

CT1 Übungsaufgaben

Unterprogramme und Stack

Aufgabe

Im folgenden Assembler-Programm werden die vier Unterprogramme procA, procB, procC und procD aufgerufen und einige Register mit PUSH gesichert und zurückgeladen. Zu Beginn des Programmes steht der Stackpointer (SP) auf 0x20000660.

Bestimmen Sie zu den angegebenen Zeitpunkten Z1. . .Z4 den Inhalt des Stacks. Schreiben Sie daneben die Bedeutung des Stack-Inhaltes, z.B. Inhalt R1 von procX oder ret adr von procY).

```

28                                endless
29 0x0800079C 490D                LDR    R1, =0x10203040
30 0x0800079E 4A0E                LDR    R2, =0x50607080
31 0x080007A0 F000 F804           BL     procA
32 0x080007A4 F000 F807           BL     procB
33 0x080007A8 E7F8                B      endless
34 0x080007AA 00 00              ALIGN

39                                procA
40 0x080007AC B406                PUSH   {R1, R2}
41                                ; <----- (Z1)
42 0x080007AE 490B                LDR    R1, =0xAABBCCDD
43 0x080007B0 4A0B                LDR    R2, =0xEEFF1020
44 0x080007B2 BC06                POP    {R1, R2}
45 0x080007B4 4770                BX     LR
46
47                                procB
48 0x080007B6 B506                PUSH   {R1, R2, LR}
49                                ; <----- (Z2)
50 0x080007B8 490A                LDR    R1, =0x11223344
51 0x080007BA 4A0B                LDR    R2, =0x55667788
52 0x080007BC F000 F801           BL     procC
53 0x080007C0 BD06                POP    {R1, R2, PC}
54
55                                procC
56 0x080007C2 B506                PUSH   {R1, R2, LR}
57                                ; <----- (Z3)
58 0x080007C4 4909                LDR    R1, =0x11111111
59 0x080007C4 4A0A                LDR    R2, =0x22222222
60 0x080007C8 F000 F801           BL     procD
61 0x080007CC BD06                POP    {R1, R2, PC}
62                                ; <----- (Z4)

```

	Z1		Z2		Z3		Z4	
0x2000'0660								
0x2000'065F	50		08		08		08	
0x2000'065E	60	Inhalt R2	00	Ret_addr	00	Ret_addr	00	Ret_addr
0x2000'065D	70	von endless	07	procB	07	procB	07	procB
0x2000'065C	80		A8		A8		A8	
0x2000'065B	10		50		50		50	
0x2000'065A	20	Inhalt R1	60	Inhalt R2	60	Inhalt R2	60	Inhalt R2
0x2000'0659	30	von endless	70	von endless	70	von endless	70	von endless
0x2000'0658	40		80		80		80	
0x2000'0657			10		10		10	
0x2000'0656			20	Inhalt R1	20	Inhalt R1	20	Inhalt R1
0x2000'0655			30	von endless	30	von endless	30	von endless
0x2000'0654			40		40		40	
0x2000'0653					08			
0x2000'0652					00	Ret_addr		
0x2000'0651					07	procC		
0x2000'0650					C0			
0x2000'064F					55			
0x2000'064E					66	Inhalt R2		
0x2000'064D					77	von procB		
0x2000'064C					88			
0x2000'064B					11			
0x2000'064A					22	Inhalt R1		
0x2000'0649					33	von procB		
0x2000'0648					44			
0x2000'0647								
0x2000'0646								
0x2000'0645								
0x2000'0644								