Aufgabe 1: ***DSP 56001 Register-Adressierungsarten***

Der X-Speicher eines DSP56001 sei gemäß Tabelle 1.1 belegt.

Adresse	Inhalt
$14F	$555
$150	$444
$151	$333
$152	$222
$153	$111

Tabelle 1.1: Speicherbelegung

a) Es ist M0=$FFFF. Geben Sie für jeden der in Tabelle 1.2 aufgeführten Befehle den Inhalt der angegebenen Register nach Ausführung des Befehles an! Tragen Sie die jeweilige Bezeichnung des Adressierungsmodus in die Spalte „Bezeichnung“ ein!

	R0	N0	R0	A	Bezeichnung
`MOVE X:(R0)+,A`	**$150**	**$2**			
`MOVE X:-(R0),A`	**$150**	**$2**			
`MOVE X:(R0)+N0,A`	**$150**	**$2**			
`MOVE X:(R0+N0),A`	**$150**	**$2**			

Tabelle 1.2: Befehlsfolge für den DSP56001

b) Ist der folgende Befehl beim DSP56001 zulässig ?
`MOVE X:(R0+N4),A`
Begründen Sie Ihre Antwort !

c) Durch Programmierung des Modifier-Registers M0 lassen sich beim DSP56001 drei verschiedene Grundarten der Adressberechnung einstellen. Nennen Sie diese und geben Sie jeweils eine typische Anwendung an !

Aufgabe 2: ***Register und Speicher des DSP***

Ein DSP führt das folgende Programmstück aus:

```
MOVE    #$020000,X0     ; schreibe eine 24 bit Zahl im HEX-Format in X0
MOVE    #$020000,Y0
CLR     A
DO      #128,ENDE
MAC     X0,Y0,A
ENDE
```

a) Tragen Sie in der folgenden Tabelle im Hexadezimalformat die Inhalte in die Teilstücke des Akkumulators A ein, die sich nach Abarbeitung der DO-Schleife des Programms ergeben.

A2	A1	A0
$	$	$

Hinweise:

- *ein $-Zeichen nach dem # bedeutet, dass eine Zahl im Hexadezimal-Format folgt*
- *kein Zeichen nach dem # bedeutet, dass eine Dezimalzahl folgt*
- *Für die Abarbeitung des obigen Programms ist die bei DSPs übliche Fraktalzahlen-interpretation anzusetzen.*

Im X-Datenspeicher eines DSP soll mit Hilfe des AGU-Registers M1 ein Ringpuffer der Länge 41 eingerichtet werden, dessen Untergrenze bei der Adresse 64 liegt.

b) Welche Zahl – in Dezimaldarstellung – ist in M1 einzutragen?
c) Welchen Wert – in Dezimaldarstellung – hat die Obergrenze des Ringpuffers?

Jetzt werde mit dem oben eingestellten Ringpuffer der Länge 41 das folgende Programmstück abgearbeitet:

```
MOVE    #100,R1
MOVE    #2,N1
REP     #4              ; wiederhole den folgenden Befehl viermal
MOVE    X:(R1)+N1,X0
STOP                    ; DSP in Stopp-Zustand versetzen
```

d) Welchen Inhalt hat jetzt das Pointerregister R1?

Aufgabe 3: ***Datenverarbeitung mit dem DSP***

Ein DSP führt mit Daten, die über einen A/D-Wandler eingelesen werden, den untenstehenden Programmausschnitt aus.
Der externe A/D-Wandler ist unter der X-Datenspeicher-Adresse $1000 angeschlossen und liefert in fünf aufeinander folgenden Maschinenzyklen die folgenden Werte in Dezimaldarstellung:

1/32, 1/16, 1/8, 1/4, 1/2

Im Y-Datenspeicher ist ein Ringpuffer der Länge 5 ab Adresse $40 mit folgendem Inhalt angelegt:

Adresse	Inhalt (dezimal)
$40	1/2
$41	1/4
$42	1/8
$43	1/16
$44	1/32

Es wird der folgende Programmausschnitt betrachtet:

```
MOVE   #$1000,R1          ; Werte von externem ADW einlesen
MOVE   #$40,R5            ; Pointer auf internen Y-Datenspeicher

MOVE   X:(R1),X0          ; Startwert in X0
MOVE   Y:(R5)+,Y0         ; Startwert in Y0
CLR    A                  ; Akku A löschen

DO     #5,ENDE
MAC    X0,Y0,A    X:(R1),X0    Y:(R5)+,Y0

ENDE
```

a) Tragen Sie in der nachfolgenden Tabelle im Hexadezimalformat die Inhalte in die Teilstücke des Akkumulators A ein, die sich nach Abarbeitung des Programmausschnitts ergeben!

A2	A1	A0
$	$	$

b) Welche Zahl – in Dezimaldarstellung – ist in welches Modifier-Register M*n* einzutragen, damit der benötigte Ringpuffer eingerichtet wird?

c) Erklären Sie, weshalb bei der MAC-Instruktion im obigen Programmausschnitt in der Operandengruppe `X:(R1),X0` kein + Zeichen hinter (R1) steht!

Aufgabe 4: ***Korrelation auf einem DSP***

Der DSP soll zur Berechnung einer Korrelationsfunktion eingesetzt werden. Dazu soll die Eingangssignalfolge $x(k)$ mit einer Referenzsignalfolge $y(k)$ wie folgt verknüpft werden:

$$z = \sum_{k=0}^{255} x(k) \cdot y(k) \,. \tag{4.1}$$

Die 256 Abtastwerte des Eingangssignals $x(k)$ mit $k = 0 \ldots 255$ liegen zu Beginn im X-Datenspeicher ab Adresse 0. Die 256 Abtastwerte des Referenzsignals $y(k)$ mit $k = 0 \ldots 255$ liegen im Y-Datenspeicher ab Adresse 0 (siehe Bild 4.1).

Adresse	X-Speicher	Zeiger	Adresse	Y-Speicher	Zeiger
	:			:	
	x (0)			y (0)	
	x (1)			y (1)	
	:			:	
	x (254)			y (254)	
0:	x (255)	← R0	0:	y (255)	← R4

Bild 4.1: Speicherbelegung vor dem ersten Unterprogrammaufruf

a) Geben Sie die zur Initialisierung der Adressberechnung notwendigen Assemblerbefehle an, so dass im X-Speicher und im Y-Speicher jeweils ein Ringspeicher mit 256 Speicherzellen ab Adresse 0 initialisiert wird! Verwenden Sie das Register R0 als Zeiger auf die im X-Speicher abgelegten Werte und das Register R4 als Zeiger auf die im Y-Speicher abgelegten Werte!

b) Schreiben Sie ein Unterprogramm, das den Wert z nach Gleichung (4.1) berechnet! Das Ergebnis z soll nach dem Unterprogrammende im Register A stehen. Vergessen Sie nicht, die Pointer R0 und R4 für den nächsten Unterprogrammaufruf richtig zu setzen! Außerdem muss der neue Eingangswert x (255) im X-Speicher für den nächsten Aufruf des Unterprogramms aktualisiert werden. Dieser Wert steht im Register X1.