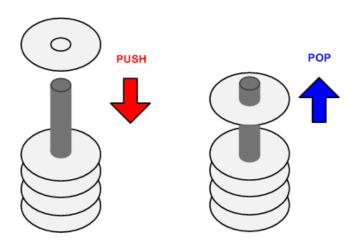
## Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C

# Chapter 8 Preserve Environment via Stack

Dr. Yifeng Zhu Electrical and Computer Engineering University of Maine

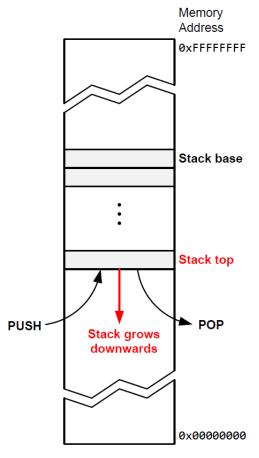
Spring 2015

- A Last-In-First-Out data structure
- Only allow to access the most recently added item
  - Also called the top of the stack
- Key operations:
  - push (add item to stack)
  - pop (remove top item from stack)

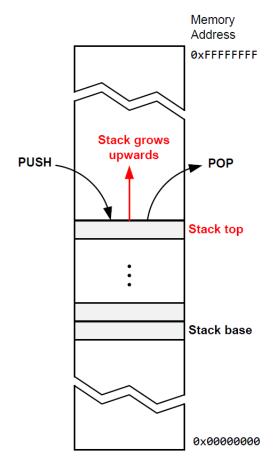




# Stack Growth Convention: Ascending *vs* Descending

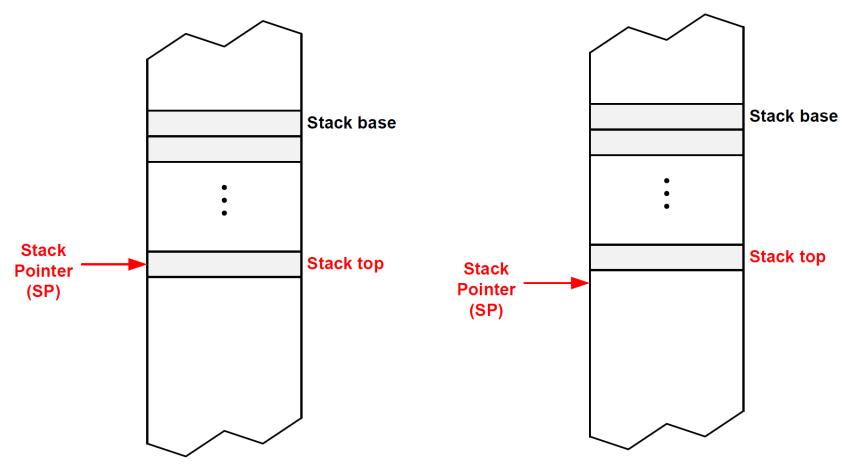


**Descending stack**: Stack grows towards low memory address



Ascending stack: Stack grows towards high memory address

# Stack Growth Convention: Full *vs* Empty

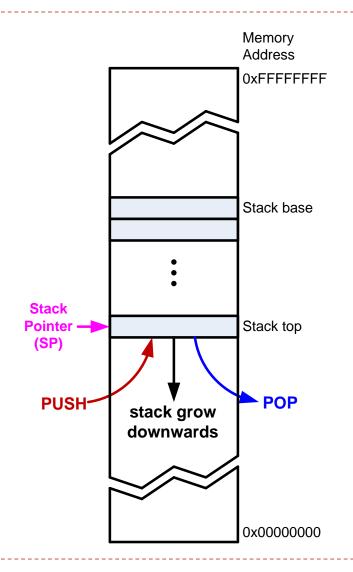


Full stack: SP points to the last item pushed onto the stack

**Empty stack**: SP points to the next free space on the stack

#### Cortex-M3 Stack

- stack pointer (SP) = R13
- Cortex-M3 uses full descending stack!
- stack pointer
  - decremented on PUSH
  - incremented on POP
  - SP starts at 0x20000200 for STM32-Discovery



## PUSH $\{Rd\}$

- ▶ (\*SP) = Rd  $\rightarrow$  full stack

#### Push multiple registers

```
They are equivalent.

PUSH {r8}

PUSH {r8, r7, r6}

PUSH {r7}

PUSH {r7}

PUSH {r6}
```

- The order in which registers listed in the register list does not matter.
- When pushing multiple registers, these registers are automatically sorted by name and the lowest-numbered register is stored to the lowest memory address.

## **POP** {*Rd*}

- $\triangleright$  SP = SP + 4  $\rightarrow$  Stack shrinks

#### Pop multiple registers

```
They are equivalent.

POP {r6, r7, r8} 

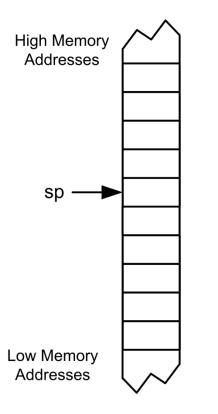
POP {r8, r7, r6} 

POP {r7}

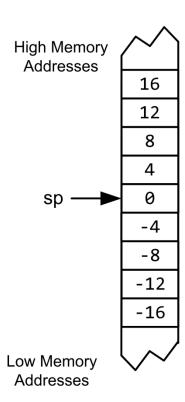
POP {r8}
```

- The order in which registers listed in the register list does not matter.
- When popping multiple registers, these registers are automatically sorted by name and the lowest-numbered register is loaded from the lowest memory address.

#### PUSH {r3, r1, r7, r2}



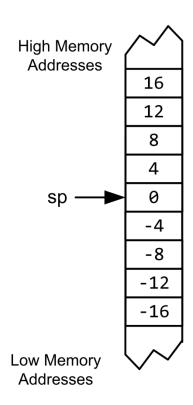
#### POP {r3, r1, r7, r2}



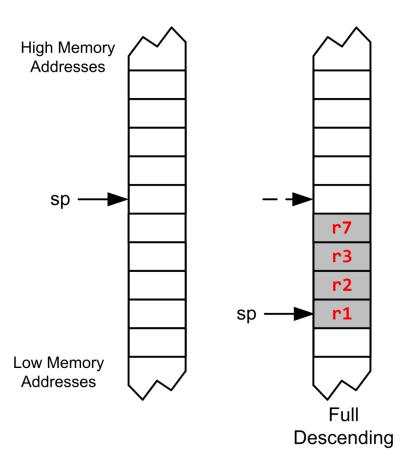
#### PUSH {r3, r1, r7, r2}

## High Memory Addresses sp r7 r3 r2 r1 sp · Low Memory Addresses Full Descending

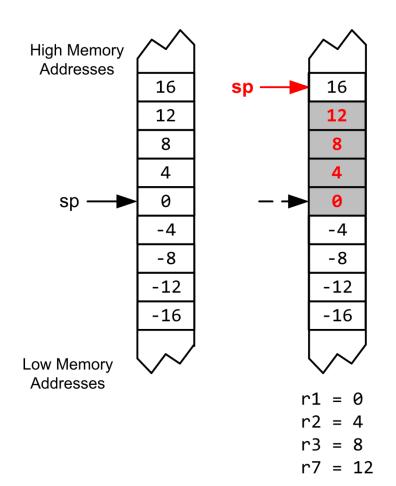
#### POP {r3, r1, r7, r2}

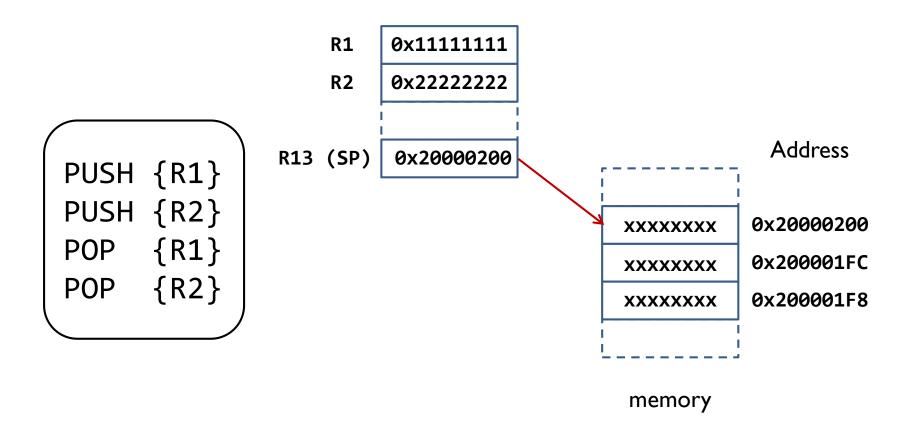


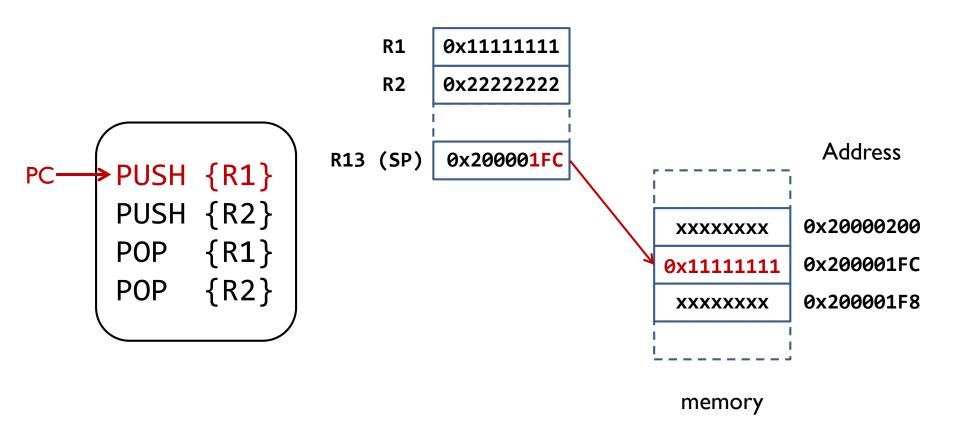
#### PUSH {r3, r1, r7, r2}

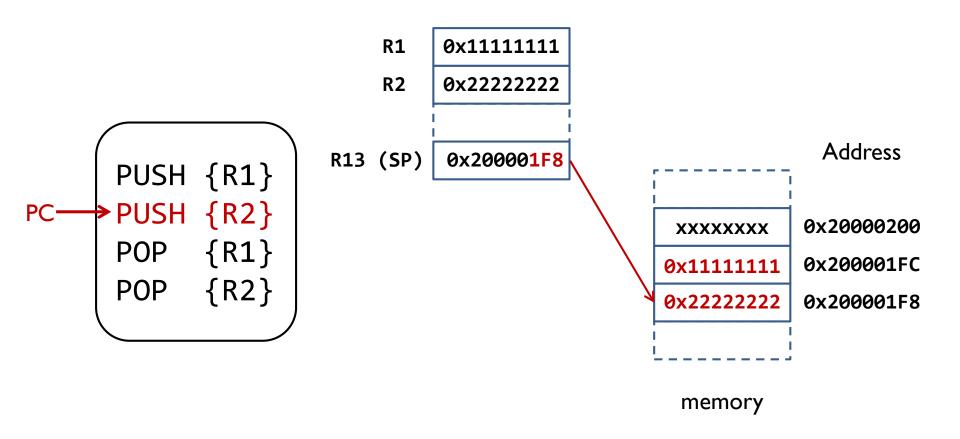


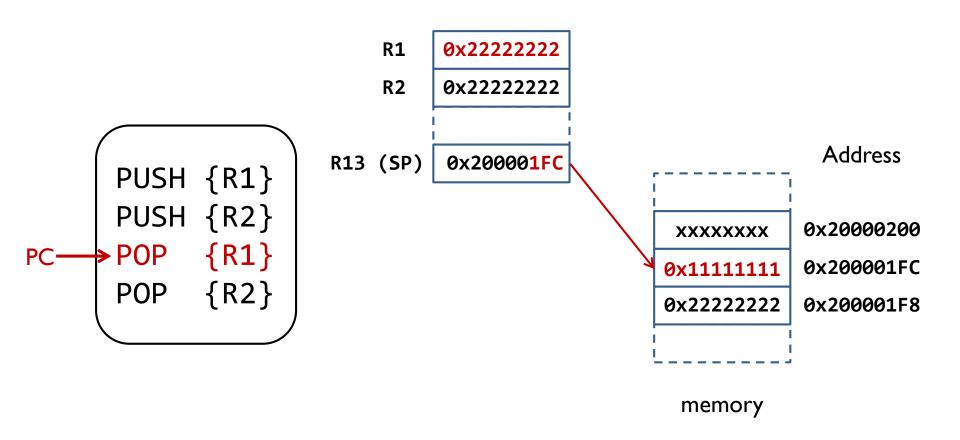
#### POP {r3, r1, r7, r2}

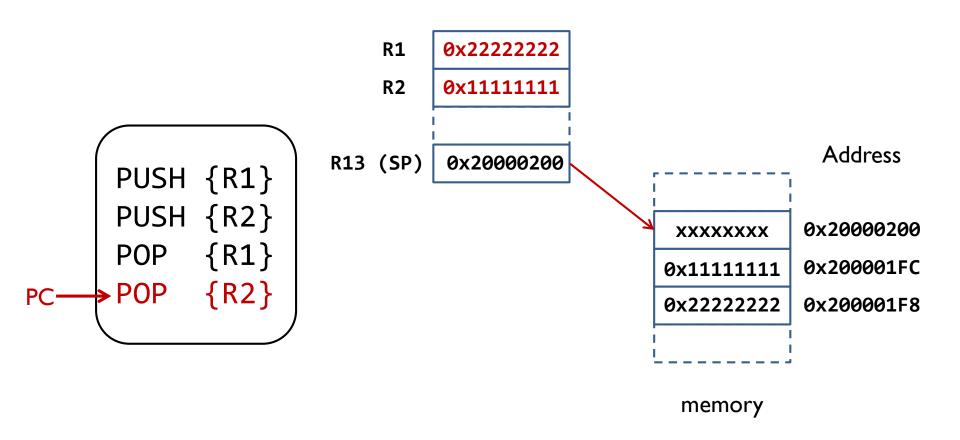








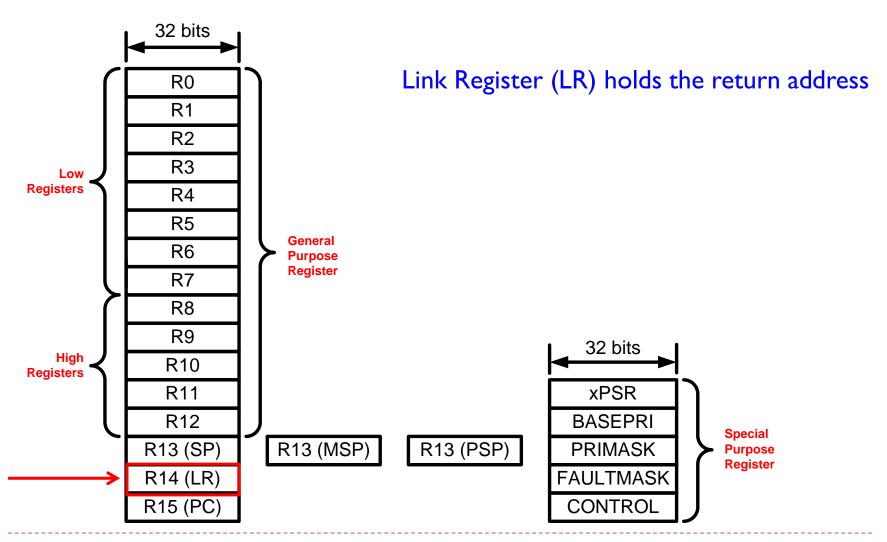




#### Subroutine

- A subroutines, also called a function or a procedure,
  - single-entry, single-exit
  - Return to caller after it exits
- When a subroutine is called, the Link Register (LR) holds the memory address of the next instruction to be executed when the subroutine exits.

## Link Register



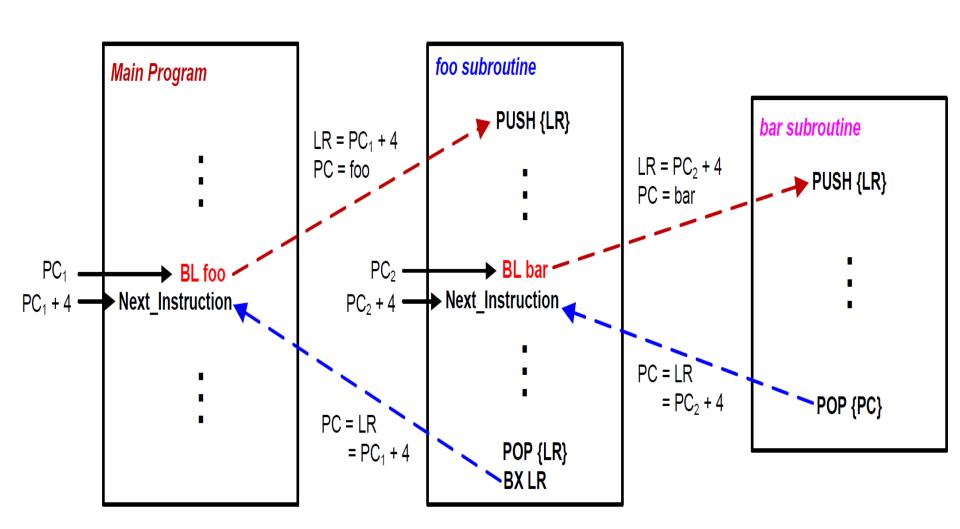
## Call a Subroutine

Caller Program	Subroutine/Callee
MOV r4, #100 BL foo ADD r4, r4, #1 ; r4 = 11, not 101	foo PROC MOV r4, #10 ; foo changes r4 BX LR ENDP

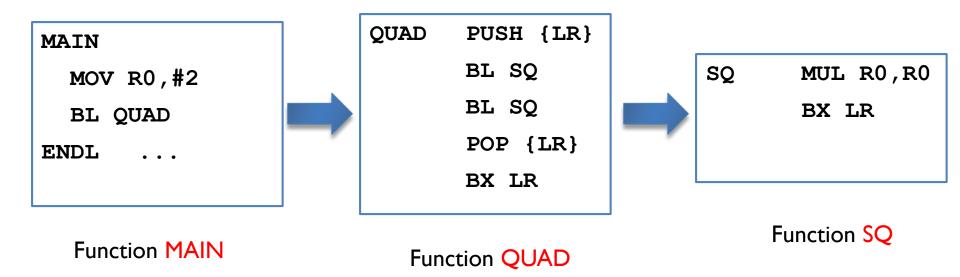
## Preserve Runtime Environment via Stack

Caller Program		Subrou	tine/Callee	
MOV r4, #100 BL foo ADD r4, r4, #1 ; r4	- = 101, not 11	foo PR PU MO PO BX	OC SH {r4} V r4, #10 P {r4}	<pre>; preserve r4 ; foo changes r4 ; Recover r4</pre>
		ENDP		

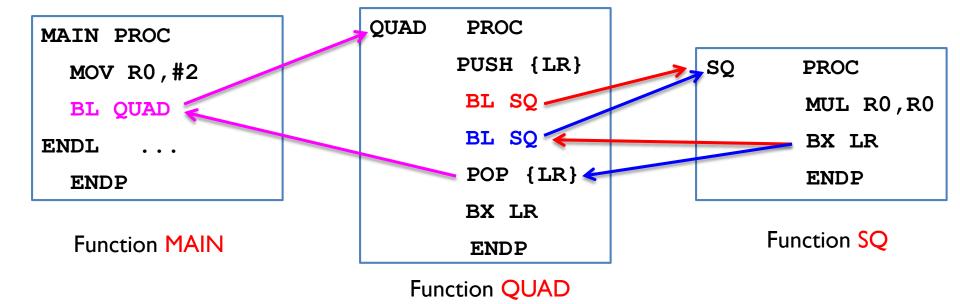
### Stacks and Subroutines



# Subroutine calling Another Subroutine



## Subroutine Calling Another Subroutine



	MOV R0,#2			xxxxxxx
	BL QUAD		/	
	B ENDL			
		R0		
SQ	MUL R0,R0		/ 1	MOV R0,#2
	BX LR		//	BL QUAD
	<b>2 2.</b> .	SP	0×20000200	B ENDL
		31	\sQ	MUL R0,R0
QUAD	PUSH {LR}	LR		
	BL SQ	PC	0×08000138	BX LR
	BL SQ	10	QUAL	PUSH {LR}
	POP {LR}			BL SQ
	BX LR			BL SQ
ENDL				POP {LR}
				BX LR

0×20000200

0x200001FC

0×200001F8

0x08000138

0x0800013F

0x08000144

0x08000148

0x0800014F

0x08000154

0x08000158

0×0800015F

0x08000164

0x08000168

	MOV R0,#2	xxxxxxxx	0×20000200
	BL QUAD		0×200001F0
	B ENDL		0×200001F8
		R0 0x02	_
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0×0800013F
		SP 0×20000200 B ENDL	0x08000144
0	D	KO MULRORO	0×08000148
QUAD	PUSH {LR}		-
	BL SQ	PC 0x0800013F BX LR	0×0800014F
	BL SQ	QUAD PUSH {LR}	0x08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL	• • •	POP {LR}	0x08000164
		BX LR	0×08000168
1			

			_
	MOV R0,#2	xxxxxxx	0×20000200
	BL QUAD		0x200001FC
	B ENDL		0×200001F8
		R0 0x02	
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0×0800013F
		SP 0×20000200 B ENDL	0×08000144
OHAD	DUGU (TD)	SO MILL RORO	0×08000148
QUAD	PUSH {LR}	LR   0x08000144 /	-
	BL SQ	PC 0x08000154 BX LR	0×0800014F
	BL SQ	QUAD PUSH {LR}	0×08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL		Preserve POP {LR}	0×08000164
		Link Register (LR)  BX LR	0×08000168
			_

	MOV R0,#2	xxxxxx	xx 0x20000200
	BL QUAD	0×080001	44 0×200001FC
	B ENDL		0×200001F8
		R0 0×02	
SQ	MUL R0,R0	MOV R0,	#2 0×08000138
	BX LR	BL QUAI	D 0x0800013F
	<b>2 2.</b> .	SP 0×200001FC B ENDL	0×08000144
QUAD	PUSH {LR}	SO MULRO	R0 0×08000148
QOILD	BL SQ	LR 0x08000144 BX LR	0×0800014F
	ъп об	PC 0x08000158	000000154
	BL SQ	QUAD PUSH {LI	R} 0×08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL		POP {LR]	) 0×08000164
		BX LR	0×08000168

MOV R0,#2	xxxxxxxx	0×20000200
BL QUAD	0×08000144	0×200001FC
B ENDL		0×200001F8
	R0 0x02	
MUL RO.RO	MOV R0,#2	0×08000138
,	BL QUAD	0×0800013F
	SP 0×20000 FC B ENDL	0×08000144
PUSH {LR}	SOLMUL RO.RO	0×08000148
	BX I R	0×0800014F
~	QVAD PUSH {LR}	0×08000154
~	BL SQ	0×08000158
	BL SQ	0×0800015F
	POP {LR}	0×08000164
<del>-</del>	DYLD	0×08000168
	BL QUAD	BL QUAD B ENDL  RO 0x02  MUL RO, RO BX LR  SP 0x200001FC PUSH {LR} BL SQ BL SQ PC 0x08000148  PUSH {LR} BL SQ POP {LR} BL SQ POP {LR}  BL SQ POP {LR}  POP {LR}  POP {LR}

	MOV R0,#2	xxxxxxxx	0×20000200
	BL QUAD	0×08000144	0×200001FC
	B ENDL	R0 0×04	0x200001F8
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0×0800013F
		SP 0x200001FC B ENDL	0×08000144
QUAD	PUSH {LR}	LR 0x0800015F SQ MUL R0,R0	0×08000148
<b>2</b>	BL SQ	PC 0x0800014F BX LR	0×0800014F
	BL SQ	QUAD PUSH {LR}	0×08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL		POP {LR}	0×08000164
		BX LR	0×08000168
			1

	MOV R0,#2	xxxxxxx	0×20000200
	BL QUAD	0×08000144	0×200001FC
	B ENDL		0×200001F8
		R0 0×04	
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0×0800013F
		SP 0×200001FC B ENDL	0×08000144
OTTAD	DIIGII (ID)	SO MUL RORO	0×08000148
QUAD	PUSH {LR}	LR 0x0800015F BX LR	0×0800014F
	BL SQ	PC   0x0800015Fk \	UXU0UUU141
	BL SQ	QUAD PUSH {LR}	0×08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL		POP {LR}	0×08000164
		BX LR	0×08000168

	MOV R0,#2	xxxxxxx	0×20000200
	BL QUAD	0×0800014	4 0x200001FC
	B ENDL		0x200001F8
		R0 0x04	
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0x0800013F
		SP 0×200001FC B ENDL	0x08000144
QUAD	PUSH {LR}	LR 0x08000164 SQ MUL R0,R0	0×08000148
2000	BL SQ	PC 0x08000148 BX LR	0×0800014F
	BL SQ	QUAD PUSH {LR}	0x08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL		POP {LR}	0×08000164
		BX LR	0×08000168

	MOV R0,#2	xxxxxxx	0×20000200
	BL QUAD	0×08000144	0x200001FC
	B ENDL		0×200001F8
		R0 0×10	
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0×0800013F
		SP 0×200001FC B ENDL	0×08000144
QUAD	PUSH {LR}	SO MUL RORO	0×08000148
QUILD	BL SQ	LR 0x08000164 BX LR	0×0800014F
	ъп об	PC 0x0800014F	0 00000154
	BL SQ	QUAD PUSH {LR}	0×08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL		POP {LR}	0×08000164
		BX LR	0×08000168

	MOV R0,#2	xxxxxxx	0×20000200
	BL QUAD	0×08000144	0×200001FC
	B ENDL		0×200001F8
		R0 0×10	
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0×0800013F
	<b>2 2.</b> .	SP 0×200001FC B ENDL	0×08000144
QUAD	PUSH {LR}	SO MULRORO	0×08000148
QUID	BL SQ	LR 0x08000164 BX LR	0×0800014F
	~	PC 0x08000164 QUAD PUSH {LR}	0×08000154
	BL SQ	POSH (LK)	000000134
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL	• • •	POP {LR}	0×08000164
		BX LR	0×08000168

	MOV R0,#2	xxxxxxxx	0×20000200
	BL QUAD		0×200001FC
	B ENDL		0×200001F8
		R0 0x10	
SQ	MUL R0,R0	MOV R0,#2	0×08000138
	BX LR	BL QUAD	0×0800013F
		SP 0×20000200 B ENDL	0×08000144
	DUCU (ID)	SO MULRORO	0×08000148
QUAD	PUSH {LR}	LR 0×08000144 BX LR	0×0800014F
	BL SQ	PC   0x08000168	UXUOUUUITI
	BL SQ	QUAD PUSH {LR}	0×08000154
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL	• • •	Recover POP {LR}	0×08000164
		Link Register (LR)  BX LR	0×08000168
			_

	MOV R0,#2	xxxxxx	xx 0x20000200
	BL QUAD	0×08000	0×200001FC
	B ENDL		0×200001F8
		R0 0x10	
SQ	MUL R0,R0	MOV R0	,#2 0×08000138
	BX LR	BL QUA	D 0x0800013F
		SP 0×20000200 B ENDL	0×08000144
	PUSH {LR}	/SO MIII RO	R0 0×08000148
QOMD	BL SQ	LR 0x08000144 BX LR	0×0800014F
	~	PC 0x08000144 QUAD PUSH {L	R} 0×08000154
	BL SQ	QUAD FUSH (L	UXUOUUU 134
	POP {LR}	BL SQ	0×08000158
	BX LR	BL SQ	0×0800015F
ENDL		POP {LR	) 0x08000164
		BX LR	0×08000168
			<del></del>