

AUFGABE 1

a) Markieren Sie mit Pfeilen nur die Abhängigkeiten der Pipelinestufen, die im gegebenen Programm zu Pipeline-Konflikten führen.



```

S1: LOAD  R1, (A1)    ; R1:= <(A1)>
S2: LOAD  R2, (A2)    ; R2:= <(A2)>
S3: LOAD  R3, (A3)    ; R3:= <(A3)>
S4: ADD   R4, R3, R2   ; R4:= R3 + R2
S5: SUB   R3, R2, R1   ; R3:= R2 - R1
S6: ADD   R2, R3, R4   ; R2:= R3 + R4
S7: STORE (A3), R4    ; <(A3)>:= R4
  
```

S2 -> S4, S3 -> S4, S4 -> S6, S5 -> S6

b)

S1
 S2
 S3
NOOP
NOOP
 S4
 S5
NOOP
NOOP
 S6
 S7

c) $T_k = k + (n-1) = 5 + (11-1) = 15$ Takte

d) $T_{max} = \text{längster Befehl} + \text{reg. Zeit}$

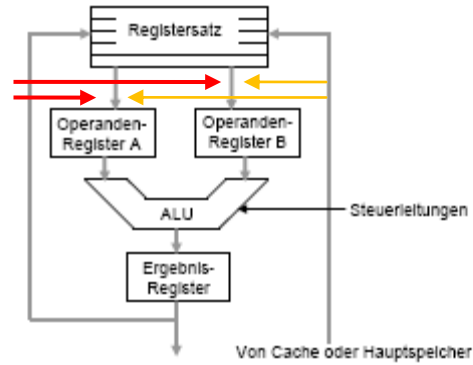
$$= 6\text{ns} + 3\text{ns} = 9\text{ns}$$

$$T_{\text{freq}} = 1 / T_{max} = 1/9\text{ns} = 111,111\text{MHZ}$$

f) $\text{Speedup} = \frac{n \cdot k}{k + (n-1)}$: $4,99 = \frac{n \cdot 5}{5 + (n-1)}$ $n = 1996$ Befehle

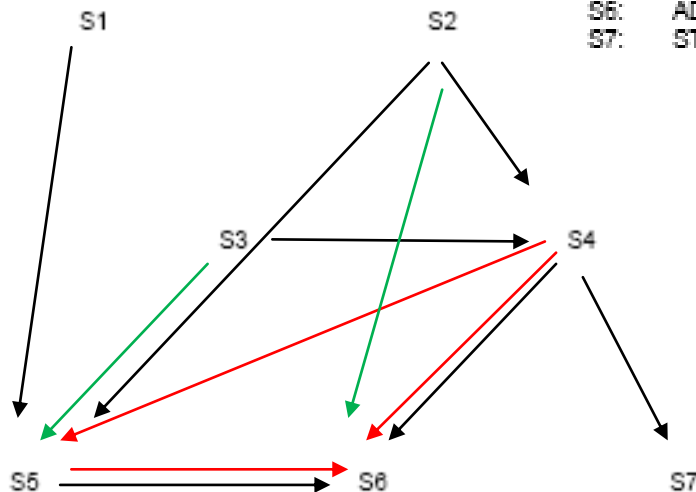
g)

g) Zeichnen Sie die Modifikation des Rechners, die notwendig ist, um die unter a) festgestellten Pipeline-Konflikte hardwaremäßig zu lösen in das folgende Bild ein.



i)

S1:	LOAD	R1, (A1)	; R1 := <(A1)>
S2:	LOAD	R2, (A2)	; R2 := <(A2)>
S3:	LOAD	R3, (A3)	; R3 := <(A3)>
S4:	ADD	R4, R3, R2	; R4 := R3 + R2
S5:	SUB	R3, R2, R1	; R3 := R2 - R1
S6:	ADD	R2, R3, R4	; R2 := R3 + R4
S7:	STORE	(A3), R4	; <(A3)> := R4



Da
Do
Dt

Seite 2.97 (script)

Reg-Reg	Reg-Speicher	Akkumulator	KELLER
Load R1,A Load R2,B mul R3,R1,R2 store C,R3	Load R1,A mul R1,B store C,R1	Load A mul B store C	Push A Push B mul pull C
Load R1,C load R2, A sub R3,R1,R2 Store D, R3	load R1,C sub R1,A store D,R1	load C sub A store D	push A push C sub pull D

	richtig	falsch
Ein sogenannter „look-through-cache“ wird bevorzugt bei Multi-Prozessorsystemen eingesetzt.		
Bei einem semi-synchronen Systembus kann die Anzahl der wait states des Prozessors variieren, abhängig vom adressierten Baustein.		
Beim Rückschreibverfahren (Copy Back) wird beim Schreiben des Prozessors bei einem Cache-Hit sowohl der Cache als auch der Hauptspeicher aktualisiert.		
Unter Data-Alignment versteht man die Verschachtelung von Hauptspeicherzugriffen auf verschiedene Speicherbänke.		
Bei SDRAM-Speicherbausteinen kann die Zykluszeit durch Interleaving verbessert werden.		
Beim sogenannten „Bus Snooping“ wertet der Prozessor zur Konsistenzsicherung die Speicheradressen anderer Busmaster beim Zugriff auf den Hauptspeicher aus.		
Zur Implementierung einer dynamischen Sprungvorhersage kann ein voll-assoziativer Cache verwendet werden.		
Die mittlere Zugriffszeit in einem Speichersystem, das aus einem Verbund von Cachespeicher und Arbeitsspeicher besteht, steigt mit der Abnahme der Hitrate.		
Bei der virtuellen Adressierung ist die Anzahl der Seiten und der Rahmen immer gleich groß.		
Der TAS-Befehl wird auf der Hardwareebene der Prozessorrealisierung als ein sogenannter Read-Modify-Write-Zyklus ausgeführt.		
Bei VLIW-Prozessoren übernimmt der Compiler die dynamische Befehlszuteilung auf die Ausführungseinheiten.		
Bei einem Mehrwege Satz-assoziativen Cachespeicher mit 2^n Zeilen sind 2^n Vergleiche erforderlich.		

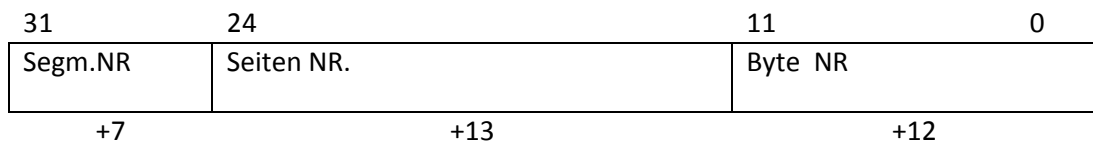
Aufgabe 4 a)

	31	5	3	0
DM	TAG	Index	Block-Nr.	
	+26	+2	+4	

	31	4	3	0
A2	TAG	Index	Block-Nr.	
	+27	+1	+4	

	31	4	
VA	TAG	Block-Nr.	
	+28	+4	

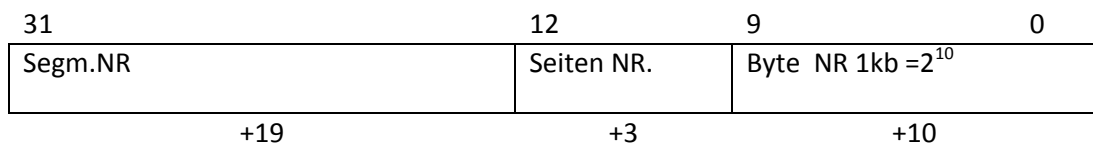
Aufgabe 5



- a) Max Segment gröÙe : $32 - 7 : 2^{25} = 32\text{MB}$
- b) Max. Seiten pro Segm: 2^{13} (mögliche adressierungen)
- c) Framegröße = 2^{12}
RAM = AnzahlRahmen * Framegröße = $2^{20} * 2^{12} = 2^{32} = 4\text{GB}$
- d) Veschnitt : Gesamt (2^{25}) – Belegt ($1024 \text{ Seiten} * 2^{12} \text{ seitengöÙe}$) = 28 MB
- e) $12 \text{ MB} = 12 * 2^{20} / 2^{12} = 12 * 2^8 = 3072 \text{ Frames}$

Aufgabe 5.2

f)



- g) $2200 \rightarrow 10 \quad 0010011000 \quad \rightarrow 2 = 3 \rightarrow 011 \quad 0010011000 = 3224$
 $6630 \rightarrow 110 \quad 0111100110 \quad \rightarrow 6 = 1 \rightarrow 001 \quad 0111100110 = 1510$
 $1021 \rightarrow 0 \quad 1111111101 \quad \rightarrow \text{Seiten Fehler (page fault)}$

Aufgabe 6

a)

K1	DC.W	\$FFFA	$\rightarrow K1 = \$FFFA \text{ (Wortgröße)}$	
K2	DS.L	\$1	$\rightarrow K2 = 1 \text{ Longwort Speicher}$	
START	MOVE.L	#K1, D0	$\rightarrow D0 = 00 \ 0F \ FF \ FA$	
	MOVE.W	K1, D1	$\rightarrow D1 = 22 \ 22 \ 01 \ 02$	
	ADD.W	D1, D2	$\rightarrow D2 = 44 \ 44 \ 56 \ 57$	$\begin{array}{r} 01 \ 02 \\ + 55 \ 55 \\ \hline 56 \ 57 \end{array}$
	MOVE.W	D2, K2		

Prozessor-Register	
Datenregister	Inhalt
D0	$\$00 \ 0F \ FF \ FA$
D2	$\$44 \ 44 \ 56 \ 57$

Tabelle 8.2 : Lösung zu Teilaufgabe a)

- c) Tragen Sie den Inhalt der Datenregister D3 und D4 (je 32 Bit) in Tabelle 8.4 ein, nachdem mit der Ausgangssituation entsprechend Tabelle 8.1 die folgenden Befehle nacheinander ausgeführt wurden :

SWAP D3 → D3 = 7777 6666
 SUB.L -(A1),D4 → -(A1) = 9-0006 → D4 =

88 88 99 99
 - 06 07 08 09

 82 81 91 90

Prozessor-Register	
Datenregister	Inhalt
D3	\$ 77 77 66 66
D4	\$ 82 81 91 90

Tabelle 8.4 : Lösung zu Teilaufgabe c)