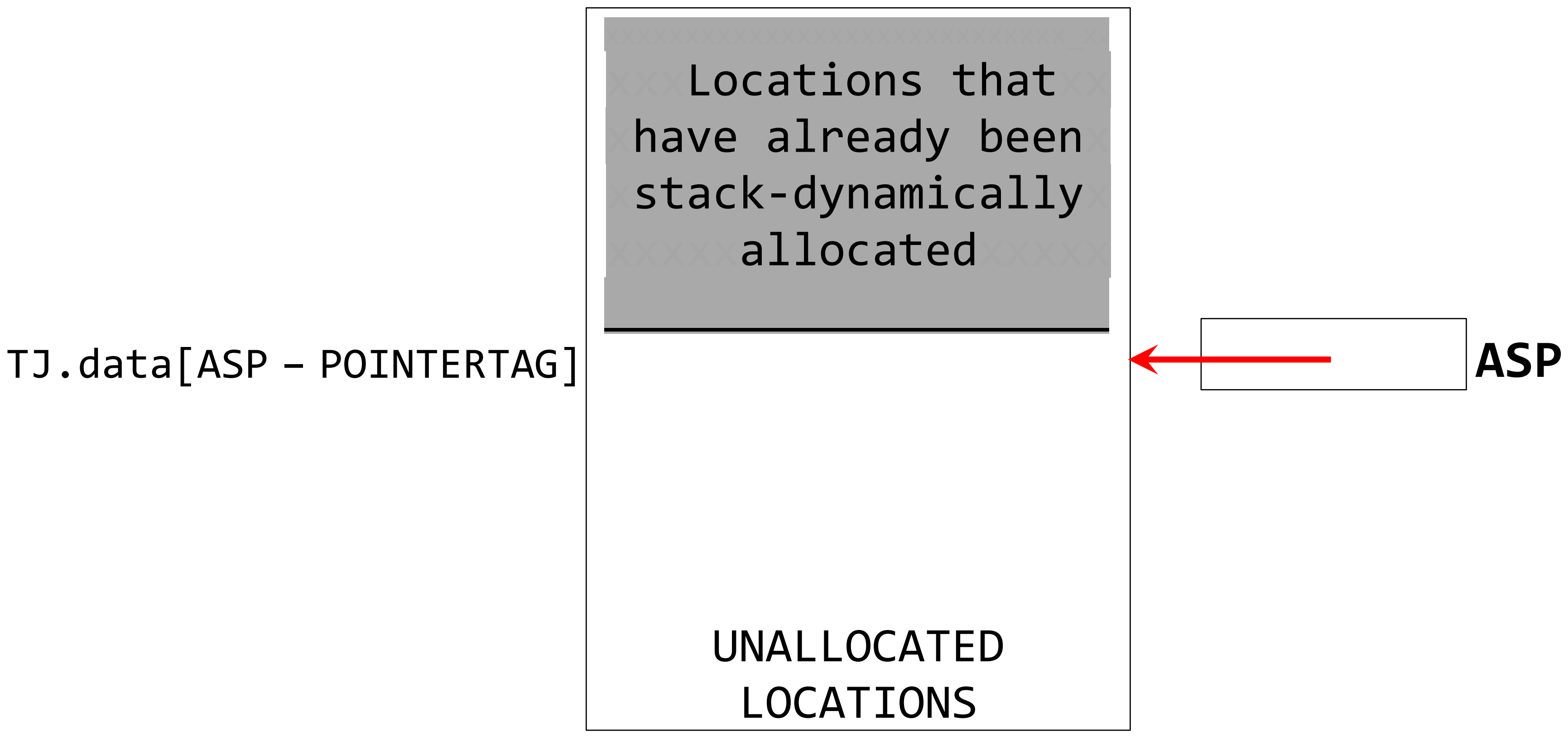
Execution of Method Call and Return

S-PUSH ***y*** is equivalent to:

```
          TJ.data[ASP – POINTERTAG] = y;   ASP++;
```

BEFORE ***execution of*** **S-PUSH** ***y***

Stack-Dynamically Allocated Part of Data Memory



Locations that have already been stack-dynamically allocated

TJ.data[ASP – POINTERTAG]

ASP

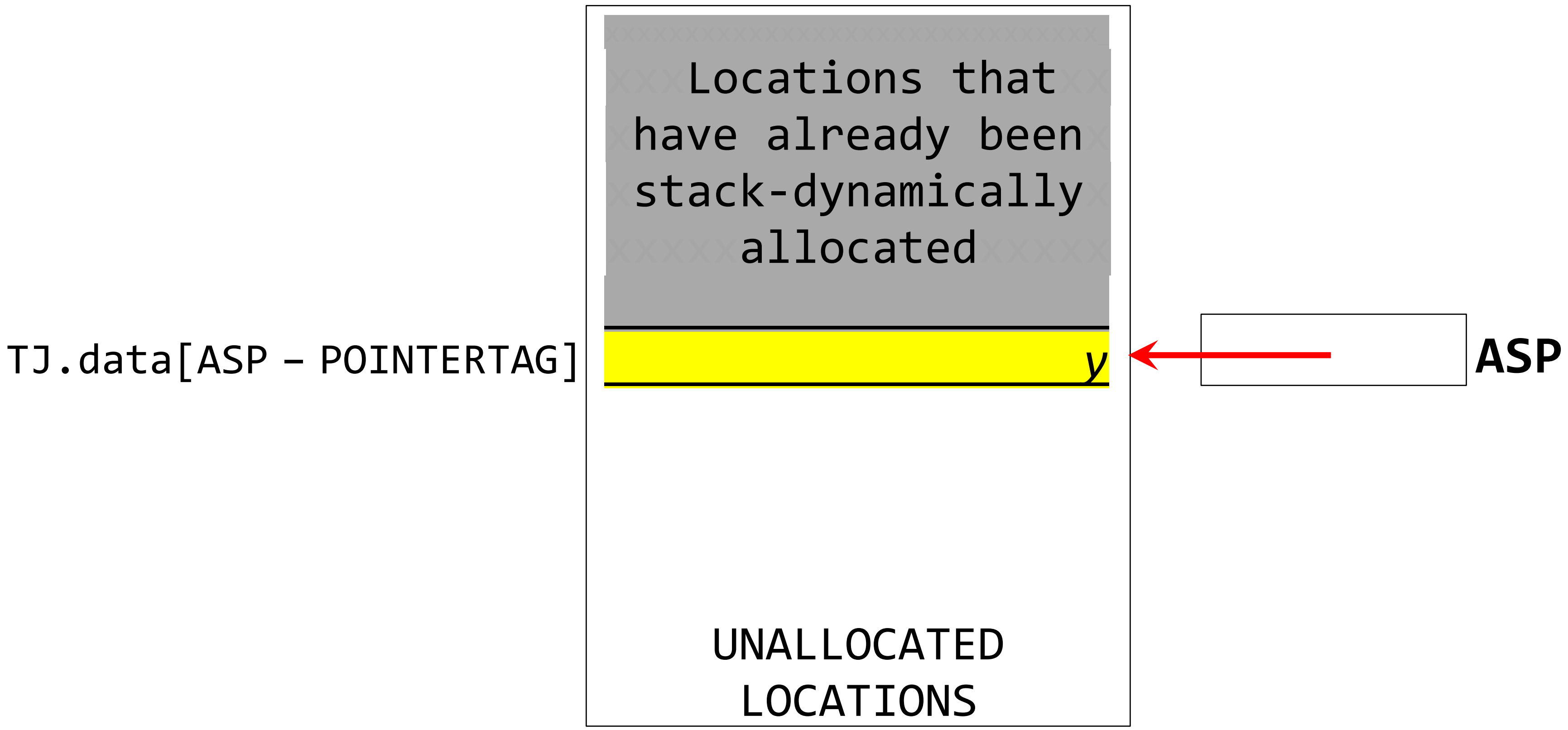
UNALLOCATED LOCATIONS

S-PUSH *y* is equivalent to:

```
TJ.data[ASP – POINTERTAG] = y;   ASP++;
```

AFTER ***execution of*** `TJ.data[ASP – POINTERTAG] = y;`

Stack-Dynamically Allocated Part of Data Memory



Locations that have already been stack-dynamically allocated

TJ.data[ASP – POINTERTAG]

y

ASP

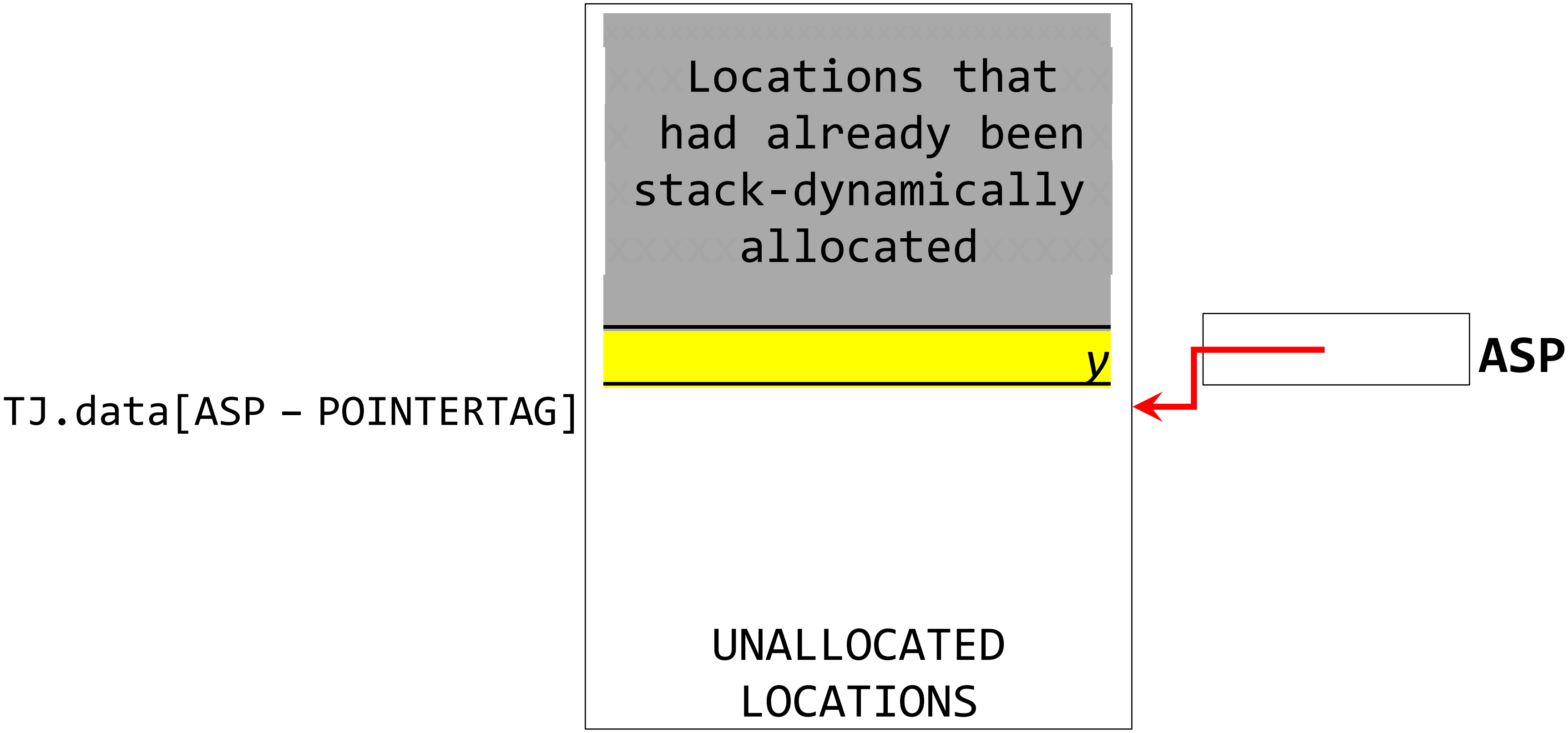
UNALLOCATED LOCATIONS

S-PUSH *y* is equivalent to:

```
            TJ.data[ASP – POINTERTAG] = y;   ASP++;
```

AFTER* execution of S-PUSH *y

Stack-Dynamically Allocated Part of Data Memory



Locations that had already been stack-dynamically allocated

y

ASP

TJ.data[ASP – POINTERTAG]

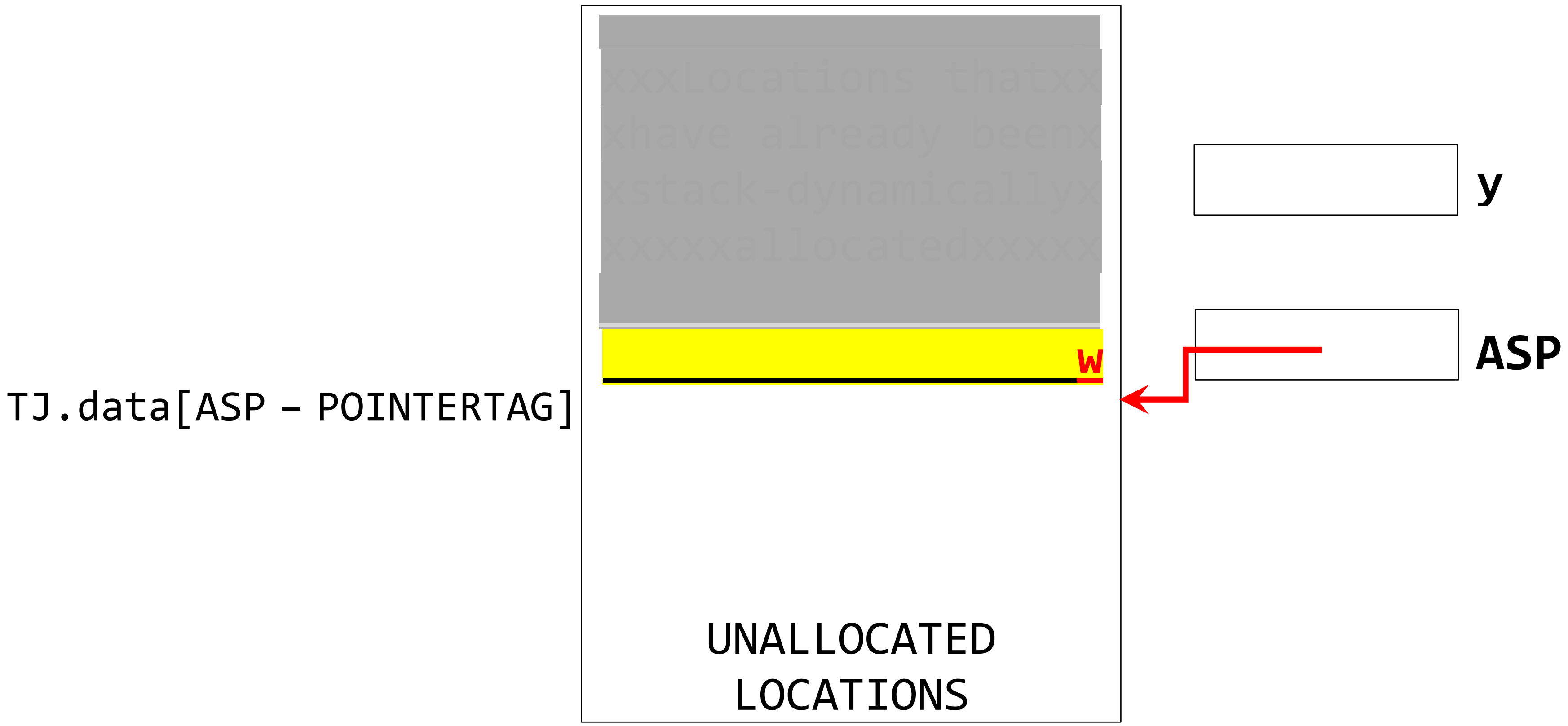
UNALLOCATED LOCATIONS

S-POP *y* is equivalent to:

```
            --ASP;  y = TJ.data[ASP – POINTERTAG];
```

BEFORE* execution of S-POP *y

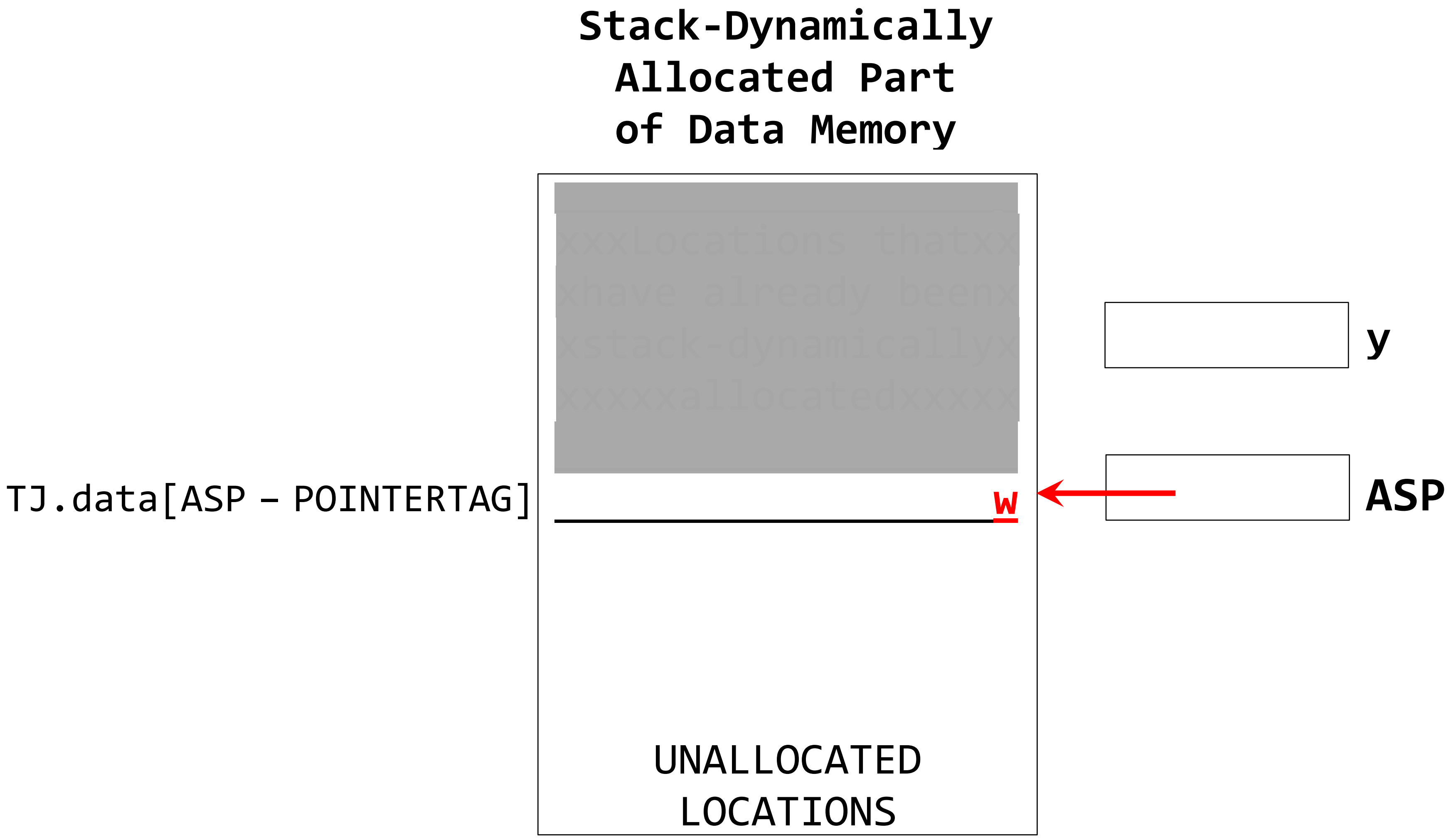
Stack-Dynamically Allocated Part of Data Memory



S-POP *y* is equivalent to:

```
            --ASP;  y = TJ.data[ASP – POINTERTAG];
```

***AFTER* execution of** --ASP;



Stack-Dynamically Allocated Part of Data Memory

xxlocations thatxx
xhave already beenx
xstack-dynamicallyx
xxxxallocatedxxxx

TJ.data[ASP – POINTERTAG]

w

y

ASP

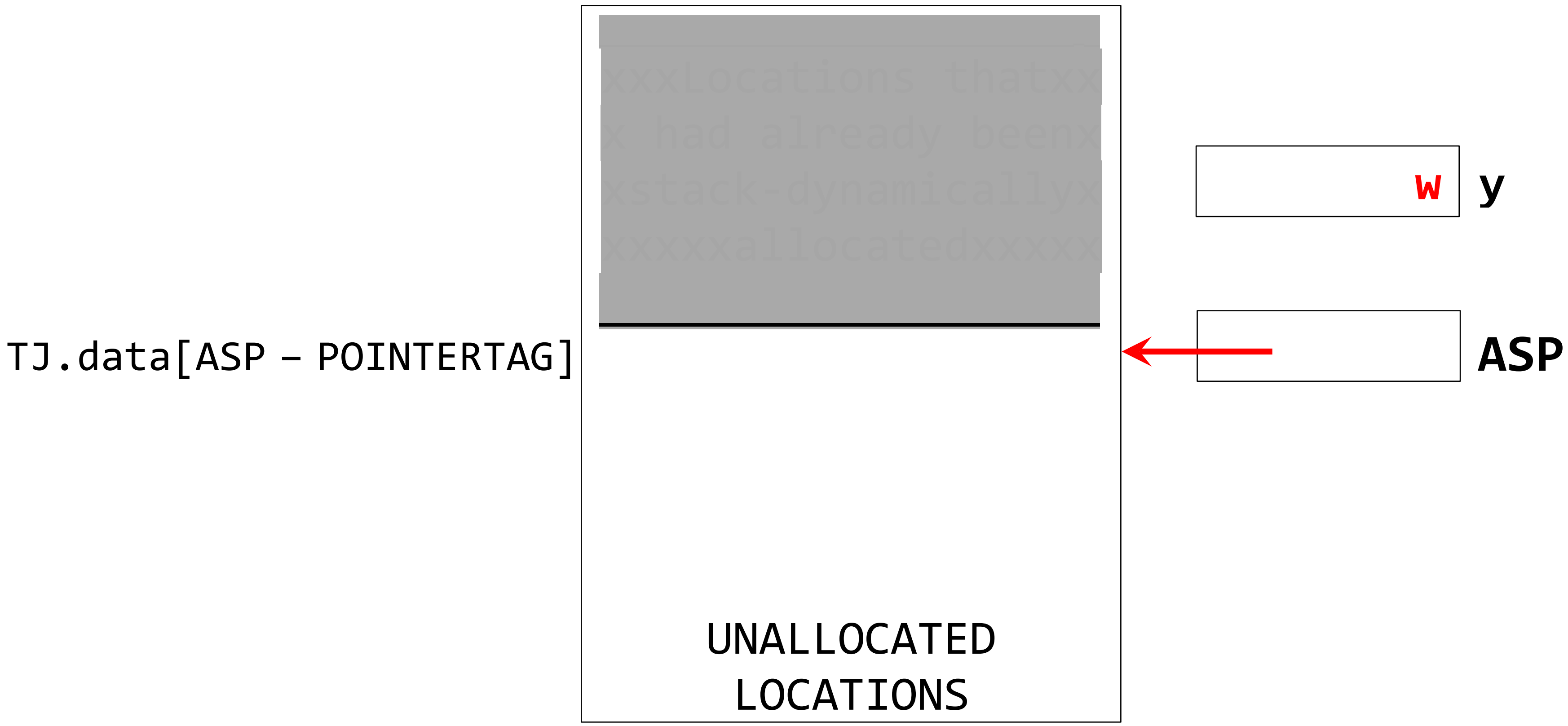
UNALLOCATED LOCATIONS

S-POP *y* is equivalent to:

```
              --ASP;  y = TJ.data[ASP – POINTERTAG];
```

AFTER* execution of S-POP *y

Stack-Dynamically Allocated Part of Data Memory



Diagrams Relating to Sec. 4 on Page 7 of the TinyJ Assignment 3 Document

Suppose a method **f** calls a method **g** as follows

g(17,23,7,3)

--e.g., within: `System.out.print(g(17,23,7,3));`

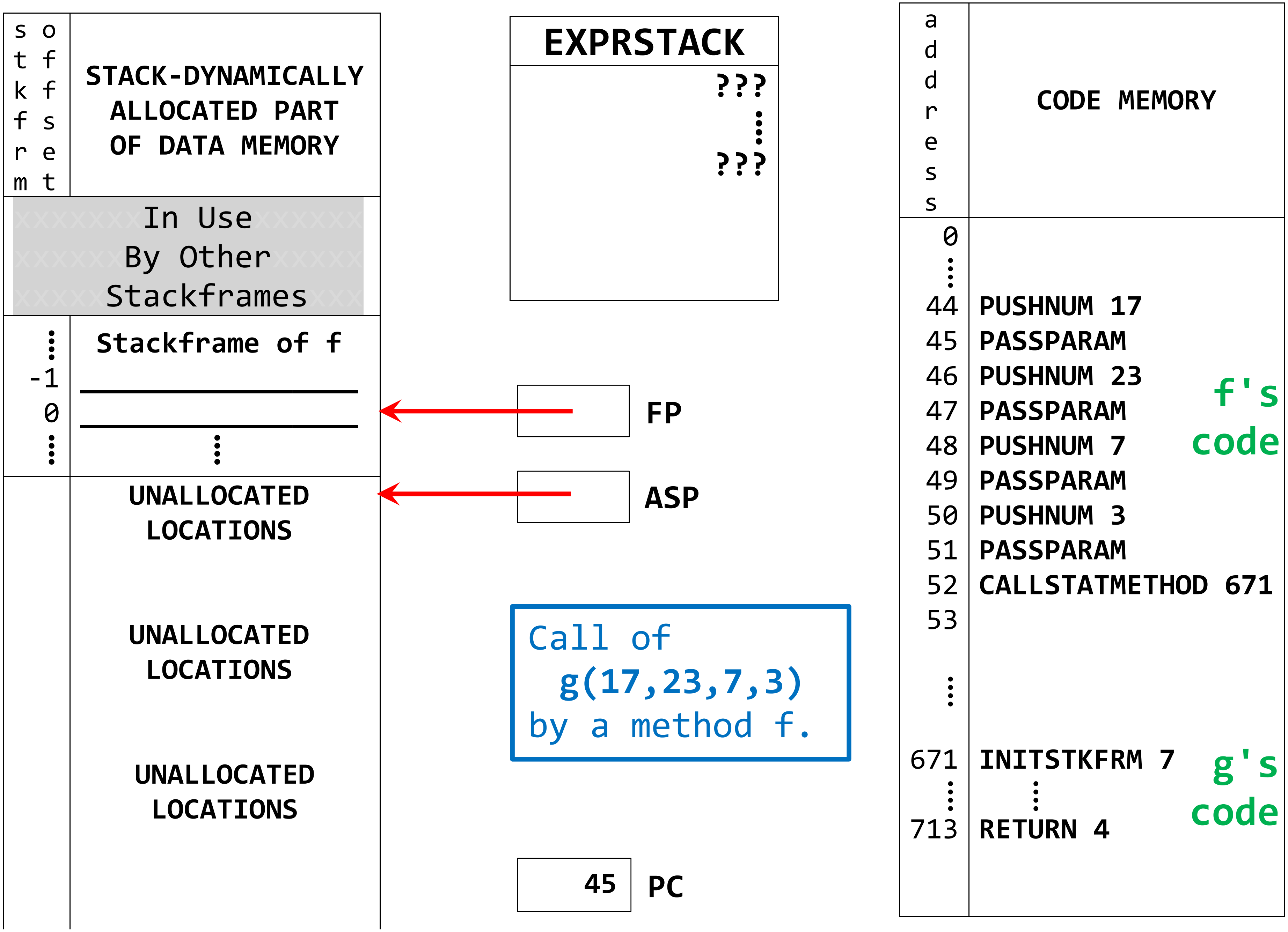
Suppose further that:

1. **7** stackframe locations are allocated for local variables declared in **g**'s body.
2. The code memory address of the first VM instruction generated for method **g** is **671**.
3. Method **g** returns control to its caller by executing:

   713: RETURN 4
4. The code memory address of the first VM instruction generated for **g(17,23,7,3)** is **44**.

The following slides show how the call **g(17,23,7,3)** would be executed, and how **713: RETURN 4** would be executed.

BEFORE Execution of: 44: PUSHNUM 17



stkfrm offset

STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f

⋮

-1

0

⋮

UNALLOCATED LOCATIONS

UNALLOCATED LOCATIONS

UNALLOCATED LOCATIONS

EXPRSTACK

???

⋮

???

FP

ASP

Call of **g(17,23,7,3)** by a method f.

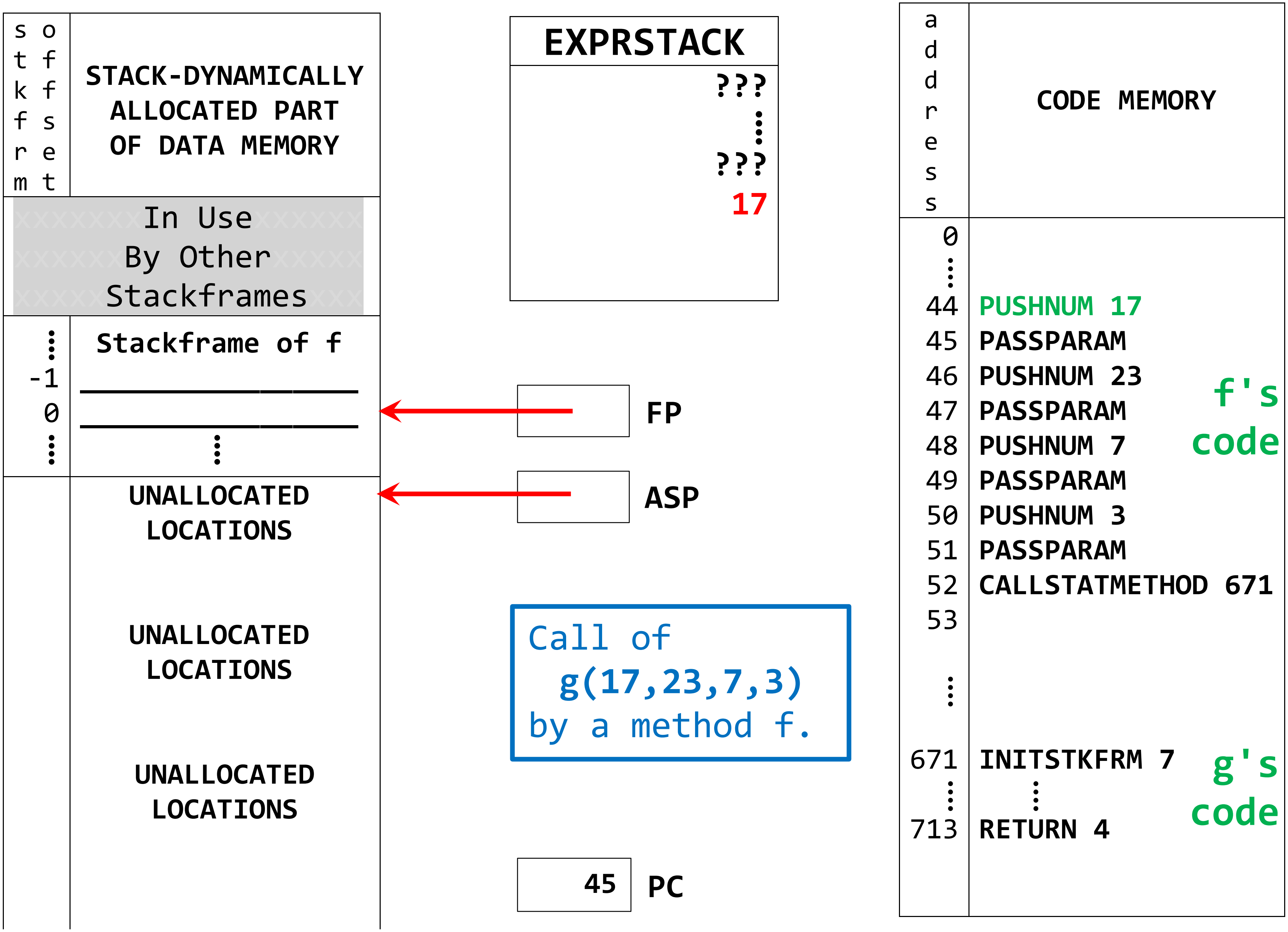
45 PC

address	CODE MEMORY
0	
⋮	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23
47	PASSPARAM
48	PUSHNUM 7
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	
⋮	
671	INITSTKFRM 7
⋮	⋮
713	RETURN 4

f's code

g's code

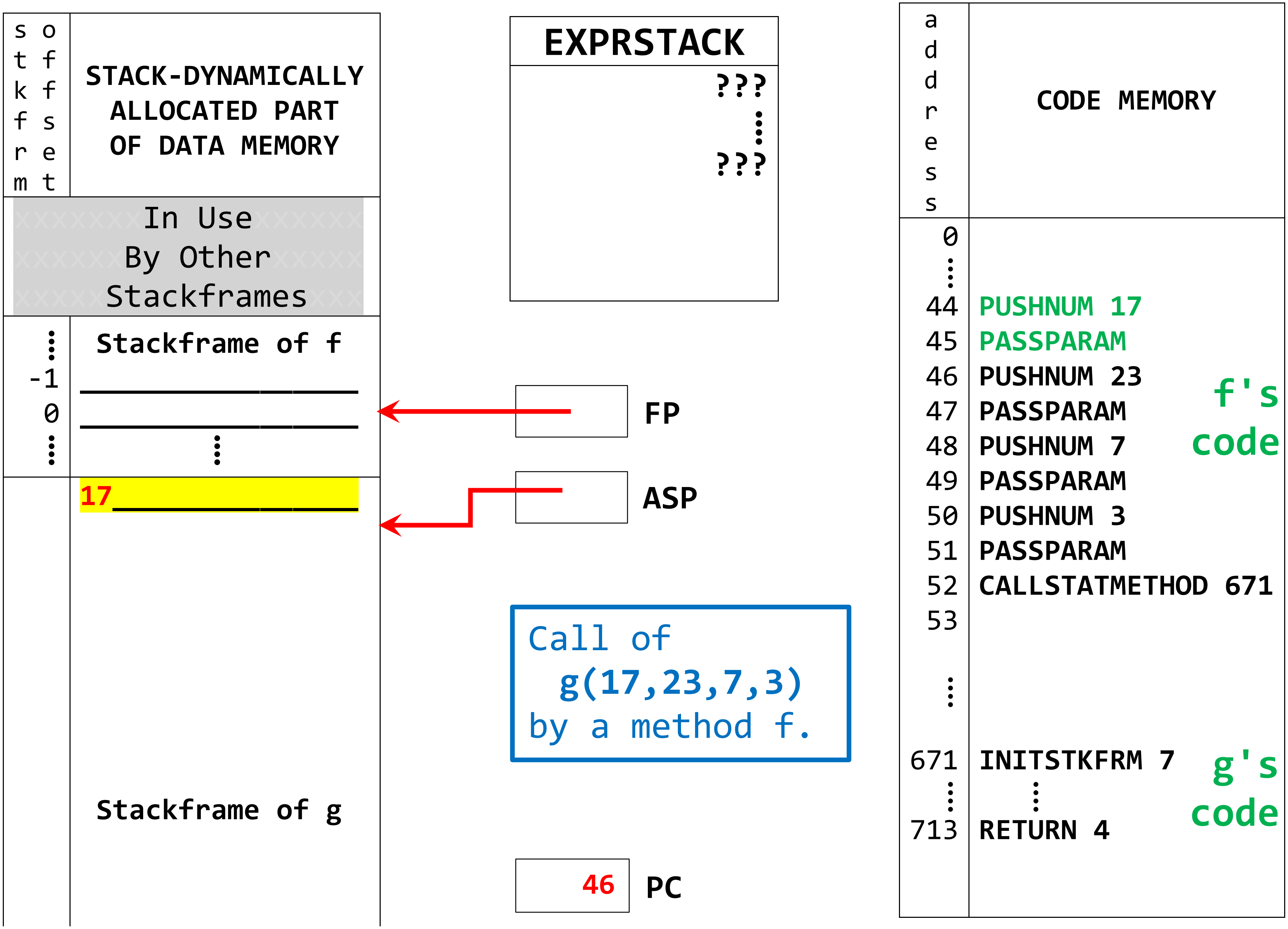
AFTER Execution of: 44: PUSHNUM 17



Call of g(17,23,7,3) by a method f.

```
  0
  ⋮
 44  PUSHNUM 17
 45  PASSPARAM
 46  PUSHNUM 23
 47  PASSPARAM
 48  PUSHNUM 7
 49  PASSPARAM
 50  PUSHNUM 3
 51  PASSPARAM
 52  CALLSTATMETHOD 671
 53
  ⋮
671  INITSTKFRM 7
  ⋮       ⋮
713  RETURN 4
```

AFTER Execution of 45: PASSPARAM



stkfrm

offset

STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f

⋮
-1
0
⋮

17

Stackframe of g

EXPRSTACK

???
⋮
???

FP

ASP

46 PC

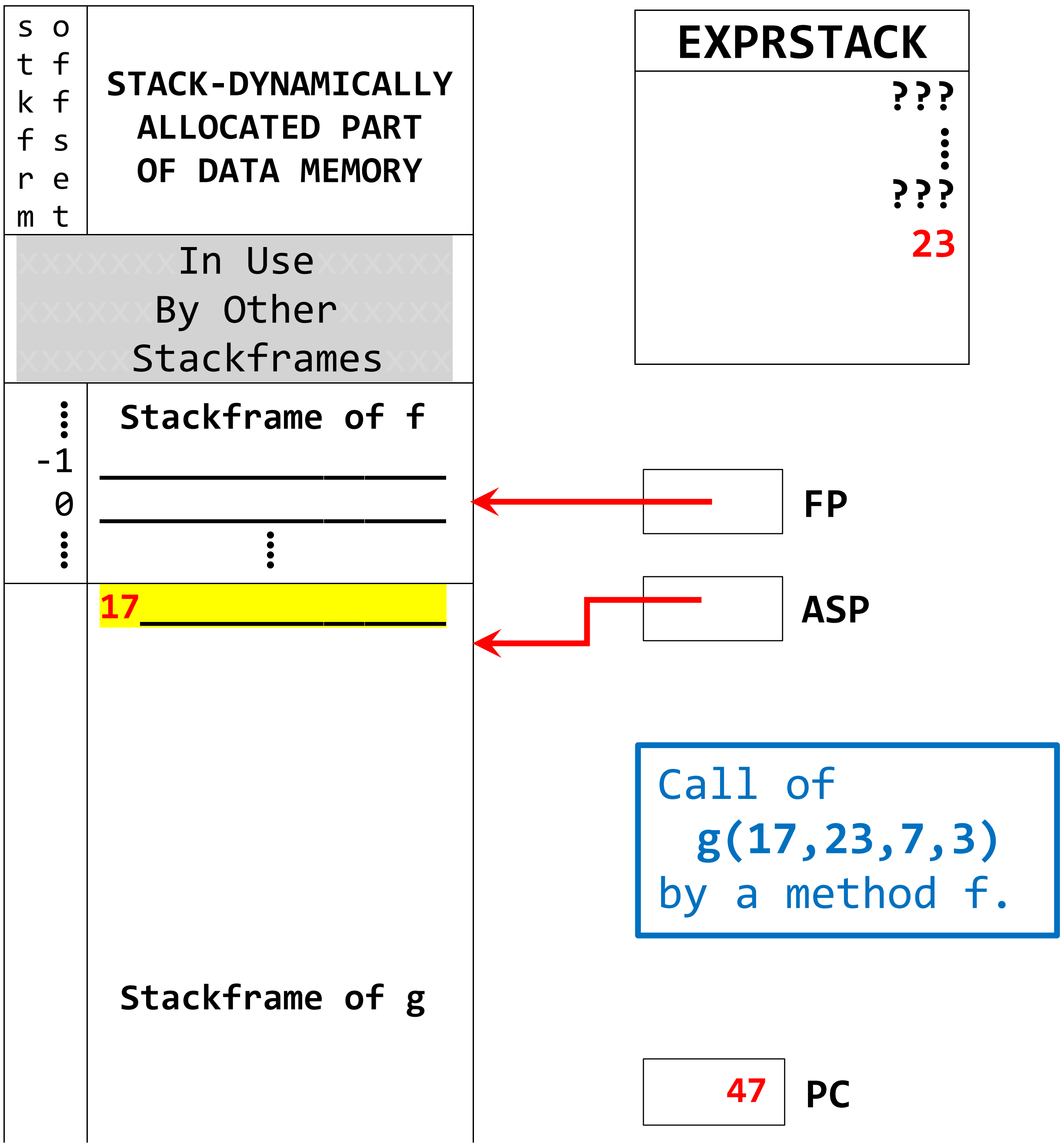
Call of
g(17,23,7,3)
by a method f.

address	CODE MEMORY
0	
⋮	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23
47	PASSPARAM
48	PUSHNUM 7
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	
⋮	
671	INITSTKFRM 7
⋮	⋮
713	RETURN 4

f's code

g's code

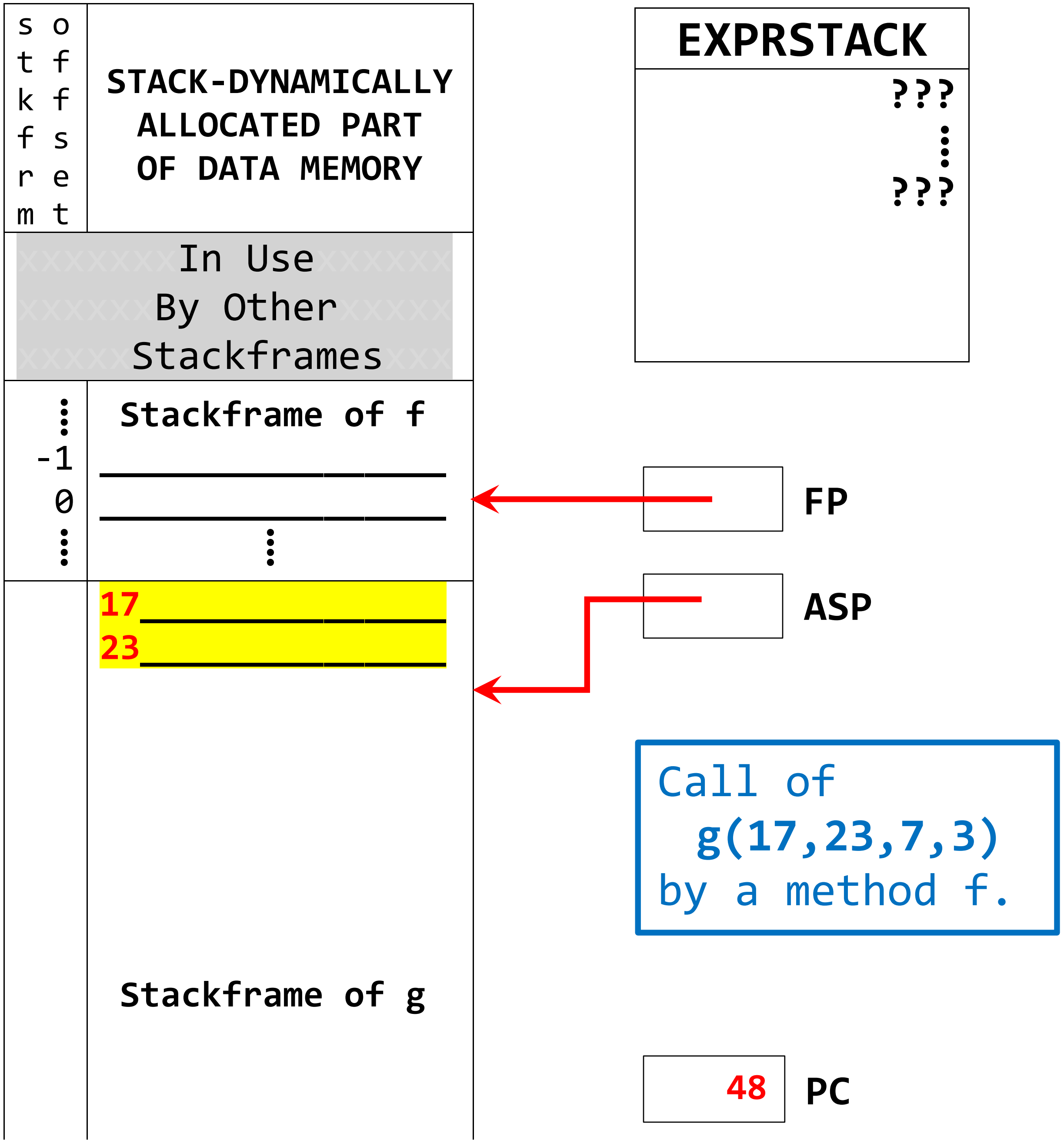
AFTER Execution of 46: PUSHNUM 23


stkfrm
offset
STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY
In Use By Other Stackframes
Stackframe of f
-1
0
17
Stackframe of g
EXPRSTACK
???
???
23
FP
ASP
47
PC

Call of **g(17,23,7,3)** by a method f.

address	CODE MEMORY	
0		
⋮		
44	PUSHNUM 17	
45	PASSPARAM	
46	PUSHNUM 23	f's code
47	PASSPARAM	
48	PUSHNUM 7	
49	PASSPARAM	
50	PUSHNUM 3	
51	PASSPARAM	
52	CALLSTATMETHOD 671	
53		
⋮		
671	INITSTKFRM 7	g's code
⋮	⋮	
713	RETURN 4	

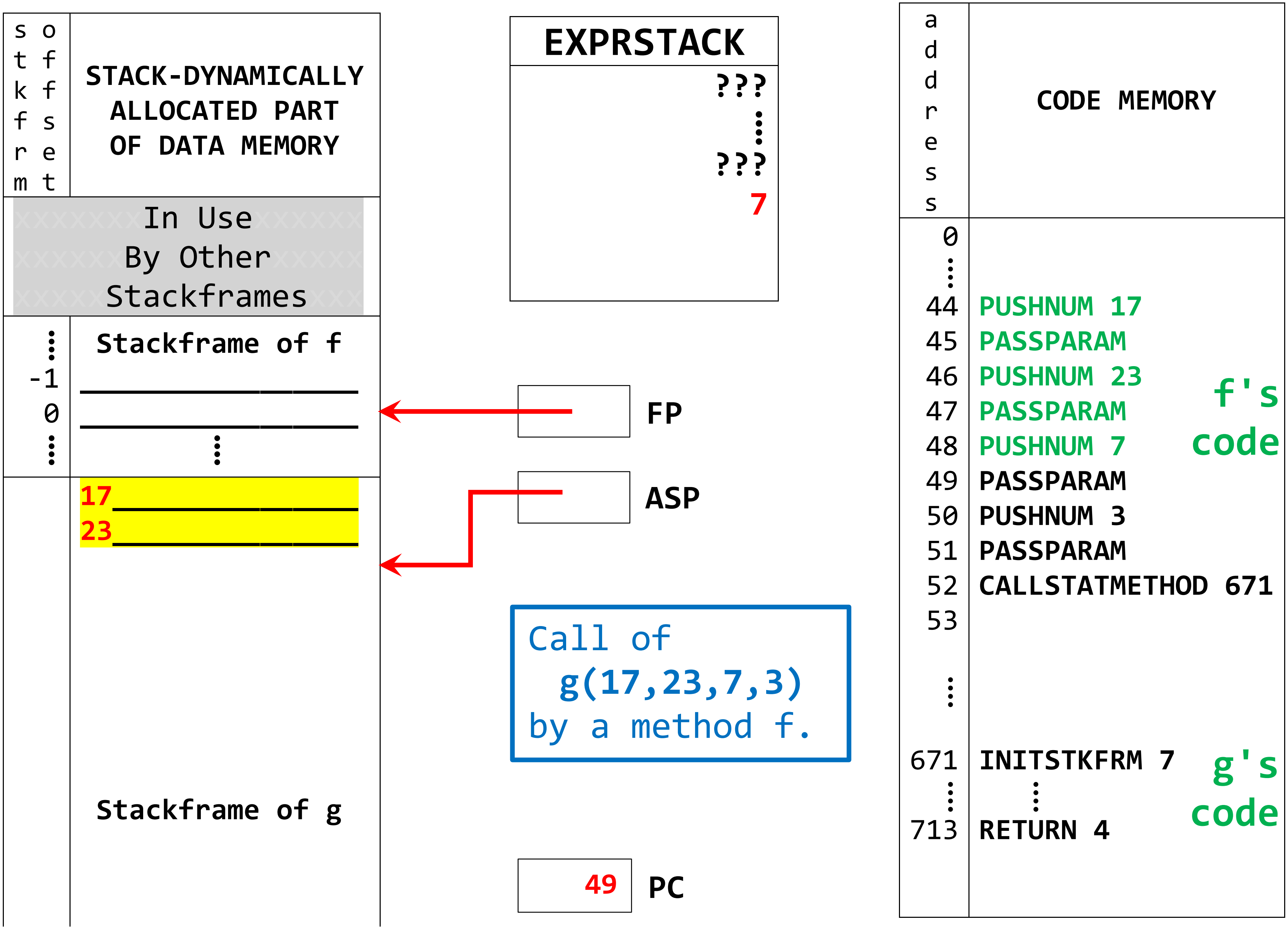
AFTER Execution of 47: PASSPARAM


stkfrm
offset
STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY
In Use By Other Stackframes
Stackframe of f
-1
0
17
23
Stackframe of g
EXPRSTACK
???
???
FP
ASP
48
PC

Call of
g(17,23,7,3)
by a method f.

address	CODE MEMORY	
0		
⋮		
44	PUSHNUM 17	f's code
45	PASSPARAM	
46	PUSHNUM 23	
47	PASSPARAM	
48	PUSHNUM 7	
49	PASSPARAM	
50	PUSHNUM 3	
51	PASSPARAM	
52	CALLSTATMETHOD 671	
53		
⋮		
671	INITSTKFRM 7	g's code
⋮	⋮	
713	RETURN 4	

AFTER Execution of 48: PUSHNUM 7



stkfrm offset	STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY
	In Use By Other Stackframes
⋮	Stackframe of f
-1	
0	
⋮	⋮
	17
	23
	Stackframe of g

EXPRSTACK

???
⋮
???
7

FP

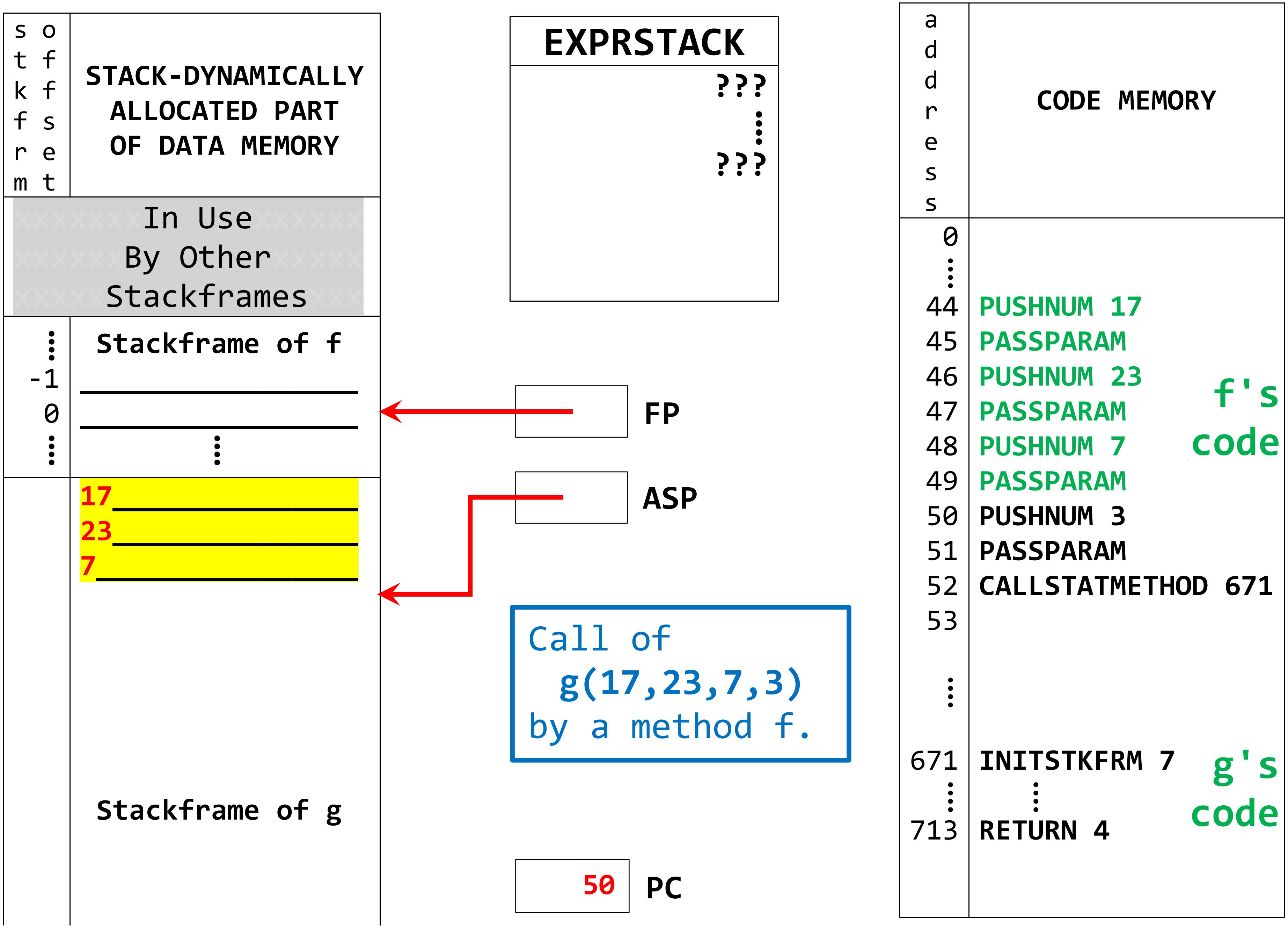
ASP

Call of
g(17,23,7,3)
by a method f.

49 PC

address	CODE MEMORY	
0		
⋮		
44	PUSHNUM 17	
45	PASSPARAM	
46	PUSHNUM 23	f's code
47	PASSPARAM	
48	PUSHNUM 7	
49	PASSPARAM	
50	PUSHNUM 3	
51	PASSPARAM	
52	CALLSTATMETHOD 671	
53		
⋮		
671	INITSTKFRM 7	g's code
⋮	⋮	
713	RETURN 4	

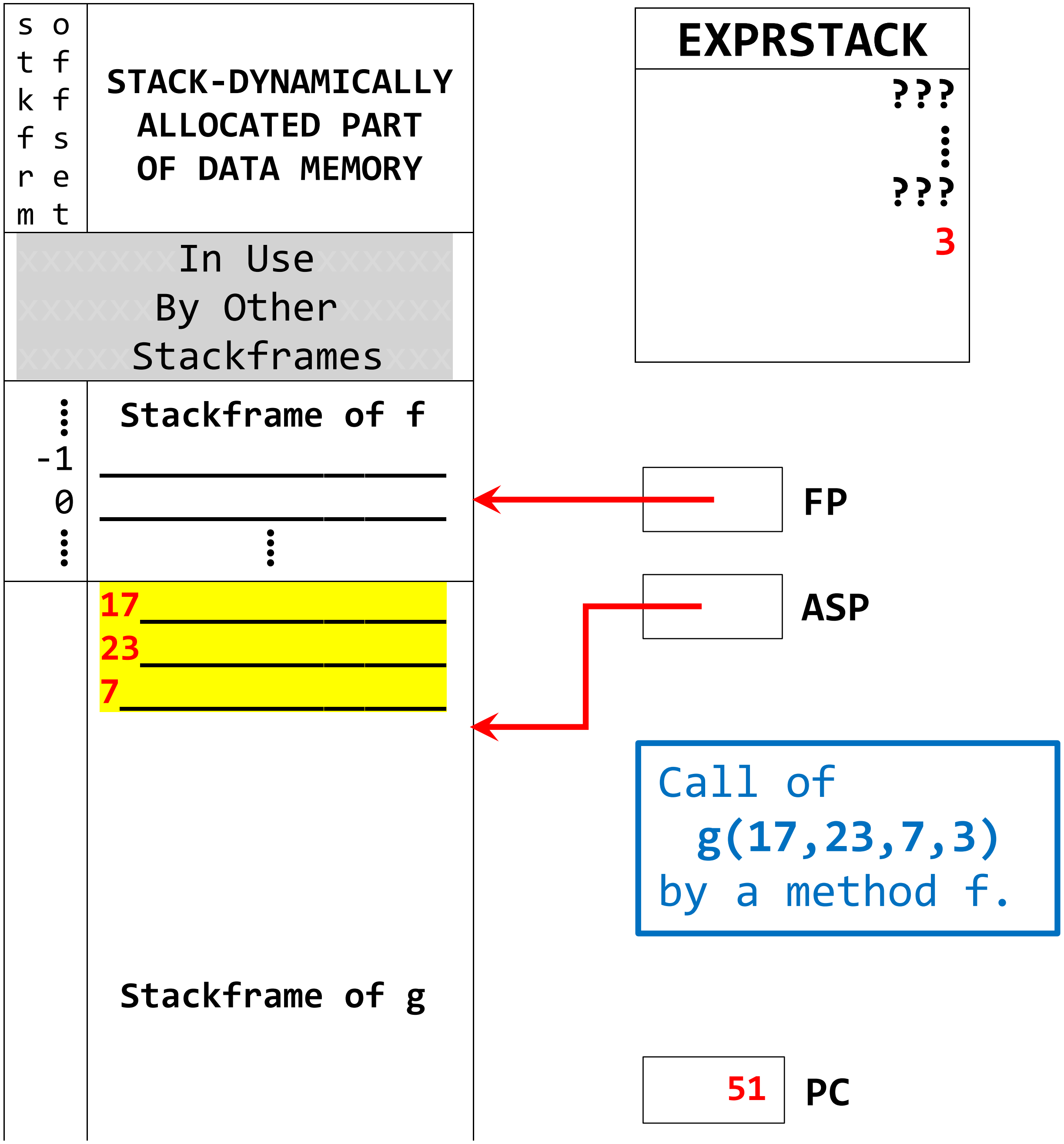
AFTER *Execution of* 49: PASSPARAM



Call of g(17,23,7,3) by a method f.

address	CODE MEMORY
0	
⋮	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23
47	PASSPARAM
48	PUSHNUM 7
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	
⋮	
671	INITSTKFRM 7
⋮	⋮
713	RETURN 4

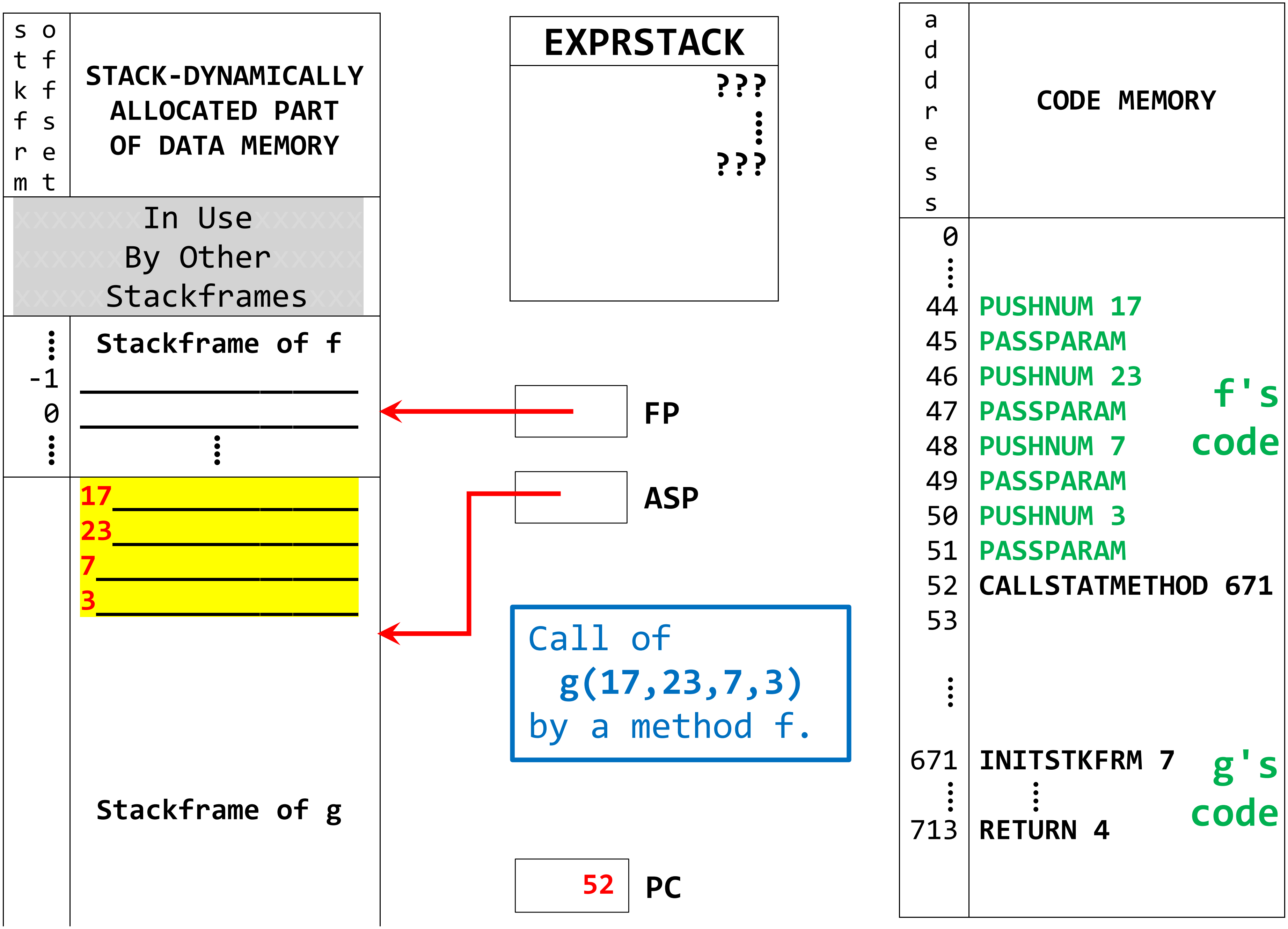
AFTER Execution of 50: PUSHNUM 3



Call of **g(17,23,7,3)** by a method f.

address	CODE MEMORY	
0		
⋮		
44	PUSHNUM 17	
45	PASSPARAM	
46	PUSHNUM 23	f's
47	PASSPARAM	code
48	PUSHNUM 7	
49	PASSPARAM	
50	PUSHNUM 3	
51	PASSPARAM	
52	CALLSTATMETHOD 671	
53		
⋮		
671	INITSTKFRM 7	g's
⋮	⋮	code
713	RETURN 4	

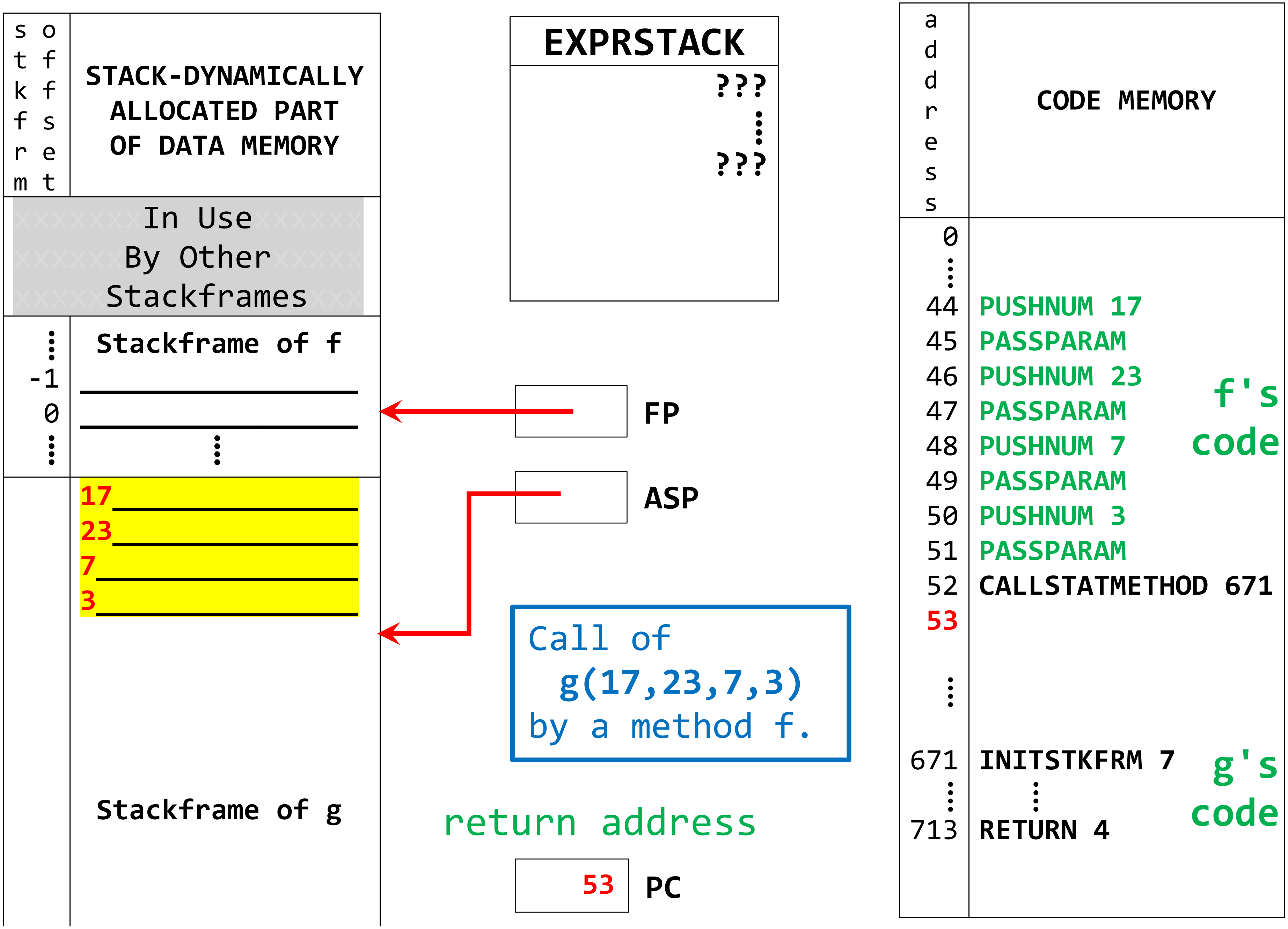
AFTER Execution of 51: PASSPARAM



```
0
⋮
44  PUSHNUM 17
45  PASSPARAM
46  PUSHNUM 23
47  PASSPARAM
48  PUSHNUM 7
49  PASSPARAM
50  PUSHNUM 3
51  PASSPARAM
52  CALLSTATMETHOD 671
53
⋮
671 INITSTKFRM 7
⋮       ⋮
713 RETURN 4
```

Call of
g(17,23,7,3)
by a method f.

After FETCH (BEFORE Execution) of 52: CALLSTATMETHOD 671



```
0
⋮
44  PUSHNUM 17
45  PASSPARAM
46  PUSHNUM 23
47  PASSPARAM
48  PUSHNUM 7
49  PASSPARAM
50  PUSHNUM 3
51  PASSPARAM
52  CALLSTATMETHOD 671
53
⋮
671 INITSTKFRM 7
⋮       ⋮
713 RETURN 4
```

Call of g(17,23,7,3) by a method f.

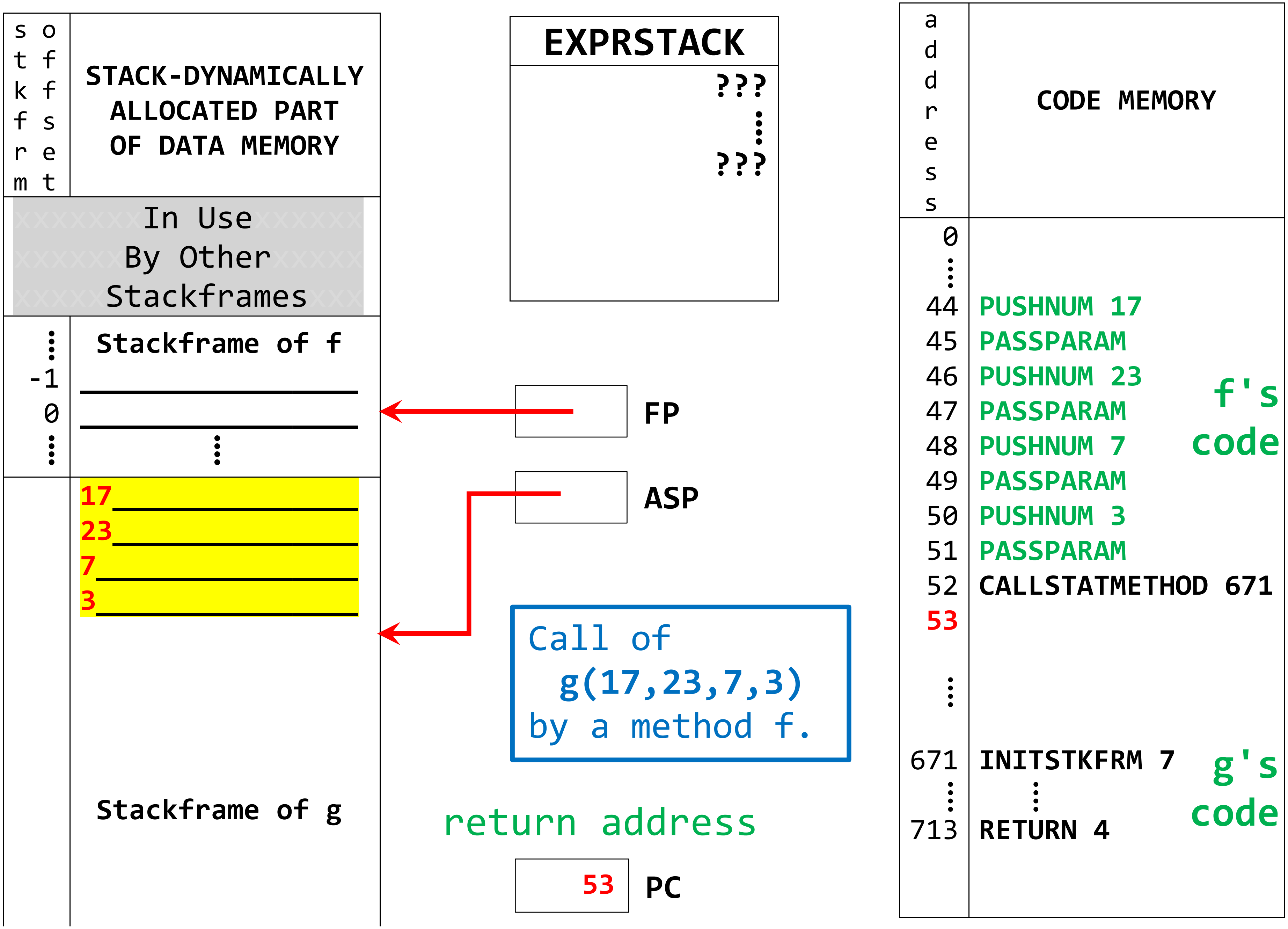
52: CALLSTATMETHOD 671

should **S-PUSH** PC [saves return addr (here, **53**) into new frame*]

and then set PC to 671 [transfers control to g's code]

*into the loc. at offset –1 of the new frame – see p. 4 of:
149.4.211.94/316/Memory-allocation-VM-instruction-set-and-hints-for-asn-2.pdf

After FETCH *(BEFORE Execution) of* 52: CALLSTATMETHOD 671



STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

stkfrm offset

In Use By Other Stackframes

Stackframe of f

-1

0

17

23

7

3

Stackframe of g

EXPRSTACK

???

???

FP

ASP

Call of g(17,23,7,3) by a method f.

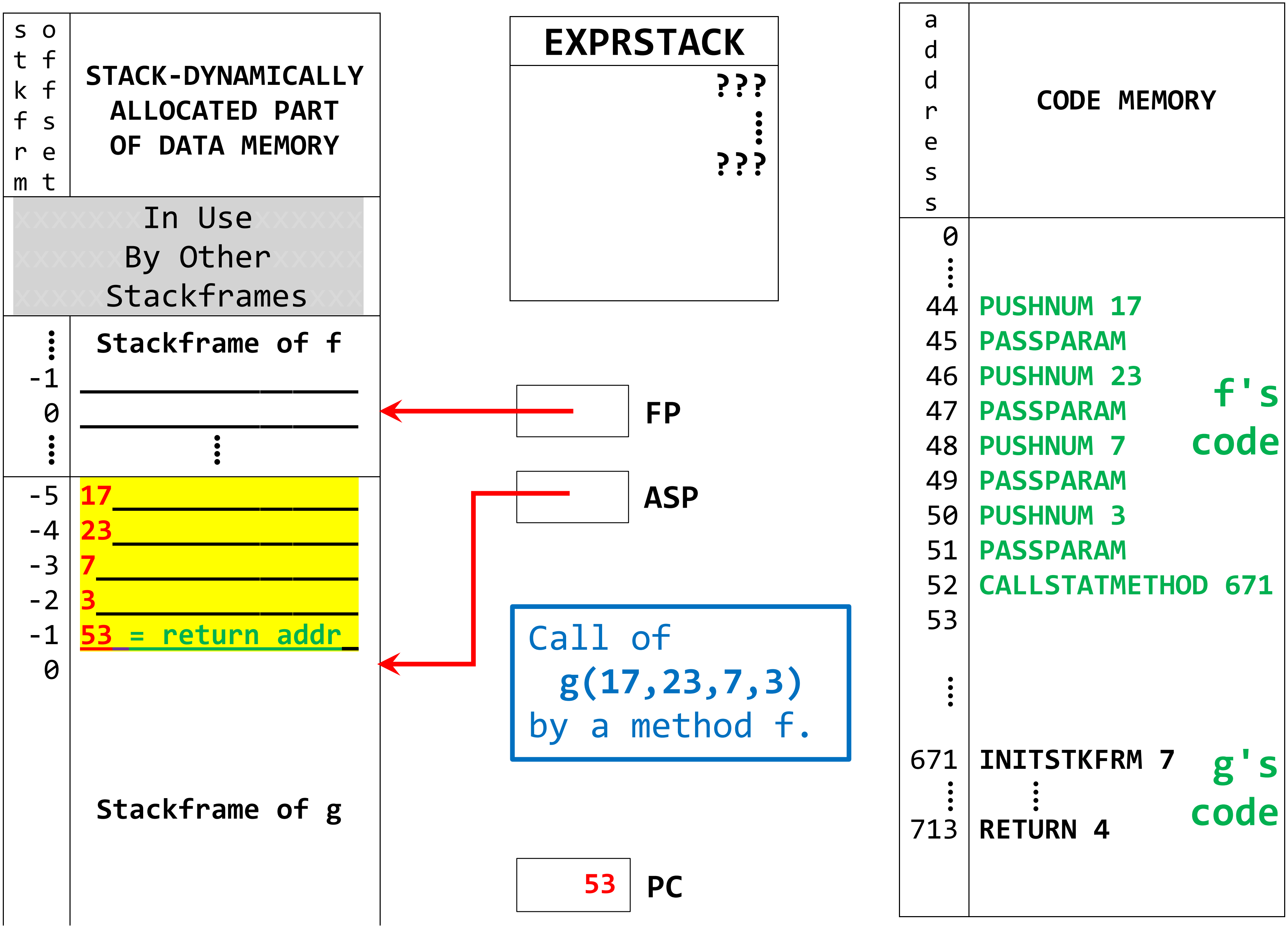
return address

53 PC

CODE MEMORY

```
address
  0
  ⋮
 44  PUSHNUM 17
 45  PASSPARAM
 46  PUSHNUM 23
 47  PASSPARAM        f's
 48  PUSHNUM 7        code
 49  PASSPARAM
 50  PUSHNUM 3
 51  PASSPARAM
 52  CALLSTATMETHOD 671
 53
  ⋮
671  INITSTKFRM 7     g's
  ⋮    ⋮              code
713  RETURN 4
```

AFTER S-PUSH PC *during Execution of* 52: CALLSTATMETHOD 671



stkfrm offset

STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f

stkfrm offset	
⋮	
-1	
0	
⋮	⋮
-5	17
-4	23
-3	7
-2	3
-1	53 = return addr
0	

Stackframe of g

EXPRSTACK

???
⋮
???

FP

ASP

Call of g(17,23,7,3) by a method f.

53 PC

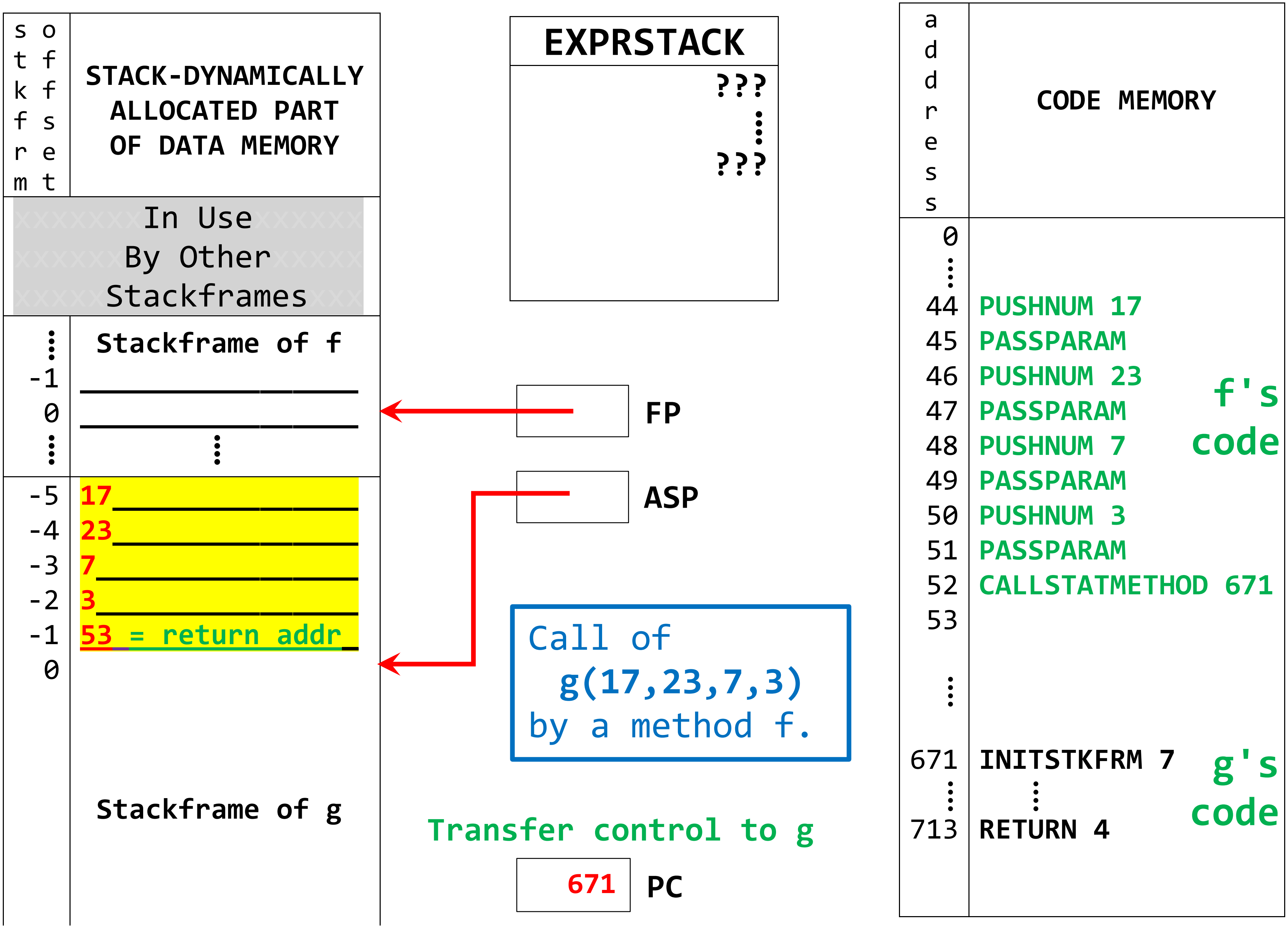
CODE MEMORY

```
address
  0
  ⋮
 44  PUSHNUM 17
 45  PASSPARAM
 46  PUSHNUM 23
 47  PASSPARAM
 48  PUSHNUM 7
 49  PASSPARAM
 50  PUSHNUM 3
 51  PASSPARAM
 52  CALLSTATMETHOD 671
 53
  ⋮
671  INITSTKFRM 7
  ⋮    ⋮
713  RETURN 4
```

f's code

g's code

AFTER Execution of 52: CALLSTATMETHOD 671



stk frm | offset | STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f
-1
0

-5 17
-4 23
-3 7
-2 3
-1 53 = return addr
0

Stackframe of g

EXPRSTACK
???
???

FP

ASP

Call of g(17,23,7,3) by a method f.

Transfer control to g

671 PC

address | CODE MEMORY

```
0
⋮
44  PUSHNUM 17
45  PASSPARAM
46  PUSHNUM 23
47  PASSPARAM
48  PUSHNUM 7
49  PASSPARAM
50  PUSHNUM 3
51  PASSPARAM
52  CALLSTATMETHOD 671
53
⋮
671 INITSTKFRM 7
⋮       ⋮
713 RETURN 4
```

f's code

g's code

671: INITSTKFRM 7

should **S-PUSH** FP [saves caller’s FP at offset 0* in the new frame]

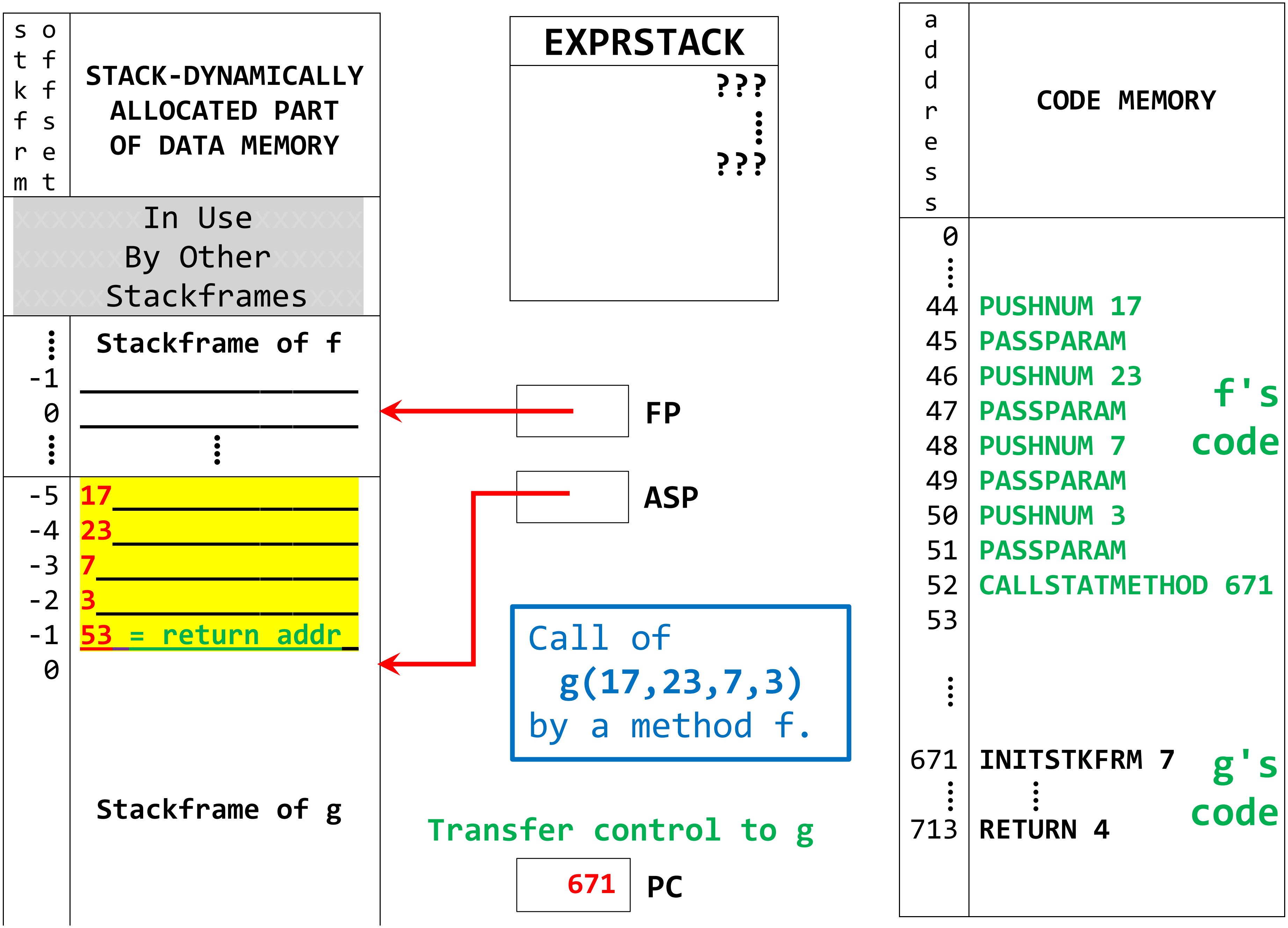
and then set FP to ASP – 1 [makes FP point to offset 0* in the new frame]

and then increase ASP by 7 [allocates space for callee's local variables]

*After the caller's FP is stored at offset 0 in the new frame, the stored pointer is called the ***dynamic link***. See p. 4 of:

149.4.211.94/316/Memory-allocation-VM-instruction-set-and-hints-for-asn-2.pdf

AFTER Execution of 52: CALLSTATMETHOD 671



stk frm offset

STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f

stk frm offset	
⋮	
-1	
0	
⋮	⋮
-5	17
-4	23
-3	7
-2	3
-1	53 = return addr
0	

Stackframe of g

EXPRSTACK

???
⋮
???

FP

ASP

Call of g(17,23,7,3) by a method f.

Transfer control to g

671 PC

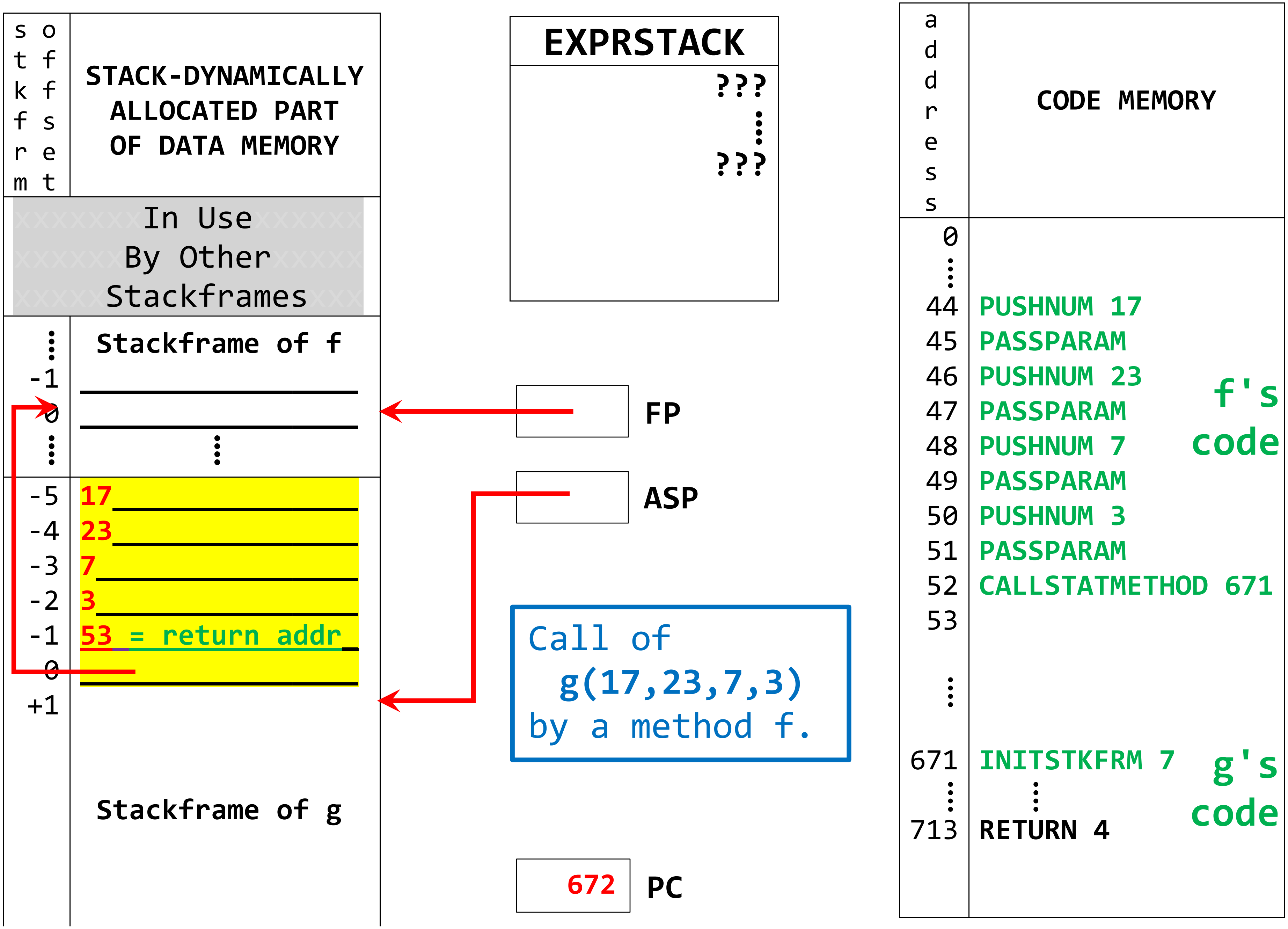
CODE MEMORY

```
address
  0
  ⋮
 44   PUSHNUM 17
 45   PASSPARAM
 46   PUSHNUM 23
 47   PASSPARAM
 48   PUSHNUM 7
 49   PASSPARAM
 50   PUSHNUM 3
 51   PASSPARAM
 52   CALLSTATMETHOD 671
 53
  ⋮
671   INITSTKFRM 7
  ⋮      ⋮
713   RETURN 4
```

f's code

g's code

After S-PUSH FP *during Execution of* 671: INITSTKFRM 7



stkfrm	offset	STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY
		In Use By Other Stackframes
	⋮	Stackframe of f
	-1	
	0	
	⋮	⋮
	-5	17
	-4	23
	-3	7
	-2	3
	-1	53 = return addr
	0	
	+1	
		Stackframe of g

EXPRSTACK

???
⋮
???

FP

ASP

Call of
g(17,23,7,3)
by a method f.

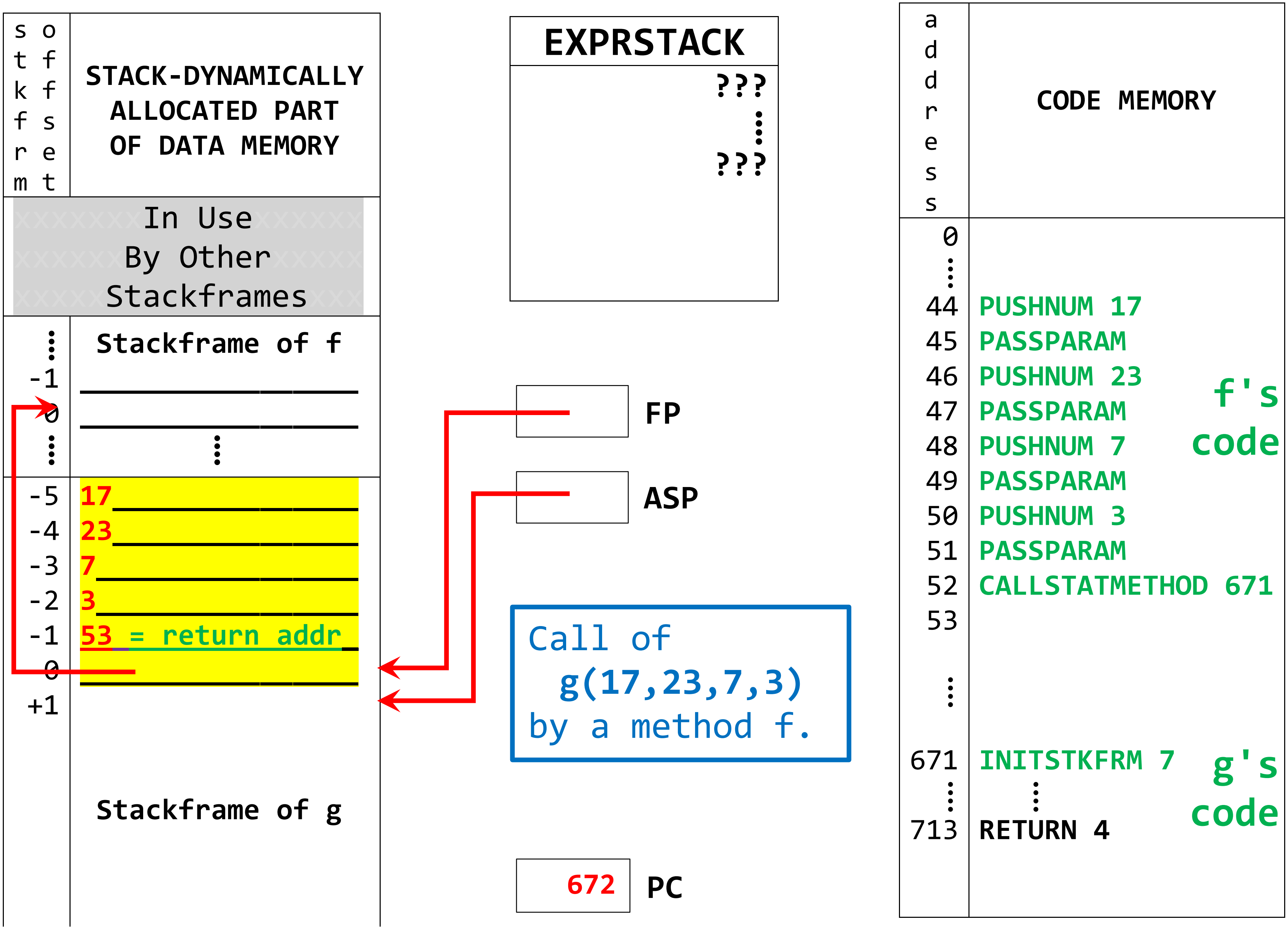
672 PC

address	CODE MEMORY
0	
⋮	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23
47	PASSPARAM
48	PUSHNUM 7
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	
⋮	
671	INITSTKFRM 7
⋮	⋮
713	RETURN 4

f's code

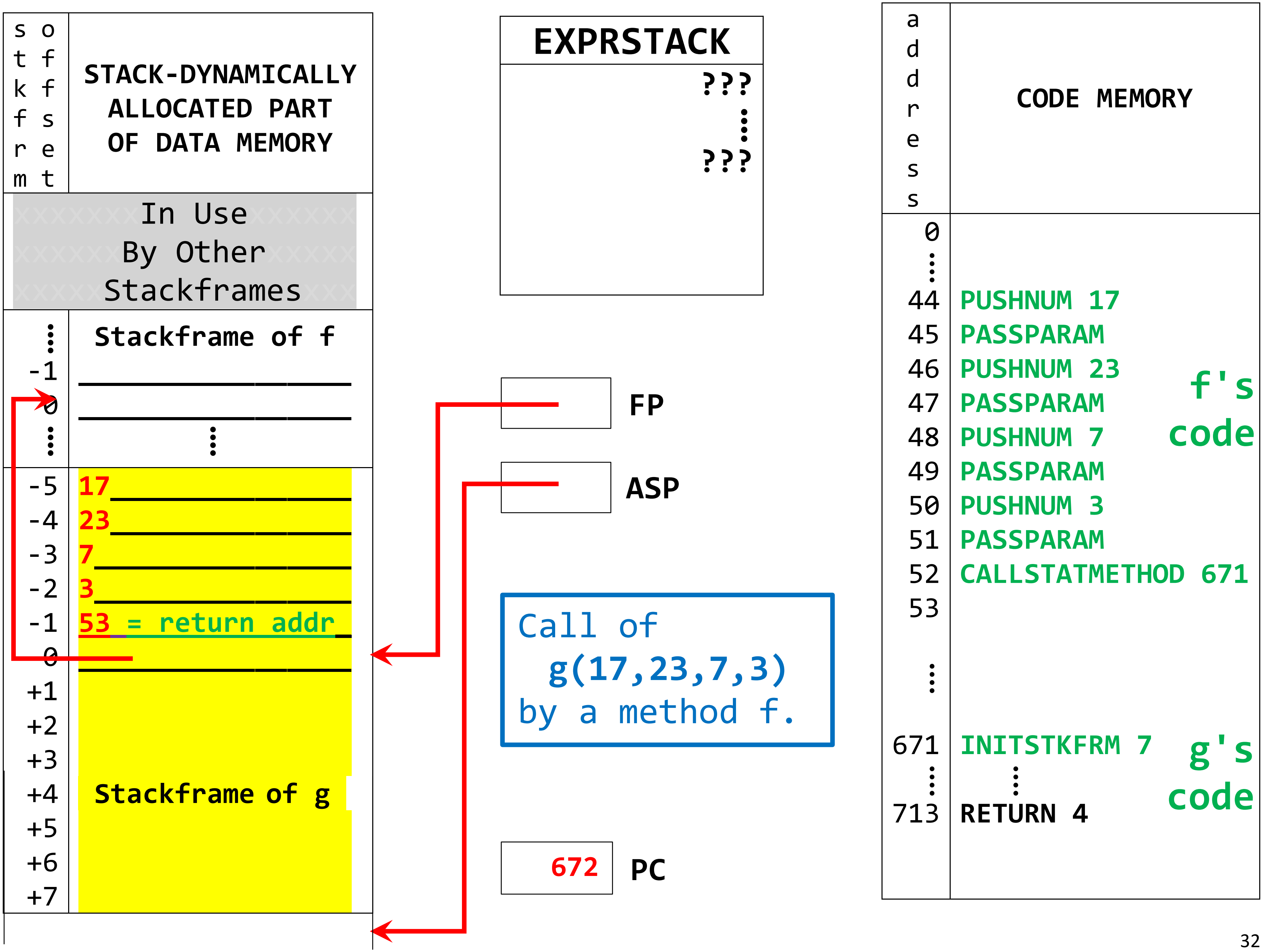
g's code

After Step 2 of Execution of 671: INITSTKFRM 7



Call of g(17,23,7,3) by a method f.

After Final Step of *Execution of* 671: INITSTKFRM 7



stkfrm offset

STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f

stkfrm offset	
⋮	
-1	
0	
⋮	⋮
-5	17
-4	23
-3	7
-2	3
-1	53 = return addr
0	
+1	
+2	
+3	
+4	Stackframe of g
+5	
+6	
+7	

EXPRSTACK

???
⋮
???

FP

ASP

Call of
g(17,23,7,3)
by a method f.

672 PC

address	CODE MEMORY	
0		
⋮		
44	PUSHNUM 17	f's code
45	PASSPARAM	
46	PUSHNUM 23	
47	PASSPARAM	
48	PUSHNUM 7	
49	PASSPARAM	
50	PUSHNUM 3	
51	PASSPARAM	
52	CALLSTATMETHOD 671	
53		
⋮		
671	INITSTKFRM 7	g's code
⋮	⋮	
713	RETURN 4	

713: RETURN 4

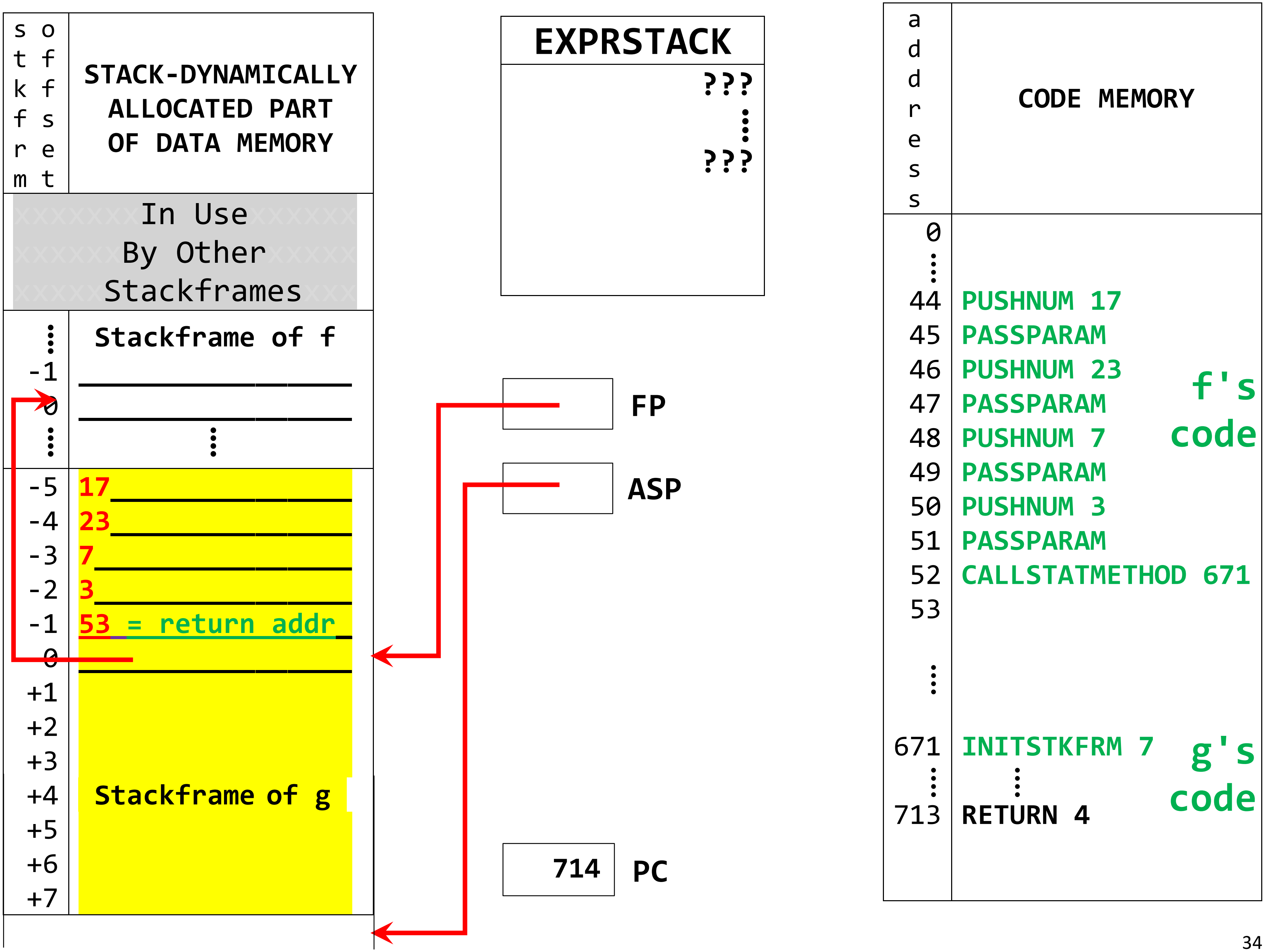
should set ASP to FP+1 [deallocates space used by callee's variables]

and then **S-POP** FP [restores caller’s FP]

and then **S-POP** PC [puts the saved return address into PC]

and then decrease ASP by 4 [deallocates space used by formal parameters]

BEFORE Execution of 713: RETURN 4



stkfrm	offset	STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY
		In Use By Other Stackframes
	⋮	Stackframe of f
	-1	
	0	
	⋮	⋮
	-5	17
	-4	23
	-3	7
	-2	3
	-1	53 = return addr
	0	
	+1	
	+2	
	+3	
	+4	Stackframe of g
	+5	
	+6	
	+7	

EXPRSTACK

???
⋮
???

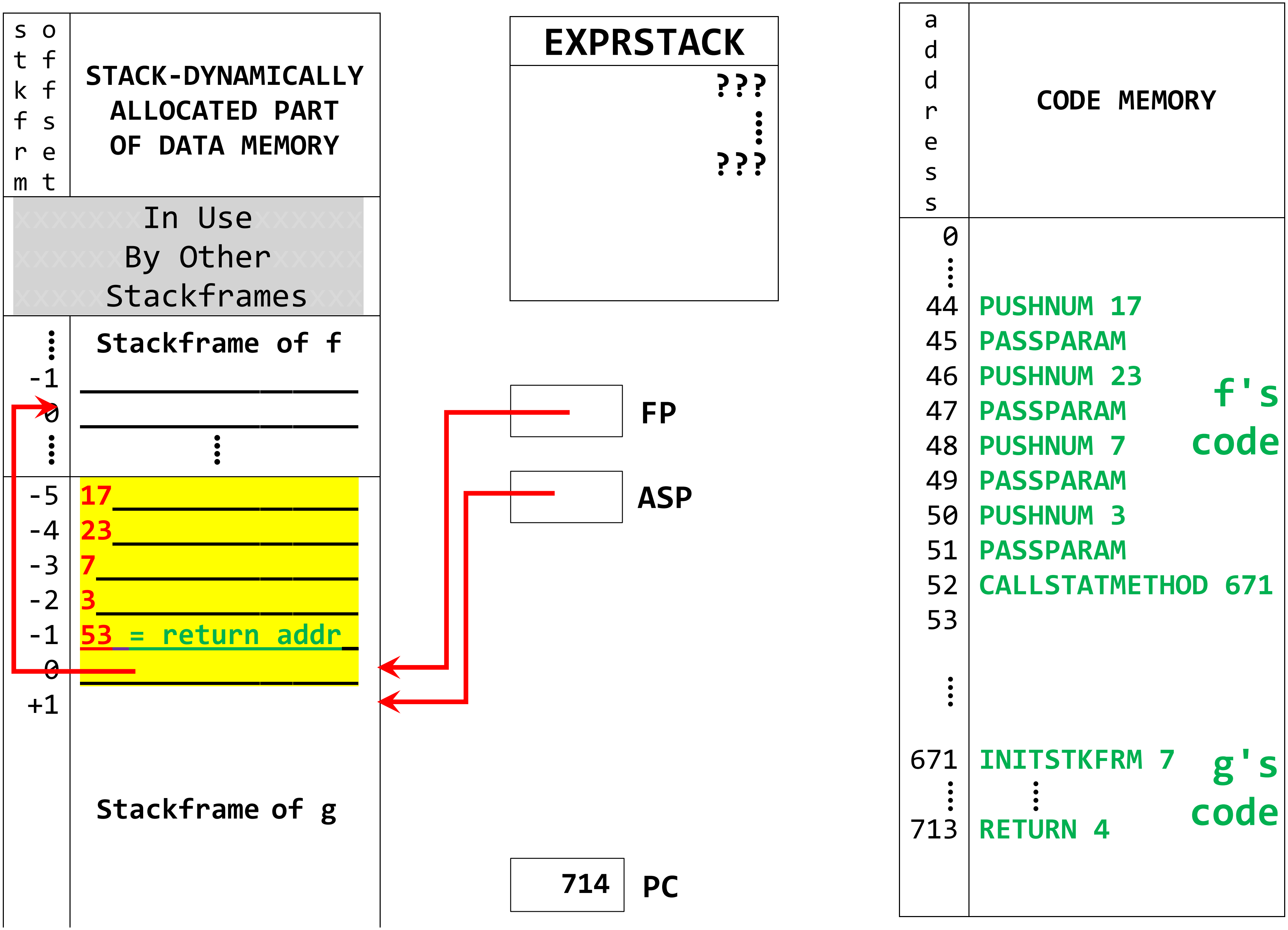
FP

ASP

714 PC

address	CODE MEMORY	
0		
⋮		
44	PUSHNUM 17	
45	PASSPARAM	
46	PUSHNUM 23	f's
47	PASSPARAM	code
48	PUSHNUM 7	
49	PASSPARAM	
50	PUSHNUM 3	
51	PASSPARAM	
52	CALLSTATMETHOD 671	
53		
⋮		
671	INITSTKFRM 7	g's
⋮	⋮	code
713	RETURN 4	

After Step 1 of *Execution of* 713: RETURN 4



STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

stkfrm offset

In Use By Other Stackframes

Stackframe of f

⋮ -1 0 ⋮

-5 17

-4 23

-3 7

-2 3

-1 53 = return addr

0

+1

Stackframe of g

EXPRSTACK

???

⋮

???

FP

ASP

714 PC

CODE MEMORY

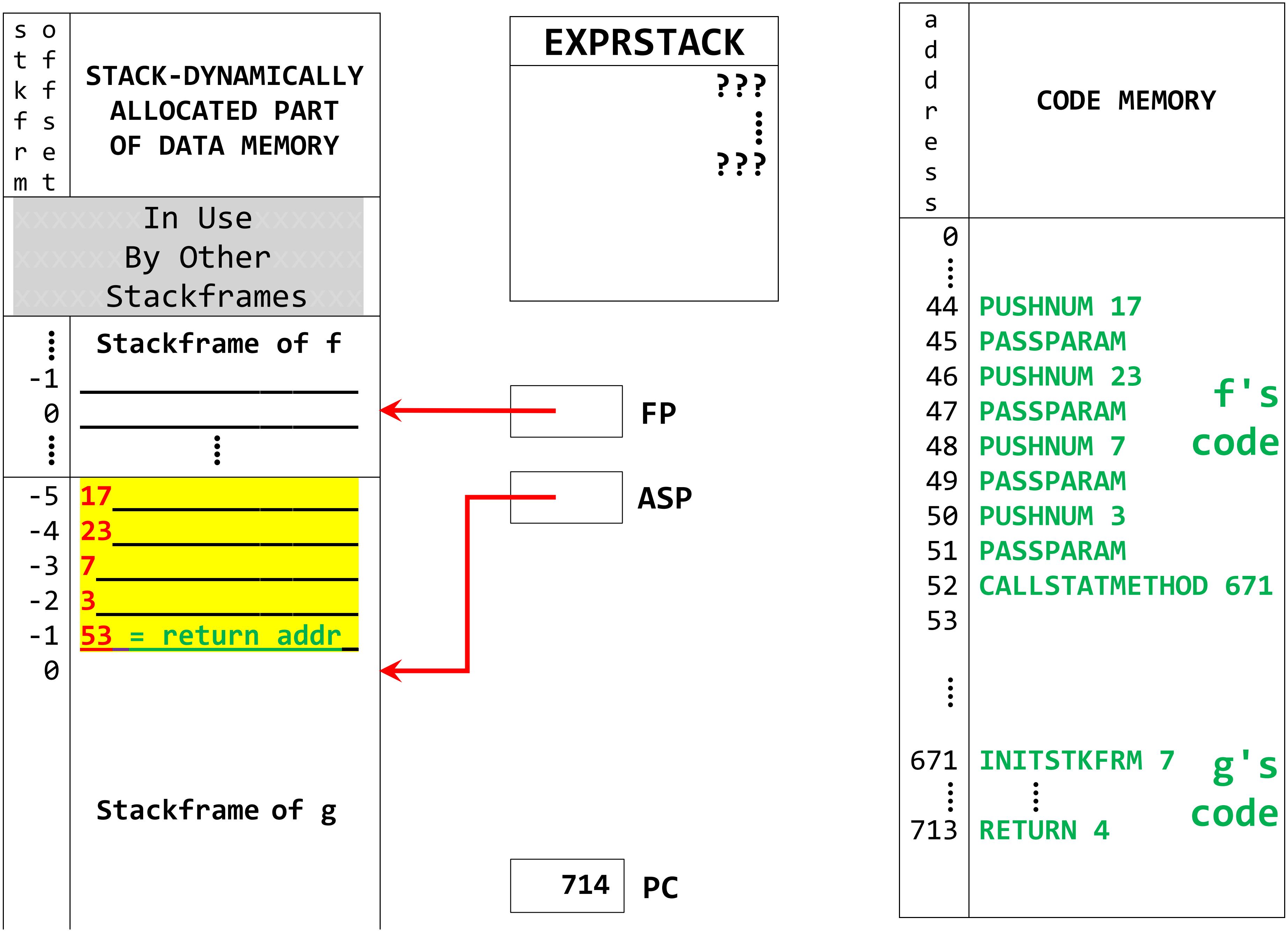
address

```
0
⋮
44   PUSHNUM 17
45   PASSPARAM
46   PUSHNUM 23
47   PASSPARAM
48   PUSHNUM 7
49   PASSPARAM
50   PUSHNUM 3
51   PASSPARAM
52   CALLSTATMETHOD 671
53
⋮
671  INITSTKFRM 7
⋮        ⋮
713  RETURN 4
```

f's code

g's code

After S-POP FP during Execution of 713: RETURN 4



stkfrm offset

STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f

⋮
-1
0
⋮

-5	17
-4	23
-3	7
-2	3
-1	53 = return addr
0	

Stackframe of g

EXPRSTACK

???
⋮
???

FP

ASP

714 PC

address

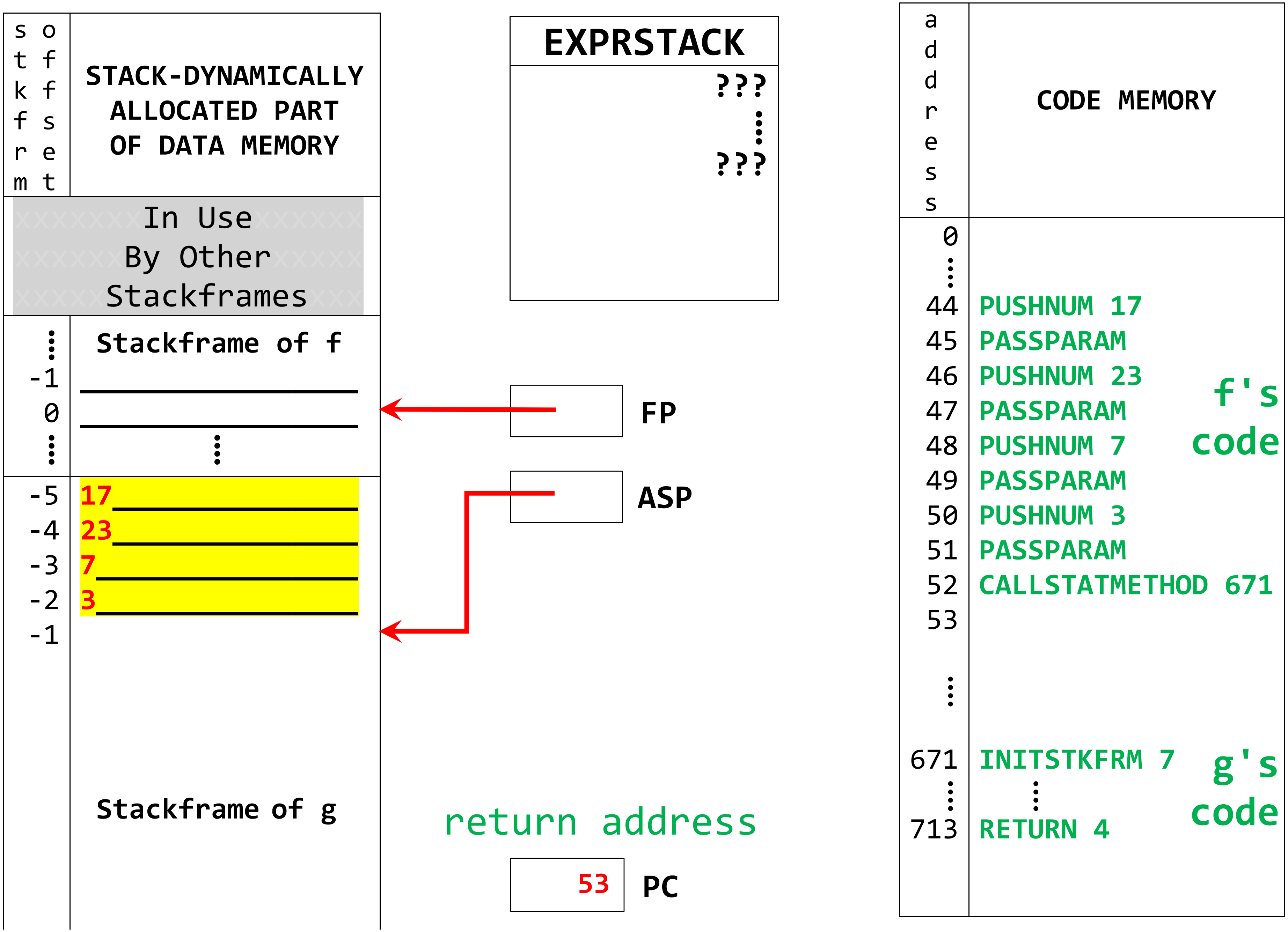
CODE MEMORY

```
  0
  ⋮
 44  PUSHNUM 17
 45  PASSPARAM
 46  PUSHNUM 23
 47  PASSPARAM
 48  PUSHNUM 7
 49  PASSPARAM
 50  PUSHNUM 3
 51  PASSPARAM
 52  CALLSTATMETHOD 671
 53
  ⋮
671  INITSTKFRM 7
  ⋮     ⋮
713  RETURN 4
```

f's code

g's code

After S-POP PC *during of Execution of* 713: RETURN 4



STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

stkfrm	offset	
		In Use By Other Stackframes
	⋮	Stackframe of f
	-1	
	0	
	⋮	⋮
	-5	17
	-4	23
	-3	7
	-2	3
	-1	
		Stackframe of g

EXPRSTACK

???
⋮
???

FP

ASP

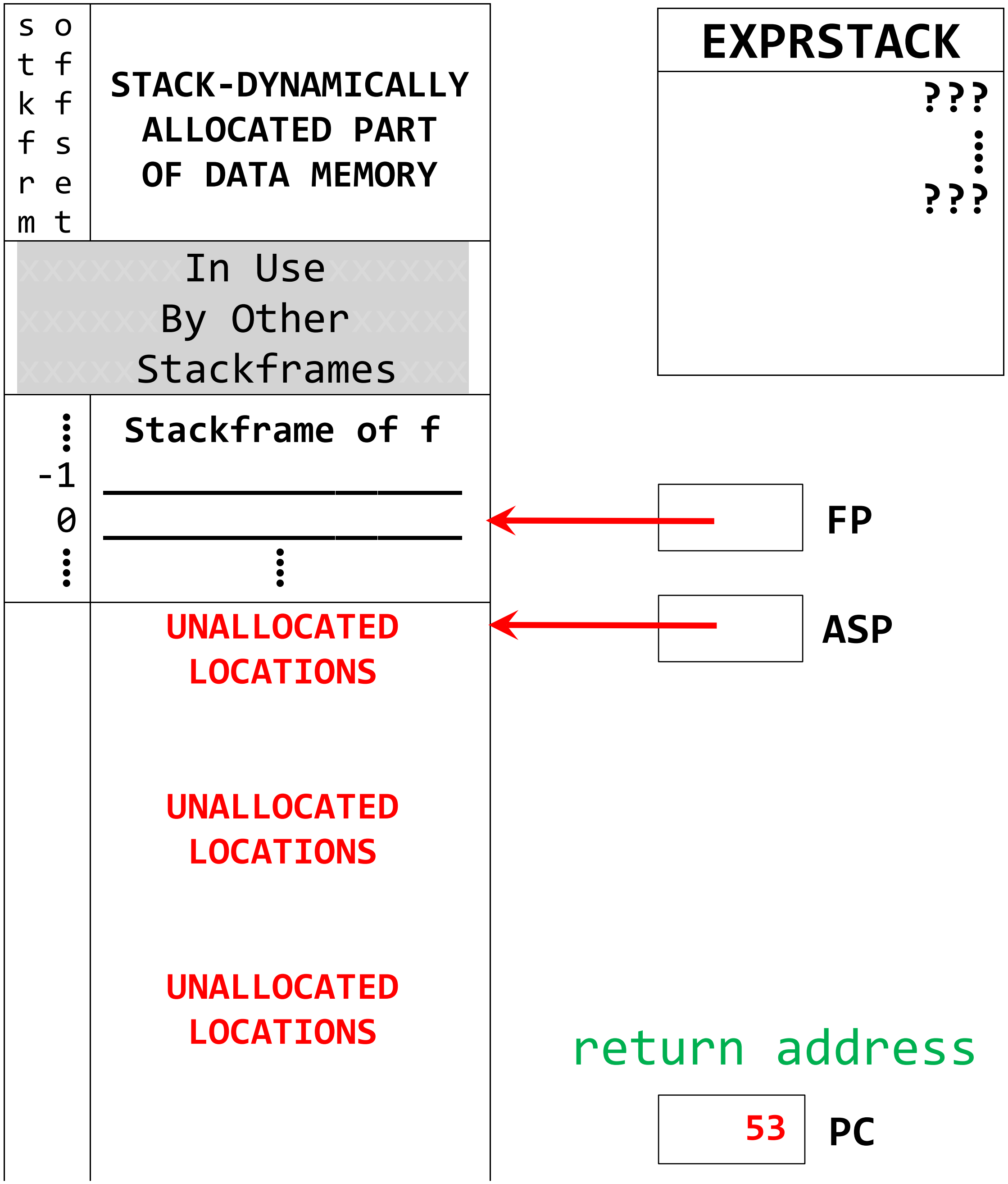
return address

PC: 53

CODE MEMORY

```
address
  0
  ⋮
 44  PUSHNUM 17
 45  PASSPARAM
 46  PUSHNUM 23
 47  PASSPARAM          f's
 48  PUSHNUM 7          code
 49  PASSPARAM
 50  PUSHNUM 3
 51  PASSPARAM
 52  CALLSTATMETHOD 671
 53
  ⋮
671  INITSTKFRM 7       g's
  ⋮      ⋮              code
713  RETURN 4
```

After Final Step of *Execution of* 713: RETURN 4



stkfrm offset

STACK-DYNAMICALLY ALLOCATED PART OF DATA MEMORY

In Use By Other Stackframes

Stackframe of f

⋮
-1
0
⋮

UNALLOCATED LOCATIONS

UNALLOCATED LOCATIONS

UNALLOCATED LOCATIONS

EXPRSTACK

???
⋮
???

FP

ASP

return address

53 PC

address	CODE MEMORY	
0		
⋮		
44	PUSHNUM 17	f's code
45	PASSPARAM	
46	PUSHNUM 23	
47	PASSPARAM	
48	PUSHNUM 7	
49	PASSPARAM	
50	PUSHNUM 3	
51	PASSPARAM	
52	CALLSTATMETHOD 671	
53		
⋮		
671	INITSTKFRM 7	g's code
⋮	⋮	
713	RETURN 4	