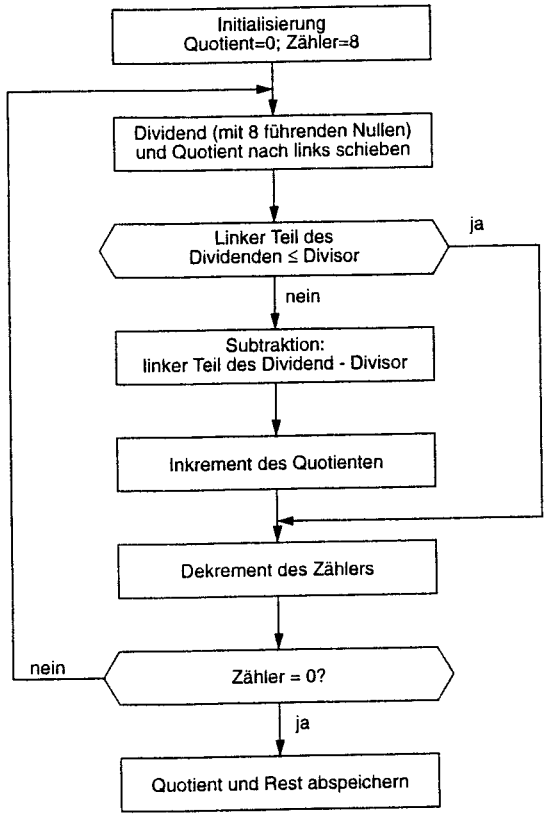
Lösungsvorschlag Aufgabe H1:

Die Division wird hier auf eine Kombination von Subtraktionen und Schiebeoperationen zurückgeführt. Hierbei wird eine Hilfsvariable zum Zählen der Subtraktionsschritte benötigt. Diese wird hier in der Speicherzelle $0500 gespeichert.

a)

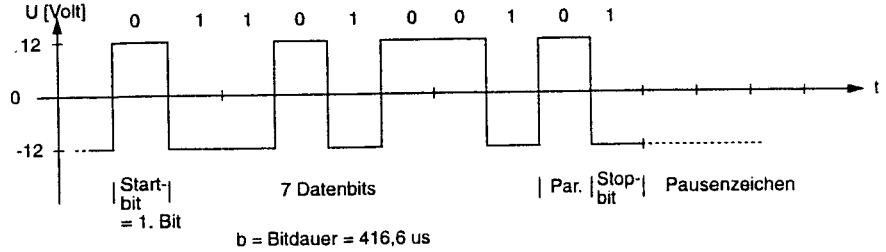


b)Die Assemblerdarstellung der Routine lautet z.B.:

DIV	LDA #8	Zähler mit der Adresse $0500 mit...
	STA $0500	...dem Wert 8 initialisieren
	LDB $0501	Dividend (mit acht Führenden Nullen) ...
	CLRA	... laden, Nullen in Register A
	CLR $0503	Quotient löschen
DIVD:	ASL $0503	Quotient linksschieben
	ASLB	Dividend linksschieben (lower byte)
	RCLA	Dividend linksschieben (higher byte)
	CMPA $0502	Dividend ≤ Divisor ? ...
	BLO NOSUB	... Wenn ja, gehe zu NOSUB
	SUBA $0502	Alter Dividend-Divisor=neuer Dividend
	INC $0503	Quotient inkrementieren
NOSUB:	DEC $0500	Zähler dekrementieren, Zähler=0? ...
	BNE DIVD	... wenn ja, gehe zu DIVD
	STA $0504	Rest speichern, Quotient ist bereits unter Adresse $503 gespeichert

Lösungsvorschlag Aufgabe H2:

a) CR = 0 0 1 1 1 0 1 0 = $3A, IR = 0 1 1 0 0 1 0 1 = $65
b) $4B = 0 1 0 0 1 0 1 1



c) SR = 1 0 0 1 1 0 1 1 = $ 9 B

Begründung
PE = 1: gerade Parität verletzt, da 9. Bit=Paritätsbit=L-Pegel (‘1’) statt H-Pegel (‘0’)
FE = 1: Framing Error, da 10. Bit=Stopbit=H-Pegel statt L-Pegel
OE = ?: Overrun, aus Aufgabenstellung nicht zu entnehmen, hier willkürlich als 0 angenommen
RDRF=1: Zeichen empfangen, aber noch nicht von der CPU gelesen
TDRE=1: vorgegeben
$\overline{DCD}$ = 0: vorgegeben
$\overline{DSR}$ = $\overline{DTR}$: (Nullmodem) -> $\overline{DSR}$ = $\overline{DTR}$ = 0
IF = 1: Interruptanforderung des Empfängers liegt vor

Lösungsvorschlag Aufgabe H3:

a) Die Zahl Z lautet als Dezimalzahl: $Z = 0{,}6953125 \cdot 10^3$.
Im angegebenen Format der einfach-genauen Gleitpunkzahl lautet sie:

0	0	0	0	1	0	1	0	0	1	0	1	0	1	1	0	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0

b) Das Produkt der beiden dargestellten Zahlen 13 und -11 lautet als Dezimaldarstellung: -143,
als gepackte ganze Dezimalzahl in hexadezimaler Darstellung: $8000000000000000143,
als angegebene einfach-genaue Gleitpunkzahl

0	0	0	1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

und als gepackte Gleitpunktdezimalzahl in hexadezimaler Darstellung:
$8002000143000000000000000.

Lösungsvorschlag Aufgabe H4:

Reihenfolge der Speicherzugriffe:

Speicherzugriff	hit	miss
LDA $80DE	x	-
STA $A1CC	x	-
LDA $9F3D	-	x
LDA $743D	-	x
STA $E4FA	x	-
LDA $9122	-	x
LDA $E4FA	x	-
STA $9122	x	-
LDA $02FA	x	-
LDA $AA22	-	x
STA $0022	-	x

Akumulatorbelegung nach dem letzten Speicherzugriff: **A = #22**

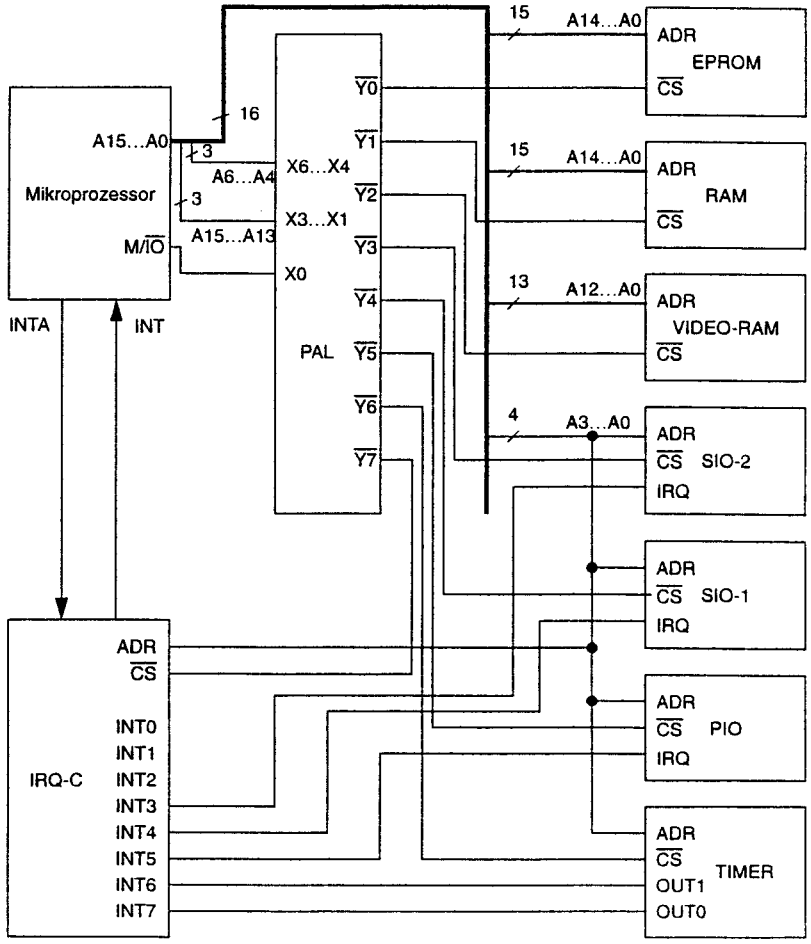
Ausschnitt aus der Cache-Belegung nach dem letzten Speicherzugriff (die Änderungen sind hervorgehoben):

Index	Way A		SB-Bit	Way B	
	Tag	Datum		Tag	Datum
FA	02	EB	->	E4	**C3**
DE	BB	86	<-	80	F7
CD	7C	22	←	92	9B
CC	A1	**F7**	->	3D	F7
7F	A3	DE	→	90	92
3D	**9F**	**74**	←	**74**	**C3**
2F	86	BB	←	AA	53
22	**AA**	**22**	→	**91**	**C3**

Änderungen im Ar. .sspeicher:

Adresse	Datum
E4FA	C3
A1CC	F7
9122	C3
0022	22
-	-

Lösungsvorschlag Aufgabe H5:



Für die Gleichungen des PLAs gilt:

$$\overline{Y0} = \overline{X0\ \overline{X3}}$$
$$\overline{Y1} = \overline{X0\ X3\ (\overline{X2} \vee \overline{X1})}$$
$$\overline{Y2} = \overline{X0\ X3\ X2\ X1}$$
$$\overline{Y3} = \overline{\overline{X0}\ X6\ \overline{X5}\ X4}$$
$$\overline{Y4} = \overline{\overline{X0}\ X6\ \overline{X5}\ \overline{X4}}$$
$$\overline{Y5} = \overline{\overline{X0}\ X6\ X5\ \overline{X4}}$$
$$\overline{Y6} = \overline{\overline{X0}\ \overline{X6}\ X5\ X4}$$
$$\overline{Y7} = \overline{\overline{X0}\ \overline{X6}\ X5\ \overline{X4}}$$

Lösungsvorschlag Aufgabe H6:

0	1	1	1	1	1	1	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0	1	1	0	1	1	1	0	1	0	1	0	1	0
HDLC-Flag								Zieladresse=9A								Rahmennummer=3								37=ASCII(7)							2A=ASCII(*)						

1	0	1	0	1	1	0	1	1	0	1	0	0	1	1	1	0	0	1	0	1	1	1	0	1	1	0	0	0	1	0	0	0	0	0
56=ASCII(V)							69=ASCII(i)							65=ASCII(e)							6C=ASCII(l)							20=ASCII(BLANK)						

1	0	0	0	1	0	1	1	1	1	0	0	1	0	1	1	0	0	1	1	0	1	1	0	1	1	1	1	1	***0***	1	0	1	1	0	0
45=ASCII(E)							72=ASCII(r)							66=ASCII(f)							6F=ASCII(o)							6C=ASCII(l)							

1	1	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	1	1	1	1	1	1	0
67=ASCII(g)							21=ASCII(!)							CRC-Prüfsumme																HDLC-Flag							

Das „l" aus „Erfolg" enthält eine zusätzliche 0 (bit stuffing).