

Study Questions (Chapter 04 – Part 1)

1. The program in Tutorial 14 utilizes an FD stack. Modify the program to make it utilize an ED stack. Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
Type, assemble, and run your modified program to make sure that it works properly.
2. The program in Tutorial 14 utilizes an FD stack. Modify the program to make it utilize an FA stack. Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
Type, assemble, and run your modified program to make sure that it works properly.
3. The program in Tutorial 14 utilizes an FD stack. Modify the program to make it utilize an EA stack. Graphically show the content of the stack and the location of the SP after executing each instruction in your program.
Type, assemble, and run your modified program to make sure that it works properly.

ARM^Q的8种指令后缀:

ia (increase after) 先传输, 再地址+4
ib (increase before) 先地址+4, 再传输
da (decrease after) 先传输, 再地址-4
db (decrease before) 先地址-4, 再传输
fd (full decrease) 满递减 堆栈^Q
ed (empty decrease) 空递减堆栈
fa (full add) 满递增堆栈
ea (empty add) 空递增堆栈

例如:

stmia sp, {r0 - r12}

将r0存入sp指向的 内存^Q处 (假设为0x30001000); 然后地址+4 (即指向0x30001004), 将r1存入该地址; 然后地址再+4 (指向0x30001008), 将r2存入该地址……直到r12内容放入 (0x30001030), 指令完成。

4种栈结构:

空栈: 栈指针指向空位, 每次存入时可以直接存入然后栈指针移动一格; 而取出时需要先移动一格才能取出

满栈: 栈指针指向栈中最后一格数据, 每次存入时需要先移动栈指针一格再存入; 取出时可以直接取出, 然后再移动栈指针

增栈: 栈指针移动时向地址增加的方向移动的栈

减栈: 栈指针移动时向地址减小的方向移动的栈

ARM种的4种栈结构: $SVB\ sp, sp\#4$

$MOV\ r7, sp$ ↓

1. 满减栈: 进栈 (先移动指针再入栈, 指针往地址减小的方向移动); 出栈 (先出栈, 栈指针往地址增大的地方移动)。
 2. 满增栈: 进栈 (先移动指针再入栈, 指针往地址增大的方向移动); 出栈 (先出栈, 栈指针往地址减小的地方移动)。
 3. 空减栈: 进栈 (先进栈, 栈指针往地址减小的方向移动); 出栈 (先移动指针再出栈, 栈指针往地址增大的方向移动)
 4. 空增栈: 进栈 (先进栈, 栈指针往地址增大的方向移动); 出栈 (先移动指针再出栈, 栈指针往地址减小的方向移动)
- 备注: 使用时不用纠结时哪一种栈结构, 只要进栈和出栈用同一种栈结构就不会出错。

备注:

ARM默认使用满减栈

ED: STMFD sp!, {fp, lr}	FA ^{FA} STMFD sp!, {fp, lr}
MOV fp, sp	MOV fp, sp
SUB sp, sp, #4	SUB ADD sp, sp, #4
LDR r2, [fp, #8]	LDR r2, [fp, #8 ^{#-8}]
ADD r2, r2, #120	ADD r2, r2, #120
STR r2, [fp, #-4]	STR r2, [fp, #-4 ^{#4.}]
ADD sp, sp, #4	SUB ADD sp, sp, #4
LDMFD sp!, {fp, pc}	LDMFD sp!, {fp, pc}

ED STMED sp!, {fp, lr}	EA STMEA sp!, {fp, lr}
SUB sp, sp, #4.	ADD sp, sp, #4.
MOV fp, sp	MOV fp, sp
LDR r2, [fp, #8].	LDR r2, [fp, #8 ^{#-8}].
ADD r2, r2, #120	ADD r2, r2, #120
STR r2, [fp, #-4].	STR r2, [fp, #4].
ADD sp, sp, #4.	SUB sp, sp, #4.
LDMED sp!, {fp, pc}.	LDMED sp!, {fp, pc}.