das Lauflicht ist ein einzelnes gesetztes Bit, das stetig von unten nach oben wandert, wenn es oben rausrutscht, wird es unten wieder eingefügt.
Gesucht:

a) Struktogramm

A. Den Quotienten in den Input-Port einlesen	
B. Den Quotient mit 2 multiplizieren	
IF REST der Multiplikation == 0	
THEN	ELSE
Das Resultat auf dem Output-Port ausgeben lassen	Springe zu Punkt A
Springe zu Punkt B	

b) Assemblerprogramm

Programm					
Assemblercode		Maschinencode		Zyklen	Kommentar
Label	Mnemonic	1. Wort	2. Wort (Daten, Adresse)		
A	IN	1000	-	7	
	OUT	1011	-		
B	ASL	1110	-	5	
	JC A	0111	? (0000)	8	
	OUT	1011	-	7	
	JMP B	0110	? (0001)		

c) Maschinenprogramm

ROM-Adresse	ROM-Inhalt	
	Beim 1. Durchlauf sind die Sprungadressen noch unbekannt	Beim 2. Durchlauf lassen sich die Sprungadressen einsetzen.
0 (0000)	1000	1000
1 (0001)	1011	1011
2 (0010)	1110	1110
3 (0011)	0111	0111
4 (0100)	?	0000
5 (0101)	1011	1011
6 (0110)	0110	0110
7 (0111)	?	0001
8 (1000)		
9 (0001)		

3. * Gerade oder ungerade Zahl?

2. Anforderungen:

Das Programm stellt bei einer eingegebenen Zahl (IN) fest, ob diese gerade oder ungerade ist.

- Ist die Zahl gerade, lautet die Ausgabe (OUT): 0000
- Ist die Zahl ungerade, lautet die Ausgabe (OUT): 1111

Gesucht:

a) Struktogramm

---

b) Assemblerprogramm

Programm					
Assemblercode		Maschinencode		Zyklen	Kommentar
Label	Mnemonic	1. Wort	2. Wort (Daten, Adresse)		
	IN	1000	-		Einlesen IN → R0
	ASL	1110	-		
	ASL	1110	-		
	ASL	1110	-		
	ASL	1110	-		Jetzt R0 == 0
	JNC A	1100	? (1001)		
	MVI R0 15	0100	1111		1111 → R0
A	OUT	1011	-		R0 → OUT

c) Maschinenprogramm

ROM-Adresse	ROM-Inhalt	
	Beim 1. Durchlauf sind die Sprungadressen noch unbekannt	Beim 2. Durchlauf lassen sich die Sprungadressen einsetzen.
0 (0000)	1000	1000
1 (0001)	1110	1110
2 (0010)	1110	1110
3 (0011)	1110	1110
4 (0100)	1110	1110
5 (0101)	1100	1100
6 (0110)	?	1001
7 (0111)	0100	0100
8 (1000)	1111	1111
9 (0001)	1011	1011

4. Maschinencode-Analyse I

3. Analysieren Sie folgenden Maschinencode. Was passiert mit der Eingabe (IN)?

ROM-Adresse:	ROM-Inhalt
0000	1000 → Input einlesen (IN)
0001	1001 → Register 0 in Register 1 kopieren (MOV R1, R0)
0010	1110 → Register 0 links schieben, Überlauf in carry (ASL)
0011	1101 → Register 1 + Register 0 (ohne carry) (AND R1)
0100	1011 → Register 0 in Output –Port (OUT)

➔ *Input einlesen, Multiplikation mit 3, Ausgeben*

5. * Maschinencode-Analyse II

4. Analysieren Sie folgenden Maschinencode. Was für Ausgabe-Folgen (OUT) resultieren in Abhängigkeit eines bestimmten Eingabe-Wertes (IN)? Können Sie das Ergebnis nachvollziehen?

ROM-Adresse:	ROM-Inhalt	
0000	1000	→ IN
0001	1001	→ MOV R1, R0
0010	1110	→ ASL
0011	1101	→ ADD R1
0100	1011	→ OUT
0101	0110	→ JMP
0110	0001	→#0001

➔ *Input einlesen, Multiplikation mit 3, Resultat ausgeben und zur Multiplikation hochspringen und dies wiederholen.*