		ADDR	DATA
		\$FF9C	
		\$FFA0	
		\$FFA4	
		\$FFA8	
SP	\$FFF4	\$FFAC	
		\$FFB0	
		\$FFB4	
FP	\$FFF8	\$FFB8	
		\$FFBC	
		\$FFC0	
PC	\$1100	\$FFC4	
		\$FFC8	
		\$FFCC	
LR	\$400	\$FFD0	
		\$FFD4	
		\$FFD8	
		\$FFDC	
		\$FFE0	
		\$FFE4	
	Frame	\$FFE8	
	attivo	\$FFEC	
	attivo	\$FFF0	
		\$FFF4	
		\$FFF8	
		•	

E' stato appena eseguito il salto alla funzione MAIN con una chiamata a BL. Il valore del PC è stato salvato nel registro LR (R14) e il PC è stato aggiornato. Sono dentro MAIN

\$1100	STMFD	SP!, {FP,LR}	int main()
	MOV	FP, SP	{
	SUB	SP, SP, #16	int W, X, Y, Z;
\$110C			R(X, Y, &Z);
			}
\$1140	LDR	R2, [FP, #-8]	
	LDR	R1, [FP, #-12]	
	SUB	R0, FP, #16	
	STMFD	SP!, {R0-R2}	
\$1150	BL	R	
\$1154	ADD	SP, SP, #12	
\$1158			
	MOV	SP, FP	
	LDMFD	SP!, {FP, PC}	

		ADDR	DATA
		\$FF9C	
		\$FFA0	
		\$FFA4	
		\$FFA8	
SP	\$FFEC	\$FFAC	
		\$FFB0	
		\$FFB4	
FP	\$FFEC	\$FFB8	
		\$FFBC	
		\$FFC0	
PC	\$1108	\$FFC4	
		\$FFC8	
		\$FFCC	
LR	\$400	\$FFD0	
		\$FFD4	
		\$FFD8	
		\$FFDC	
		\$FFE0	
		\$FFE4	
		\$FFE8	
		\$FFEC	\$FFF8
		\$FFF0	\$400
		\$FFF4	
		\$FFF8	
		•	

Salvo il valore di LR nello stack, assieme al valore di FP. Poi aggiorno il valore di FP: ho creato un nuovo frame

\$1100	STMFD	SP!, {FP,LR}	int main()
	MOV	FP, SP	{
\$110C	SUB	SP, SP, #16	int W, X, Y, Z; R(X, Y, &Z);
\$1140 \$1150 \$1154 \$1158	LDR LDR SUB STMFD BL ADD	R2, [FP, #-8] R1, [FP, #-12] R0, FP, #16 SP!, {R0-R2} R SP, SP, #12	•
	MOV LDMFD	SP, FP SP!, {FP, PC}	

		ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8	DATA	
SP	\$FFDC	\$FFAC \$FFB0 \$FFB4 \$FFB8		
FP	\$FFEC	\$FFBC \$FFC0 \$FFC4		
PC	\$110C	\$FFC8 \$FFCC \$FFD0		
LR	\$400	\$FFD4 \$FFD8		7
		\$FFDC \$FFE0 \$FFE4 \$FFE8 \$FFEC \$FFF0	0 0 0 0 \$FFF8 \$400	Z Y X W
		\$FFF4 \$FFF8	Ψ-100	

Ho allocato lo spazio per le variabili locali. Terminata la creazione del frame

\$1100	STMFD MOV	SP!, {FP,LR} FP, SP	int main() {
	SUB	SP, SP, #16	int W, X, Y, Z;
\$110C			R(X, Y, &Z); }
\$1140	LDR LDR SUB STMFD	R2, [FP, #-8] R1, [FP, #-12] R0, FP, #16 SP!, {R0-R2}	
\$1150	BL	R	
\$1154 \$1158	ADD	SP, SP, #12	
	MOV LDMFD	SP, FP SP!, {FP, PC}	

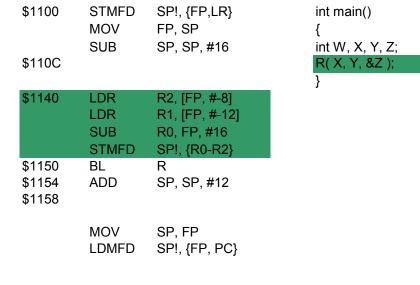
		ADDR	DATA	
		\$FF9C		
		\$FFA0		
		\$FFA4		
		\$FFA8		
		\$FFAC		
SP	\$FFDC	\$FFB0		
		\$FFB4		
		\$FFB8		
FP	\$FFEC	\$FFBC		
		\$FFC0		
		\$FFC4		
PC	\$1140	\$FFC8		
		\$FFCC		
		\$FFD0		
LR	\$400	\$FFD4		
		\$FFD8		
		\$FFDC	3	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

Vengono eseguite le istruzioni contenute in main (non esplicitate) Le variabili locali assumono dei valori calcolati da queste istruzioni

\$1100 \$110C	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #16	int main() { int W, X, Y, Z R(X, Y, &Z)
			}
\$1140 \$1150 \$1154 \$1158	LDR LDR SUB STMFD BL ADD	R2, [FP, #-8] R1, [FP, #-12] R0, FP, #16 SP!, {R0-R2} R SP, SP, #12	
	MOV LDMFD	SP, FP SP!, {FP, PC}	

		ADDR \$FF9C	DATA	I	Mi prepa
		\$FFA0			Nel fran
		\$FFA4			
		\$FFA8			
		\$FFAC			
SP	\$FFD0	\$FFB0			
		\$FFB4			
		\$FFB8			
FP	\$FFEC	\$FFBC			
		\$FFC0			
		\$FFC4			
PC	\$1150	\$FFC8			
		\$FFCC			
		\$FFD0	\$FFDC	&Z	
LR	\$400	\$FFD4	27	Υ	
		\$FFD8	111	X	
		\$FFDC	3	Z	
		\$FFE0	27	Υ	
		\$FFE4	111	X	
		\$FFE8	12	W	
		\$FFEC	\$FFF8		
		\$FFF0	\$400		
		\$FFF4			
		\$FFF8			

Mi preparo alla chiamata alla subroutine R Nel frame vengono preparati i parametri per la funzione R



		ADDR	DATA	
		\$FF9C		
		\$FFA0		
		\$FFA4		
		\$FFA8		
		\$FFAC		
SP	\$FFD0	\$FFB0		
		\$FFB4		
		\$FFB8		
FP	\$FFEC	\$FFBC		
		\$FFC0		
		\$FFC4		
PC	\$1200	\$FFC8		
		\$FFCC		
		\$FFD0	\$FFDC	&Z
LR	\$1154	\$FFD4	27	Υ
		\$FFD8	111	Х
		\$FFDC	3	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		
				-

E' stato appena eseguito il salto (chiamata a BL)
Il valore del PC è stato salvato nel registro LR e il suo valore aggiornato.
Il PC è stato aggiornato: sono dentro R

```
$1200
         STMFD
                   SP!, {FP,LR}
                                          void R(int I, int J, int *O)
         MOV
                   FP, SP
         SUB
                   SP, SP, #12
                                          int A, B, C;
$120C
                                          *O = I+J;
         LDR
                   R0, [FP, #16]
                                          if(A == 0)
         LDR
                   R1, [FP, #12]
                                          R(A, B, &C);
         ADD
                   R0, R0, R1
         LDR
                   R1, [FP, #8]
         STR
                   R0, [R1]
$1238
         LDR
                   R0, [FP, #-4]
         CMP
                   R0, #0
$1248
         BNE
                   $1268
$124C
         LDR
                   R2, [FP, #-4]
         LDR
                   R1, [FP, #-8]
         SUB
                   R0, FP, #12
         STMFD
                   SP!, {R0-R2}
$125C
         BL
                   R
$1260
         ADD
                   SP, SP, #12
$1264
$1268
         MOV
                   SP, FP
                   SP!, {FP, PC}
         LDMFD
```

		ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8	DATA	
SP	\$FFC8	\$FFAC \$FFB0 \$FFB4		
FP	\$FFC8	\$FFB8 \$FFBC \$FFC0 \$FFC4		
PC	\$1208	\$FFC8 \$FFCC \$FFD0	\$FFEC \$1154 \$FFDC	FP' LR(PC) &Z
LR	\$1154	\$FFD4 \$FFD8 \$FFDC \$FFE0 \$FFE4 \$FFE8	27 111 3 27 111 12	Y X Z Y X
		\$FFEC \$FFF0 \$FFF4 \$FFF8	\$FFF8 \$400	•••

Salvo il valore di LR nello stack, assieme al valore di FP. Poi aggiorno il valore di FP: ho creato un nuovo frame.

\$1200	STMFD	SP!, {FP,LR}
	MOV	FP, SP
	SUB	SP, SP, #12
\$120C		
	LDR	R0, [FP, #16]
	LDR	R1, [FP, #12]
	ADD	R0, R0, R1
	LDR	R1, [FP, #8]
	STR	R0, [R1]
\$1238		
*	LDR	R0, [FP, #-4]
	CMP	R0, #0
# 4040	_	
\$1248	BNE	\$1268
\$124C	LDR	R2, [FP, #-4]
Ψ.Ξ.Θ	LDR	R1, [FP, #-8]
	SUB	
		R0, FP, #12
	STMFD	SP!, {R0-R2}
\$125C	BL	R
\$1260	ADD	SP, SP, #12
\$1264		
\$1268	MOV	SP, FP
+ .=00	LDMFD	SP!, {FP, PC}
		Oi :, \i i ' , F O j

```
void R( int I, int J, int *O )
{
int A, B, C;
*O = I+J;
if( A == 0 )
R( A, B, &C );
}
```

		ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8	DATA	
SP	\$FFBC	\$FFAC \$FFB0		
		\$FFB4 \$FFB8		
FP	\$FFC8	\$FFBC	0	С
		\$FFC0	0	В
		\$FFC4	0	Α
PC	\$120C	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1154	\$FFD4	27	Υ
		\$FFD8	111	Χ
		\$FFDC	3	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

Ho allocato lo spazio per le variabili locali Terminata la creazione del frame

\$1200	STMFD	SP!, {FP,LR}	void R(int I, int J, int *O)
	MOV	FP, SP	{
	SUB	SP, SP, #12	int A, B, C;
\$120C			*O = I+J;
	LDR	R0, [FP, #16]	if(A == 0)
	LDR	R1, [FP, #12]	R(A, B, &C);
	ADD	R0, R0, R1	}
	LDR	R1, [FP, #8]	
	STR	R0, [R1]	
\$1238			
	LDR	R0, [FP, #-4]	
	CMP	R0, #0	
\$1248	BNE	\$1268	
\$124C	LDR	R2, [FP, #-4]	
•	LDR	R1, [FP, #-8]	
	SUB	R0, FP, #12	
	STMFD	SP!, {R0-R2}	
\$125C	BL	R	
\$1260	ADD	SP, SP, #12	
\$1264			
\$1268	MOV	SP, FP	
	LDMFD	SP!, {FP, PC}	

SP	\$FFBC	ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8 \$FFAC \$FFB0 \$FFB4 \$FFB8	DATA		
FP	\$FFC8	\$FFBC	44	С	
		\$FFC0	7	В	
		\$FFC4	0	Α	
PC	\$1238	\$FFC8	\$FFEC	FP'	
		\$FFCC	\$1154	LR(PC)	
		\$FFD0	\$FFDC	&Z	*0
LR	\$1154	\$FFD4	27	Υ	J
		\$FFD8	111	Χ	I
		\$FFDC	138	Z	
		\$FFE0	27	Υ	
		\$FFE4	111	X	
		\$FFE8	12	W	
		\$FFEC	\$FFF8		
		\$FFF0	\$400	ĺ	
		\$FFF4		ĺ	
		\$FFF8		ĺ	

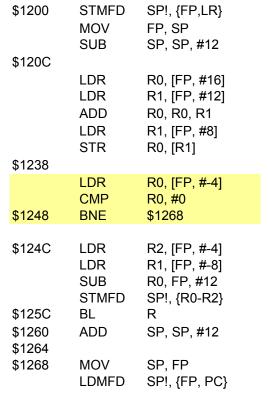
Eseguo le istruzioni in R (non esplicitate quelle che assegnano i valori ad A, B e C) Viene assegnato al parametro di uscita O il valore I+J. Dato che O è stato passato per indirizzo, di fatto viene modificato direttamente il valore della variabile Z (variabile locale di MAIN, il cui indirizzo è in FP+8)

\$1200 \$120C	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #12
	LDR LDR ADD LDR STR	R0, [FP, #16] R1, [FP, #12] R0, R0, R1 R1, [FP, #8] R0, [R1]
\$1238 \$1248	LDR CMP BNE	R0, [FP, #-4] R0, #0 \$1268
\$124C \$125C	LDR LDR SUB STMFD BL	R2, [FP, #-4] R1, [FP, #-8] R0, FP, #12 SP!, {R0-R2} R
\$1260 \$1264 \$1268	ADD MOV LDMFD	SP, SP, #12 SP, FP SP!, {FP, PC}

```
void R( int I, int J, int *O )
{
int A, B, C;
*O = I+J;
if( A == 0 )
R( A, B, &C );
}
```

SP	\$FFBC	ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8 \$FFAC \$FFB0 \$FFB4 \$FFB8	DATA	
FP	\$FFC8	\$FFBC \$FFC0	44 7	C B
PC	\$1248	\$FFC4 \$FFC8 \$FFCC	0 \$FFEC \$1154	A FP' LR(PC)
LR	\$1154	\$FFD0 \$FFD4	\$FFDC 27	&Z Y
		\$FFD8 \$FFDC \$FFE0	111 138 27	X Z Y
		\$FFE4 \$FFE8 \$FFEC	111 12 \$FFF8	X W
		\$FFF0 \$FFF4 \$FFF8	\$400	

Eseguo le istruzioni in R La condizione A==0 è vera: viene eseguita la chiamata ricorsiva alla subroutine R



void R(int I, int J, int *O)

int A, B, C;

if(A == 0)

R(A, B, &C);

*O = I+J;

		ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8 \$FFAC	DATA	
SP	\$FFB0	\$FFB0	\$FFBC	&C
		\$FFB4	7	В
		\$FFB8	0	Α
FP	\$FFC8	\$FFBC	44	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$125C	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1154	\$FFD4	27	Υ
		\$FFD8	111	Χ
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	X
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

Mi preparo alla chiamata ricorsiva Inserisco nello stack i valori dei parametri di ingresso della subroutine da chiamare

```
$1200
         STMFD
                   SP!, {FP,LR}
         MOV
                   FP, SP
         SUB
                   SP, SP, #12
$120C
         LDR
                   R0, [FP, #16]
         LDR
                   R1, [FP, #12]
         ADD
                   R0, R0, R1
         LDR
                   R1, [FP, #8]
         STR
                   R0, [R1]
$1238
         LDR
                   R0, [FP, #-4]
         CMP
                   R0, #0
$1248
         BNE
                   $1268
$124C
         LDR
                   R2, [FP, #-4]
         LDR
                   R1, [FP, #-8]
         SUB
                   R0, FP, #12
                  SP!, {R0-R2}
         STMFD
         BL
$125C
                   R
$1260
         ADD
                   SP, SP, #12
$1264
$1268
         MOV
                   SP, FP
                  SP!, {FP, PC}
         LDMFD
```

void R(int I, int J, int *O)

int A, B, C;

if(A == 0)

R(A, B, &C);

*O = I+J;

		ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8 \$FFAC	DATA	
SP	\$FFB0	\$FFB0	\$FFBC	&C
		\$FFB4	7	В
		\$FFB8	0	Α
FP	\$FFC8	\$FFBC	44	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$1200	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Y
		\$FFD8	111	Х
		\$FFDC	138	Z
		\$FFE0	27	Y
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

E' stato appena eseguito il salto (chiamata a BL)
Il valore del PC è stato salvato nel registro LR e il suo valore aggiornato.
Il PC è stato aggiornato: sono dentro R la seconda volta (ricorsione)

```
STMFD
                  SP!, {FP,LR}
$1200
                   FP, SP
         MOV
         SUB
                   SP, SP, #12
$120C
         LDR
                  R0, [FP, #16]
         LDR
                   R1, [FP, #12]
         ADD
                  R0, R0, R1
         LDR
                  R1, [FP, #8]
         STR
                   R0, [R1]
$1238
         LDR
                   R0, [FP, #-4]
         CMP
                   R0, #0
$1248
         BNE
                   $1268
$124C
         LDR
                   R2, [FP, #-4]
         LDR
                   R1, [FP, #-8]
         SUB
                  R0, FP, #12
         STMFD
                  SP!, {R0-R2}
$125C
         BL
                   R
$1260
         ADD
                   SP, SP, #12
$1264
$1268
         MOV
                   SP, FP
                  SP!, {FP, PC}
         LDMFD
```

void R(int I, int J, int *O)

int A, B, C;

if(A == 0)

R(A, B, &C);

*O = I+J;

		ADDR	DATA	
		\$FF9C		
		\$FFA0		
		\$FFA4		
		\$FFA8	\$FFC8	FP"
		\$FFAC	\$1260	LR(PC)
SP	\$FFA8	\$FFB0	\$FFBC	&C
		\$FFB4	7	В
		\$FFB8	0	Α
FP	\$FFA8	\$FFBC	44	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$1208	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	Χ
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	Χ
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		

\$FFF8

Salvo il valore di LR nello stack, assieme al valore di FP. Poi aggiorno il valore di FP: ho creato un nuovo frame

\$1200	STMFD	SP!, {FP,LR}
\$120C	MOV SUB	FP, SP SP, SP, #12
V1200	LDR LDR ADD LDR STR	R0, [FP, #16] R1, [FP, #12] R0, R0, R1 R1, [FP, #8] R0, [R1]
\$1238	LDD	DO [ED # 4]
* 40.40	LDR CMP	R0, [FP, #-4] R0, #0
\$1248	BNE	\$1268
\$124C	LDR LDR SUB STMFD	R2, [FP, #-4] R1, [FP, #-8] R0, FP, #12 SP!, {R0-R2}
\$125C	BL	R
\$1260 \$1264	ADD	SP, SP, #12
\$1268	MOV LDMFD	SP, FP SP!, {FP, PC}

```
void R( int I, int J, int *O )
{
int A, B, C;
*O = I+J;
if( A == 0 )
R( A, B, &C );
}
```

		ADDR	DATA	
		\$FF9C	0	С
		\$FFA0	0	В
		\$FFA4	0	Α
		\$FFA8	\$FFC8	FP"
		\$FFAC	\$1260	LR(PC)
SP	\$FF9C	\$FFB0	\$FFBC	&C
		\$FFB4	7	В
		\$FFB8	0	Α
FP	\$FFA8	\$FFBC	44	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$120C	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	X
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	X
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

Ho allocato lo spazio per le tre variabili locali

```
$1200
         STMFD
                   SP!, {FP,LR}
                   FP, SP
         MOV
         SUB
                   SP, SP, #12
$120C
         LDR
                   R0, [FP, #16]
         LDR
                   R1, [FP, #12]
         ADD
                   R0, R0, R1
         LDR
                   R1, [FP, #8]
         STR
                   R0, [R1]
$1238
         LDR
                   R0, [FP, #-4]
         CMP
                   R0, #0
$1248
         BNE
                   $1268
                   R2, [FP, #-4]
$124C
         LDR
         LDR
                   R1, [FP, #-8]
         SUB
                   R0, FP, #12
         STMFD
                   SP!, {R0-R2}
$125C
         BL
$1260
         ADD
                   SP, SP, #12
$1264
$1268
         MOV
                   SP, FP
         LDMFD
                  SP!, {FP, PC}
```

```
void R( int I, int J, int *O )
{
int A, B, C;
*O = I+J;
if( A == 0 )
R( A, B, &C );
}
```

		ADDR	DATA	
		\$FF9C	5	С
		\$FFA0	21	В
		\$FFA4	12	Α
		\$FFA8	\$FFC8	FP"
		\$FFAC	\$1260	LR(PC)
SP	\$FF9C	\$FFB0	\$FFBC	&C
		\$FFB4	7	В
		\$FFB8	0	Α
FP	\$FFA8	\$FFBC	7	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$1238	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	Χ
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	

\$FFF4 \$FFF8 *0

Eseguo le istruzioni in R (non esplicitate quelle che assegnano i valori ad A, B e C) Viene assegnato al parametro di uscita O il valore I+J. Dato che O è stato passato per indirizzo, di fatto viene modificato direttamente il valore della variabile C (variabile locale della prima istanza di R, il cui indirizzo è in FP+8)

\$1200 \$120C	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #12
	LDR LDR ADD LDR STR	R0, [FP, #16] R1, [FP, #12] R0, R0, R1 R1, [FP, #8] R0, [R1]
\$1238	LDR CMP	R0, [FP, #-4] R0, #0
\$1248	BNE	\$1268
\$124C	LDR LDR SUB STMFD	R2, [FP, #-4] R1, [FP, #-8] R0, FP, #12 SP!, {R0-R2}
\$125C	BL	R
\$1260 \$1264	ADD	SP, SP, #12
\$1268	MOV LDMFD	SP, FP SP!, {FP, PC}

```
void R( int I, int J, int *O )
{
int A, B, C;
*O = I+J;
if( A == 0 )
R( A, B, &C );
}
```

		ADDR	DATA						
		\$FF9C	5	С	Viene valu	ıtata la con	dizione A==	0	
		\$FFA0	21	В					
		\$FFA4	12	Α					
		\$FFA8	\$FFC8	FP"					
		\$FFAC	\$1260	LR(PC)					
SP	\$FF9C	\$FFB0	\$FFBC	&C		\$1200	STMFD	SP!, {FP,LR}	void R(int I, int J, int *O)
		\$FFB4	7	В			MOV	FP, SP	{
		\$FFB8	0	Α			SUB	SP, SP, #12	int A, B, C;
FP	\$FFA8	\$FFBC	7	С		\$120C			*O = I+J;
		\$FFC0	7	В			LDR	R0, [FP, #16]	if(A == 0)
		\$FFC4	0	Α			LDR	R1, [FP, #12]	R(A, B, &C);
PC	\$1248	\$FFC8	\$FFEC	FP'			ADD	R0, R0, R1	}
		\$FFCC	\$1154	LR(PC)			LDR	R1, [FP, #8]	
		\$FFD0	\$FFDC	&Z			STR	R0, [R1]	
LR	\$1260	\$FFD4	27	Υ		\$1238			
		\$FFD8	111	Х			LDR	R0, [FP, #-4]	
		\$FFDC	138	Z			CMP	R0, #0	
		\$FFE0	27	Υ		\$1248	BNE	\$1268	
		\$FFE4	111	Х					
		\$FFE8	12	W		\$124C	LDR	R2, [FP, #-4]	
		\$FFEC	\$FFF8				LDR	R1, [FP, #-8]	
		\$FFF0	\$400				SUB	R0, FP, #12	
		\$FFF4				0.4050	STMFD	SP!, {R0-R2}	
		\$FFF8				\$125C	BL	R	
						\$1260	ADD	SP, SP, #12	
						\$1264	140)/	0D ED	
						\$1268	MOV	SP, FP	
							LDMFD	SP!, {FP, PC}	

		ADDR	DATA	- 1
		\$FF9C	5	
		\$FFA0	21	
		\$FFA4	12	
		\$FFA8	\$FFC8	FP"
		\$FFAC	\$1260	LR(PC)
SP	\$FFA8	\$FFB0	\$FFBC	&C
		\$FFB4	7	В
		\$FFB8	0	Α
FP	\$FFA8	\$FFBC	7	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$1268	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	Χ
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	Χ
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

La condizione A==0 è falsa: non viene eseguita la chiamata ricorsiva alla subroutine R Vengono eseguite le istruzioni successive fino al termine della funzione R quindi esco da R. Ripristino il vecchio SP leggendolo da FP: ho de-allocato lo spazio delle variabili locali

\$1200 \$120C	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #12
V .200	LDR LDR ADD LDR STR	R0, [FP, #16] R1, [FP, #12] R0, R0, R1 R1, [FP, #8] R0, [R1]
\$1238	LDR CMP	R0, [FP, #-4] R0, #0
\$1248	BNE	\$1268
\$124C	LDR LDR SUB STMFD	R2, [FP, #-4] R1, [FP, #-8] R0, FP, #12 SP!, {R0-R2}
\$125C	BL	R
\$1260 \$1264	ADD	SP, SP, #12
\$1268	MOV	SP, FP
	LDMFD	SP!, {FP, PC}

```
void R( int I, int J, int *O )
{
  int A, B, C;
  *O = I+J;
  if( A == 0 )
  R( A, B, &C );
}
```

		ADDR	DATA	
		\$FF9C	5	
		\$FFA0	21	
		\$FFA4	12	
		\$FFA8	\$FFC8	
		\$FFAC	\$1260	
SP	\$FFB0	\$FFB0	\$FFBC	&C
		\$FFB4	7	В
		\$FFB8	0	Α
FP	\$FFC8	\$FFBC	7	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$1260	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	Χ
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		

\$FFF8

Ripristino il vecchio FP leggendolo dallo stack Ho completato la de-allocazione del frame della (seconda) chiamata a R. Contemporaneamente, ho aggiornato il PC, inserendo il valore che avevo salvato da LR

\$1200 \$120C	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #12
\$120C	LDR	R0, [FP, #16]
	LDR	R1, [FP, #12]
	ADD	R0, R0, R1
	LDR	R1, [FP, #8]
	STR	R0, [R1]
\$1238		
	LDR	R0, [FP, #-4]
04040	CMP	R0, #0
\$1248	BNE	\$1268
\$124C	LDR	R2, [FP, #-4]
,	LDR	R1, [FP, #-8]
	SUB	R0, FP, #12
	STMFD	SP!, {R0-R2}
\$125C	BL	R
\$1260	ADD	SP, SP, #12
\$1264		
\$1268	MOV	SP, FP
	LDMFD	SP!, {FP, PC}

		ADDR	DATA	Ī
		\$FF9C	5	
		\$FFA0	21	
		\$FFA4	12	
		\$FFA8	\$FFC8	
		\$FFAC	\$1260	
SP	\$FFBC	\$FFB0	\$FFBC	
		\$FFB4	7	
		\$FFB8	0	
FP	\$FFC8	\$FFBC	7	С
		\$FFC0	7	В
		\$FFC4	0	Α
PC	\$1264	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	Х
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

Sono tornato alla funzione chiamante: la prima chiamata di R. Tornato dalla funzione, libero la memoria usata per il passaggio dei parametri: incrementando SP.

```
$1200
         STMFD
                   SP!, {FP,LR}
                                          void R( int I, int J, int *O)
                   FP, SP
         MOV
         SUB
                   SP, SP, #12
                                          int A, B, C;
$120C
                                          *O = I+J;
         LDR
                   R0, [FP, #16]
                                          if( A == 0 )
         LDR
                   R1, [FP, #12]
                                          R(A, B, &C);
         ADD
                   R0, R0, R1
         LDR
                   R1, [FP, #8]
         STR
                   R0, [R1]
$1238
         LDR
                   R0, [FP, #-4]
         CMP
                   R0, #0
$1248
         BNE
                   $1268
$124C
         LDR
                   R2, [FP, #-4]
         LDR
                   R1, [FP, #-8]
         SUB
                   R0, FP, #12
         STMFD
                   SP!, {R0-R2}
$125C
         BL
                   R
$1260
         ADD
                   SP, SP, #12
$1264
$1268
         MOV
                   SP, FP
                   SP!, {FP, PC}
         LDMFD
```

		ADDR \$FF9C \$FFA0 \$FFA4 \$FFA8	DATA 5 21 12 \$FFC8	
SP	\$FFC8	\$FFAC \$FFB0 \$FFB4	\$1260 \$FFBC 7	
		\$FFB8	0	
FP	\$FFC8	\$FFBC	7	
		\$FFC0 \$FFC4	7 0	
PC	\$1268	\$FFC8	\$FFEC	FP'
		\$FFCC	\$1154	LR(PC)
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	Χ
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

La funzione R riprende la sua esecuzione.

Il suo frame è uguale al momento in cui ha chiamato R, ad eccezione della variabile C che è stata modificata dalla subroutine, poichè era stata passata per indirizzo. Al termine delle istruzioni di R, chiudo la funzione: ripristino il vecchio SP leggendolo da FP.

\$1200	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #12
\$120C		
	LDR LDR ADD	R0, [FP, #16] R1, [FP, #12] R0, R0, R1
	LDR	R1, [FP, #8]
	STR	R0, [R1]
\$1238		
	LDR	R0, [FP, #-4]
	CMP	R0, #0
\$1248	BNE	\$1268
\$124C	LDR LDR SUB STMFD	R2, [FP, #-4] R1, [FP, #-8] R0, FP, #12 SP!, {R0-R2}
\$125C	BL	R
\$1260 \$1264	ADD	SP, SP, #12
\$1268	MOV	SP, FP
	LDMFD	SP!, {FP, PC}

```
void R( int I, int J, int *O )
{
int A, B, C;
*O = I+J;
if( A == 0 )
R( A, B, &C );
}
```

		ADDR	DATA	_
		\$FF9C	5	
		\$FFA0	21	
		\$FFA4	12	
		\$FFA8	\$FFC8	
		\$FFAC	\$1260	
SP	\$FFD0	\$FFB0	\$FFBC	
		\$FFB4	7	
		\$FFB8	0	
FP	\$FFEC	\$FFBC	7	
		\$FFC0	7	
		\$FFC4	0	
PC	\$1154	\$FFC8	\$FFEC	
		\$FFCC	\$1154	
		\$FFD0	\$FFDC	&Z
LR	\$1260	\$FFD4	27	Υ
		\$FFD8	111	Х
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	Х
		\$FFE8	12	W
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

Ripristino FP leggendolo dallo stack Ho terminato la de-allocazione del frame di R. Contemporaneamente, ho aggiornato il PC, inserendo il valore che avevo salvato da LR

\$1200	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #12	<pre>void R(int I, int J, int *O) { int A, B, C;</pre>
\$120C			*O = I+J;
	LDR LDR	R0, [FP, #16] R1, [FP, #12]	if(A == 0) R(A, B, &C);
	ADD	R0, R0, R1	}
	LDR STR	R1, [FP, #8] R0, [R1]	
\$1238			
	LDR CMP	R0, [FP, #-4] R0, #0	
\$1248	BNE	\$1268	
\$124C	LDR LDR SUB STMFD	R2, [FP, #-4] R1, [FP, #-8] R0, FP, #12 SP!, {R0-R2}	
\$125C	BL	R	
\$1260 \$1264	ADD	SP, SP, #12	
\$1268	MOV	SP, FP	
	LDMFD	SP!, {FP, PC}	

		ADDR	DATA	
		\$FF9C	5	
		\$FFA0	21	
		\$FFA4	12	
		\$FFA8	\$FFC8	
		\$FFAC	\$1260	
SP	\$FFDC	\$FFB0	\$FFBC	
		\$FFB4	7	
		\$FFB8	0	
FP	\$FFEC	\$FFBC	7	
		\$FFC0	7	
		\$FFC4	0	
PC	\$1158	\$FFC8	\$FFEC	
		\$FFCC	\$1154	
		\$FFD0	\$FFDC	
LR	\$1260	\$FFD4	27	
		\$FFD8	111	
		\$FFDC	138	Z
		\$FFE0	27	Υ
		\$FFE4	111	X
		\$FFE8	12	V
		\$FFEC	\$FFF8	
		\$FFF0	\$400	
		\$FFF4		
		\$FFF8		

Sono tornato alla funzione chiamante, MAIN. Tornato dalla funzione, libero la memoria usata per il passaggio dei parametri incrementando SP.

\$1100	STMFD MOV SUB	SP!, {FP,LR} FP, SP SP, SP, #16	int main() { int W, X, Y, Z;
\$110C			R(X, Y, &Z);
\$1140	LDR LDR SUB STMFD	R2, [FP, #-8] R1, [FP, #-12] R0, FP, #16 SP!, {R0-R2}	1
\$1150	BL	R	
\$1154	ADD	SP, SP, #12	
\$1158			
	MOV LDMFD	SP, FP SP!, {FP, PC}	

		ADDR	DATA
		\$FF9C	5
		\$FFA0	21
		\$FFA4	12
		\$FFA8	\$FFC8
		\$FFAC	\$1260
SP	\$FFF4	\$FFB0	\$FFBC
		\$FFB4	7
		\$FFB8	0
FP	\$FFF8	\$FFBC	7
		\$FFC0	7
		\$FFC4	0
PC	\$400	\$FFC8	\$FFEC
		\$FFCC	\$1154
		\$FFD0	\$FFDC
LR	\$1260	\$FFD4	27
		\$FFD8	111
		\$FFDC	138
		\$FFE0	27
		\$FFE4	111
		\$FFE8	12
		\$FFEC	\$FFF8
		\$FFF0	\$400
		\$FFF4	
		\$FFF8	
		•	

Proseguo l'esecuzione della funzione MAIN

Al termine delle istruzioni, chiudo la funzione: de-alloco il frame, ripristino il FP precedente e, grazie al load multiplo, contemporaneamente aggiorno il PC.

