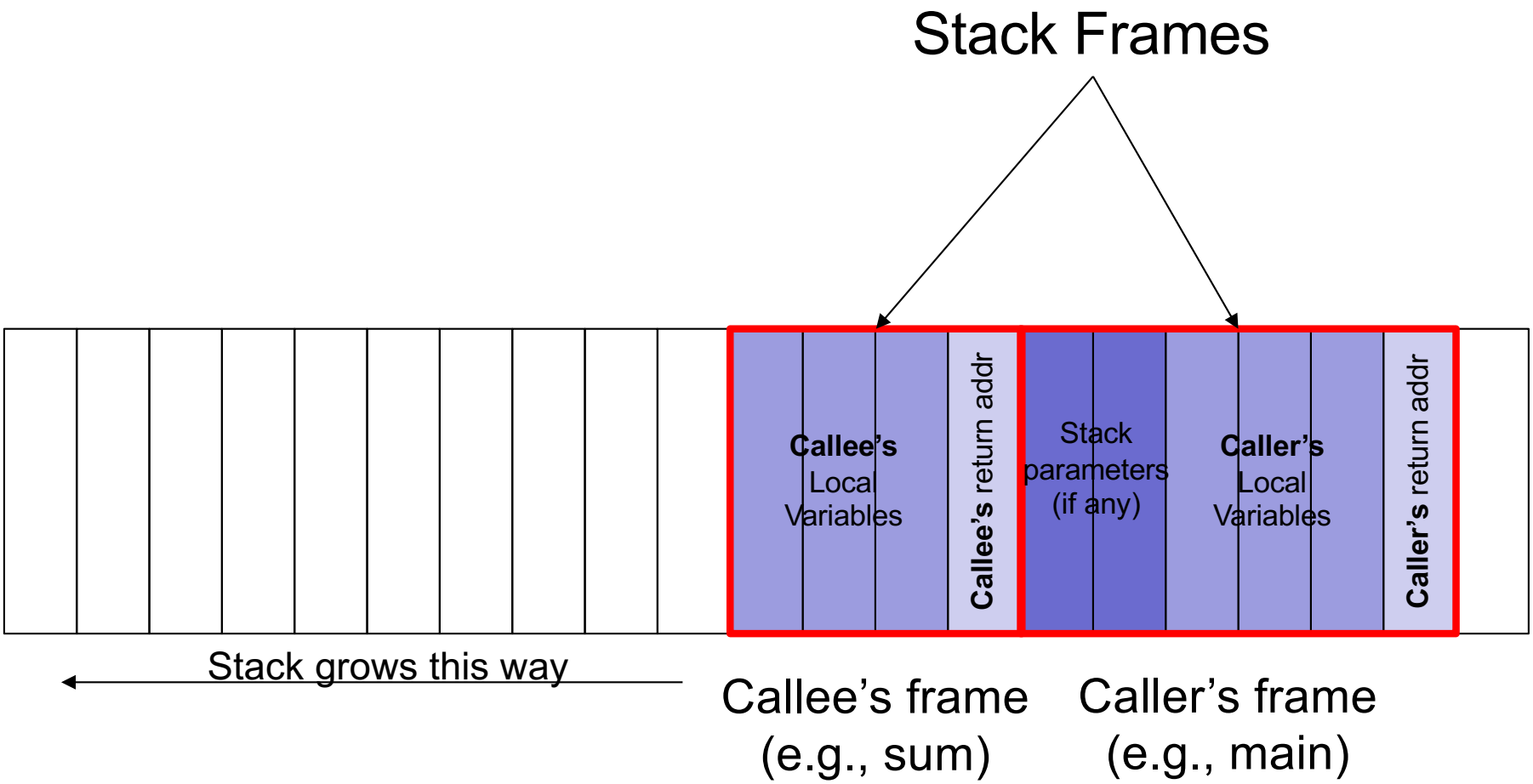
Y86 Stack Frames

- Topics
  - How do procedures organize data such as local variables, return addresses, etc.?
- Learning Objectives
  - Draw a stack frame illustrating how stack frames are established after a function call.
  - Implement construction and teardown of a call frame.

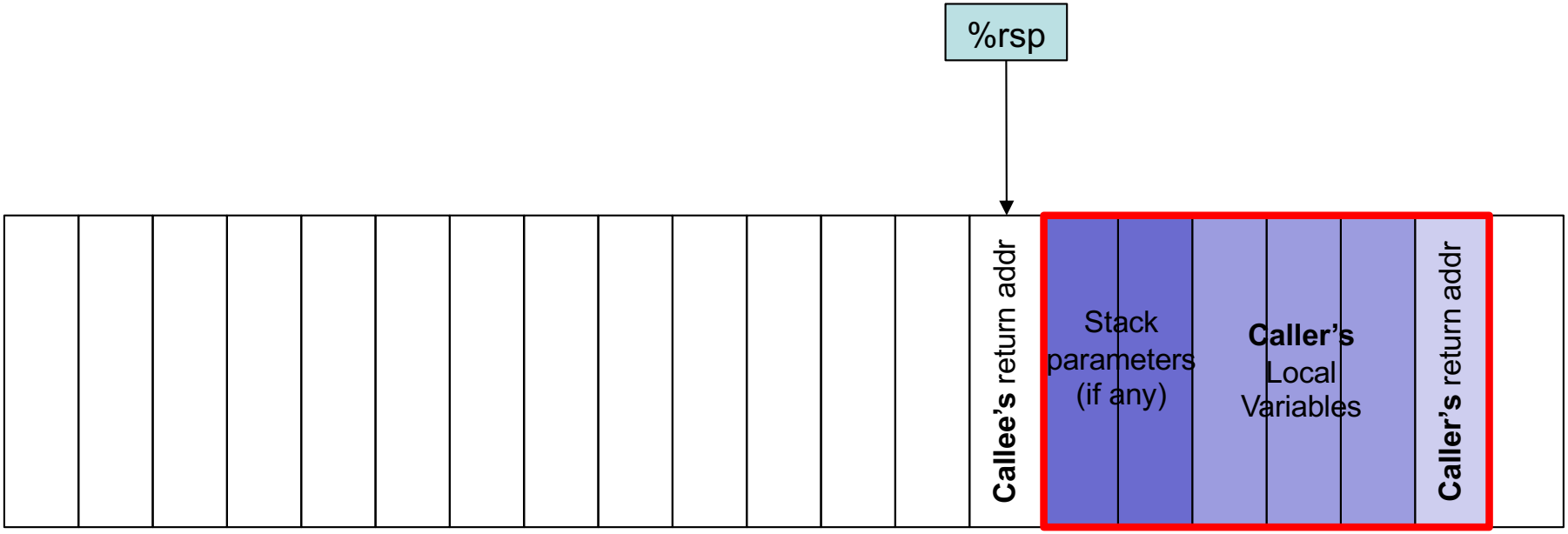
Using the Stack

- CALL: Uses the stack to store a return address
- We also use the stack to:
  - Transmit parameters (if we cannot use registers)
  - Store local variables
- The structure we use to store this information is called a **stack frame.**

Stack Frames

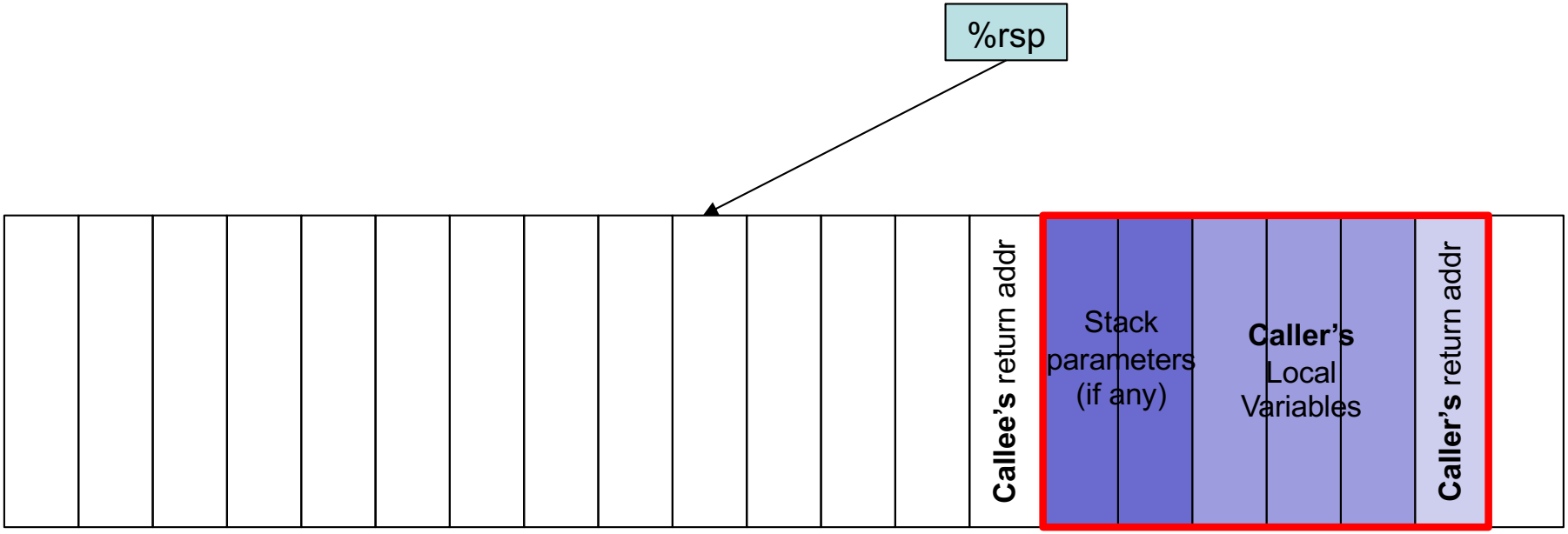


Stack Frames: no Base Pointer (1)



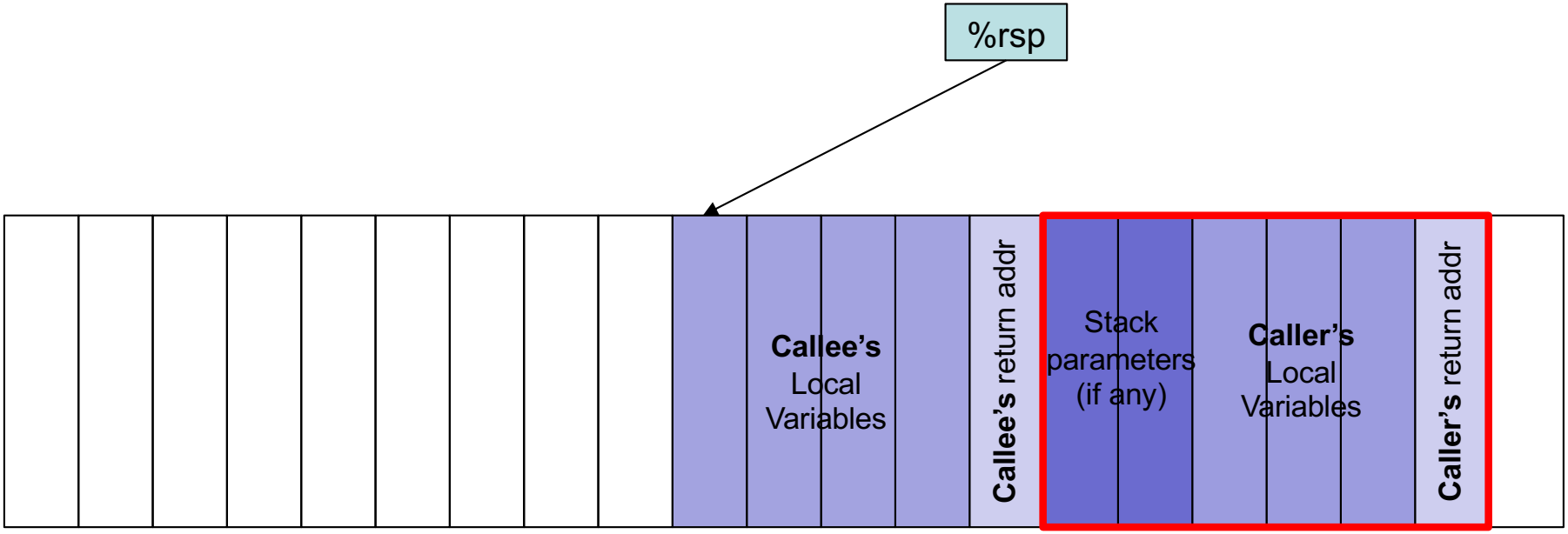
Stack Frame Setup: no Base Pointer (2)

```
irmovq    0x20, %rax
subq      %rax, %rsp
```



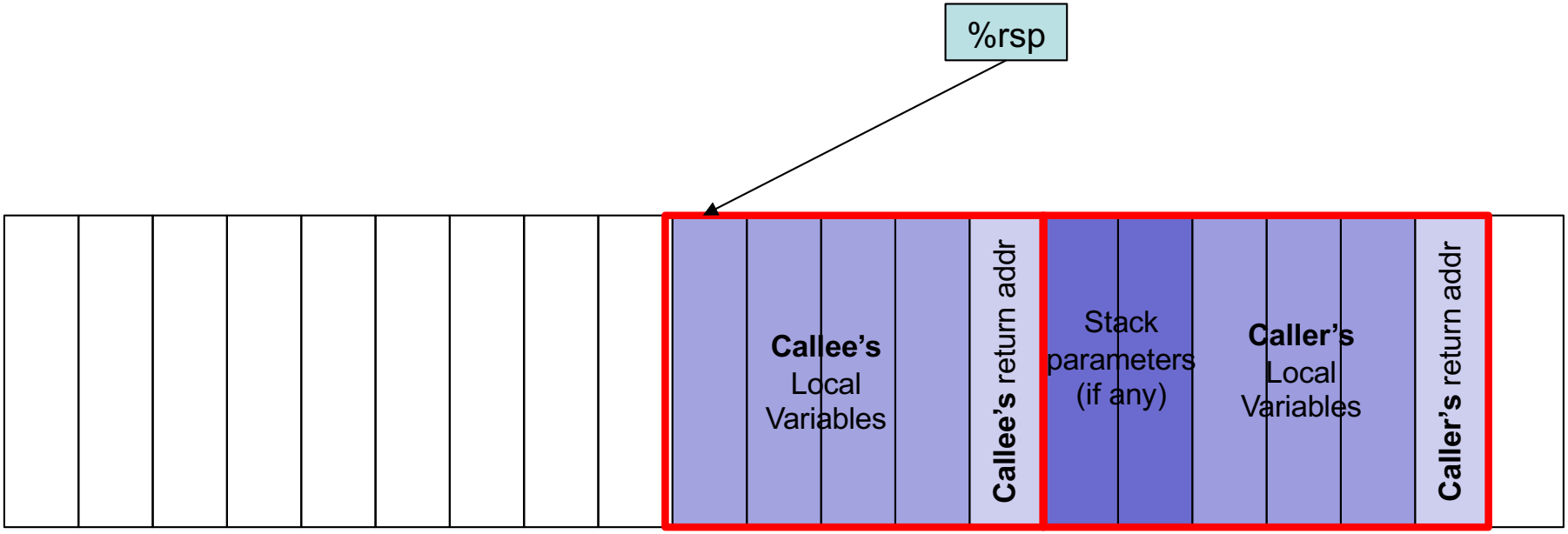
Stack Frame Setup: no Base Pointer (3)

```
irmovq    0x20, %rax
subq      %rax, %rsp
```



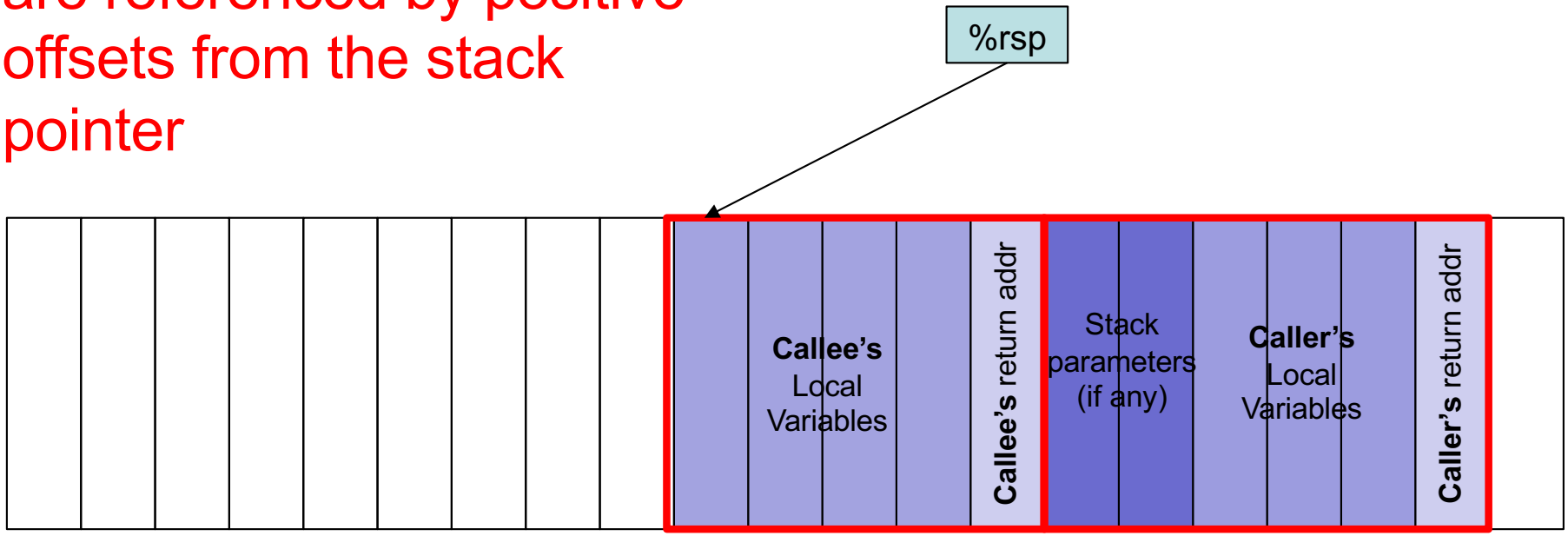
Stack Frame Setup: no Base Pointer (4)

```
irmovq    0x20, %rax
subq      %rax, %rsp
```



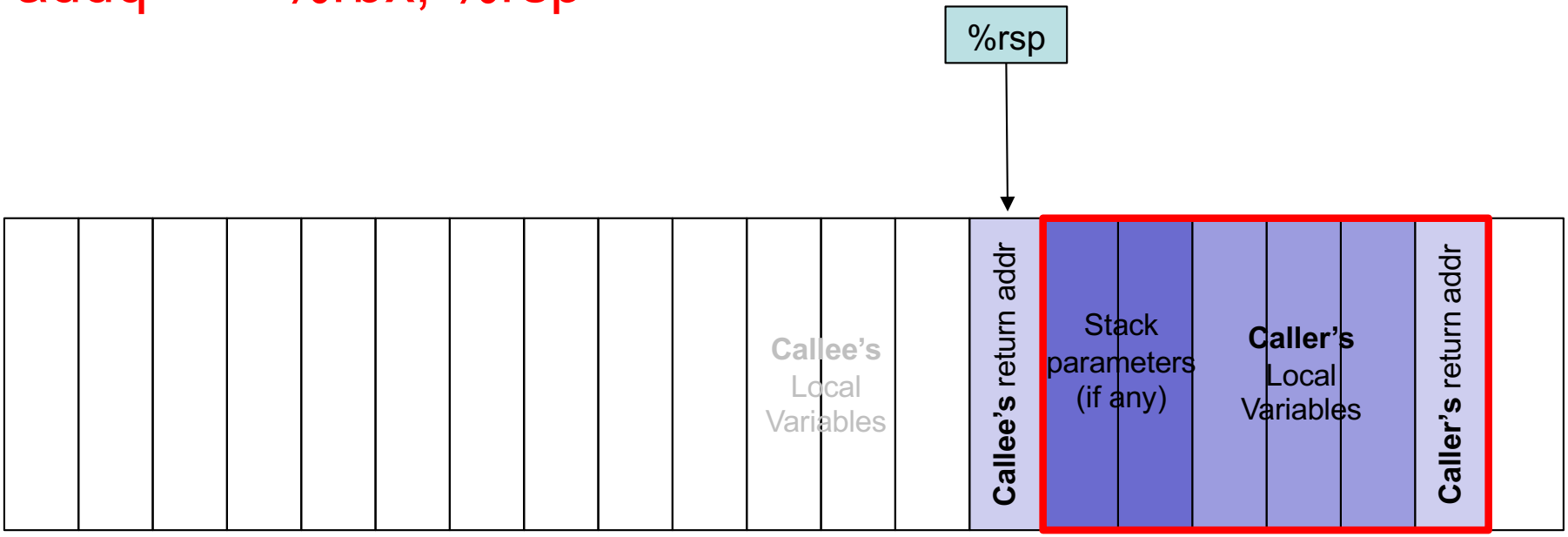
Stack Frame Setup: no Base Pointer (5)

Both local variables and stack parameters (if any) are referenced by positive offsets from the stack pointer



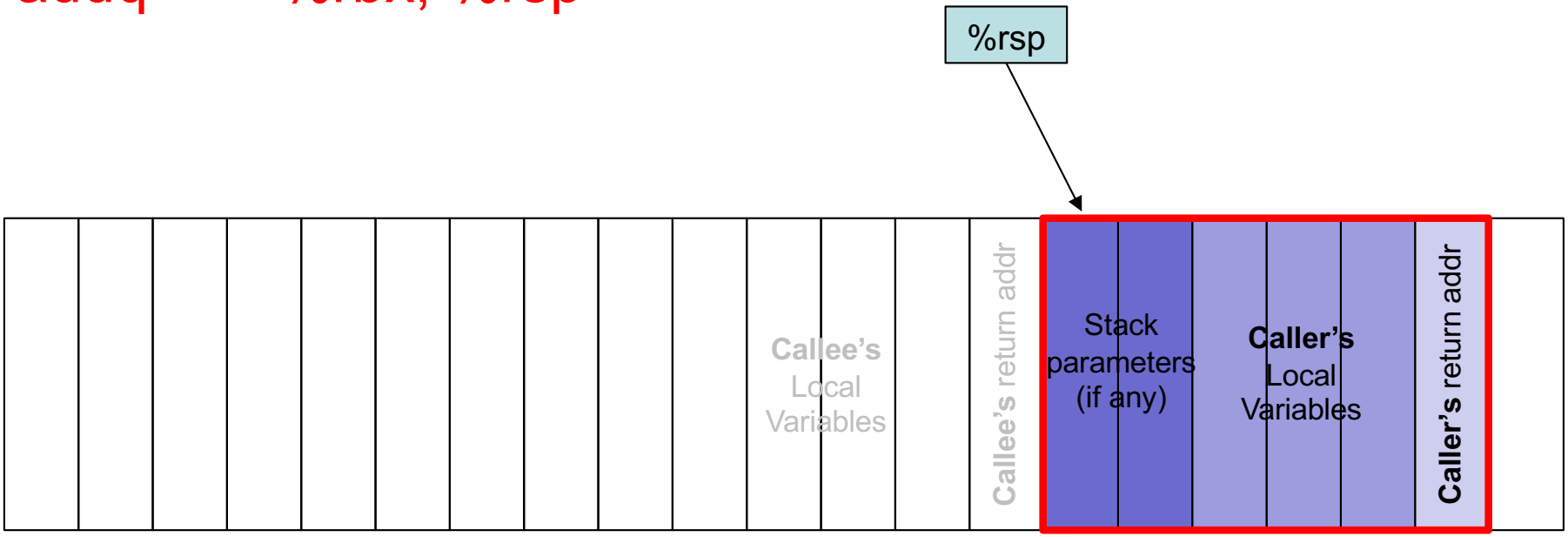
Stack Frame Teardown: no Base Pointer (6)

```
irmovq    0x20, %rbx
addq      %rbx, %rsp
```



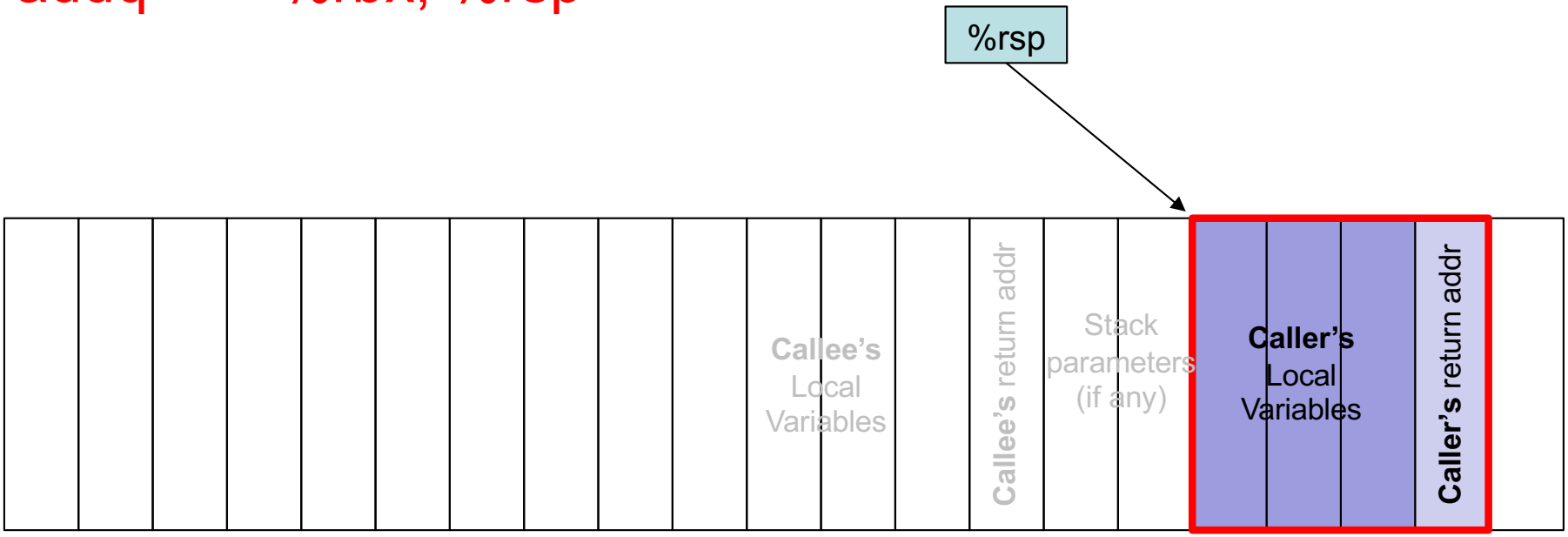
Stack Frame Teardown: no Base Pointer (7)

```
irmovq    0x20, %rbx
addq      %rbx, %rsp
```

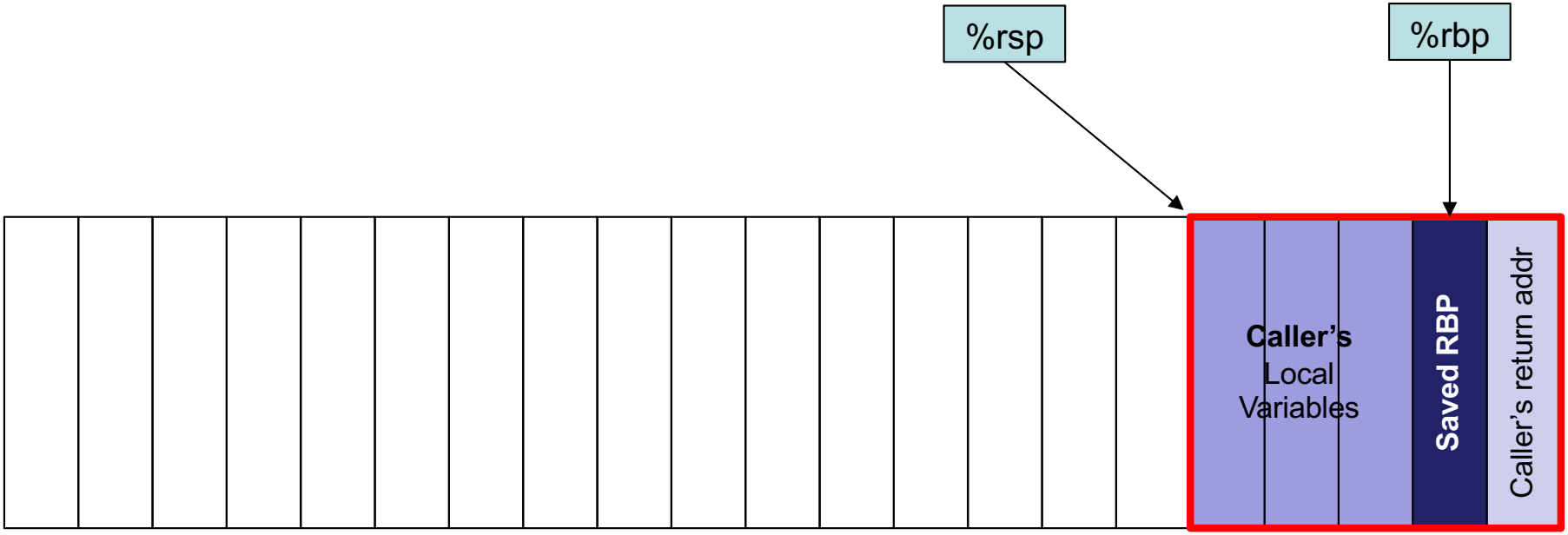


Stack Frame: no Base Pointer (8)

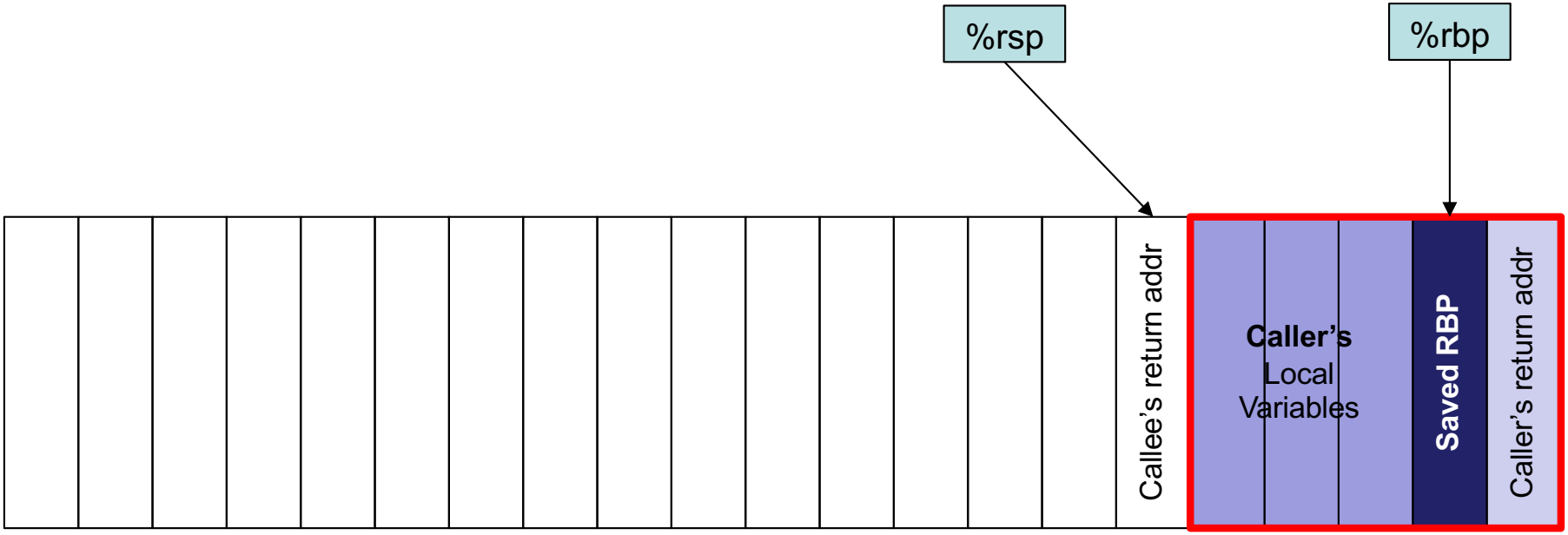
```
irmovq    0x10, %rbx
addq      %rbx, %rsp
```



Stack Frames: with Base Pointer (1)

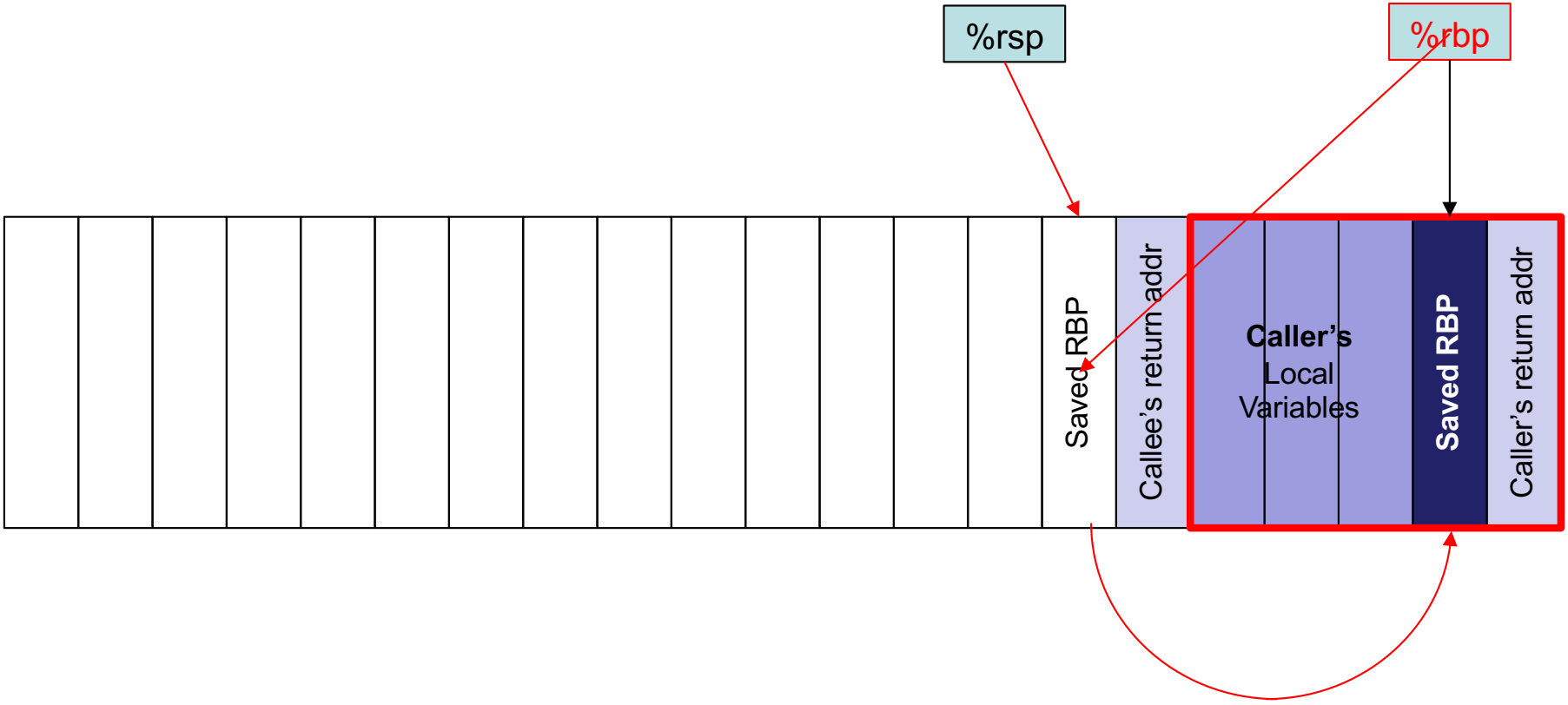


Stack Frame Setup : with Base Pointer (2)



Stack Frame Setup : with Base Pointer (3)

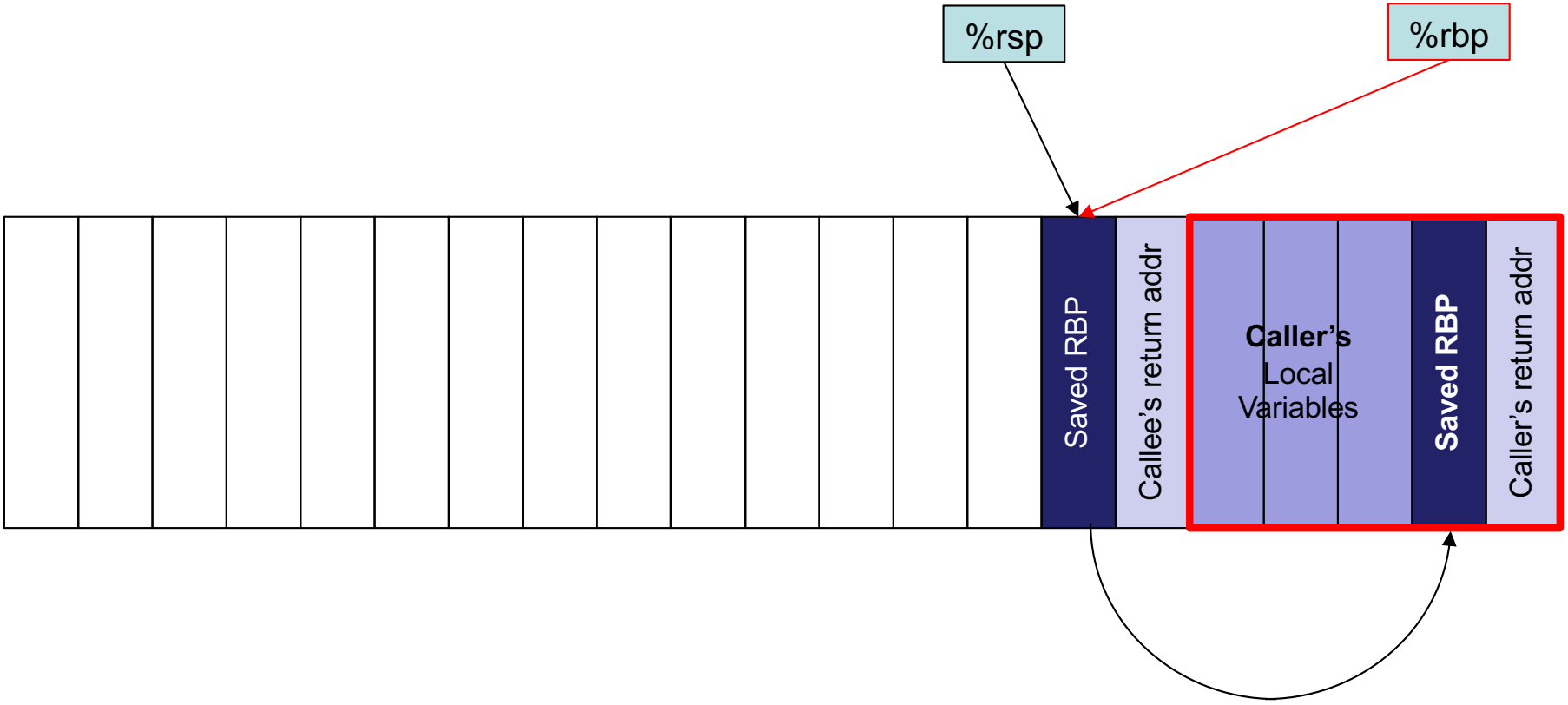
pushq %rbp



Stack Frame Setup: with Base Pointer (4)

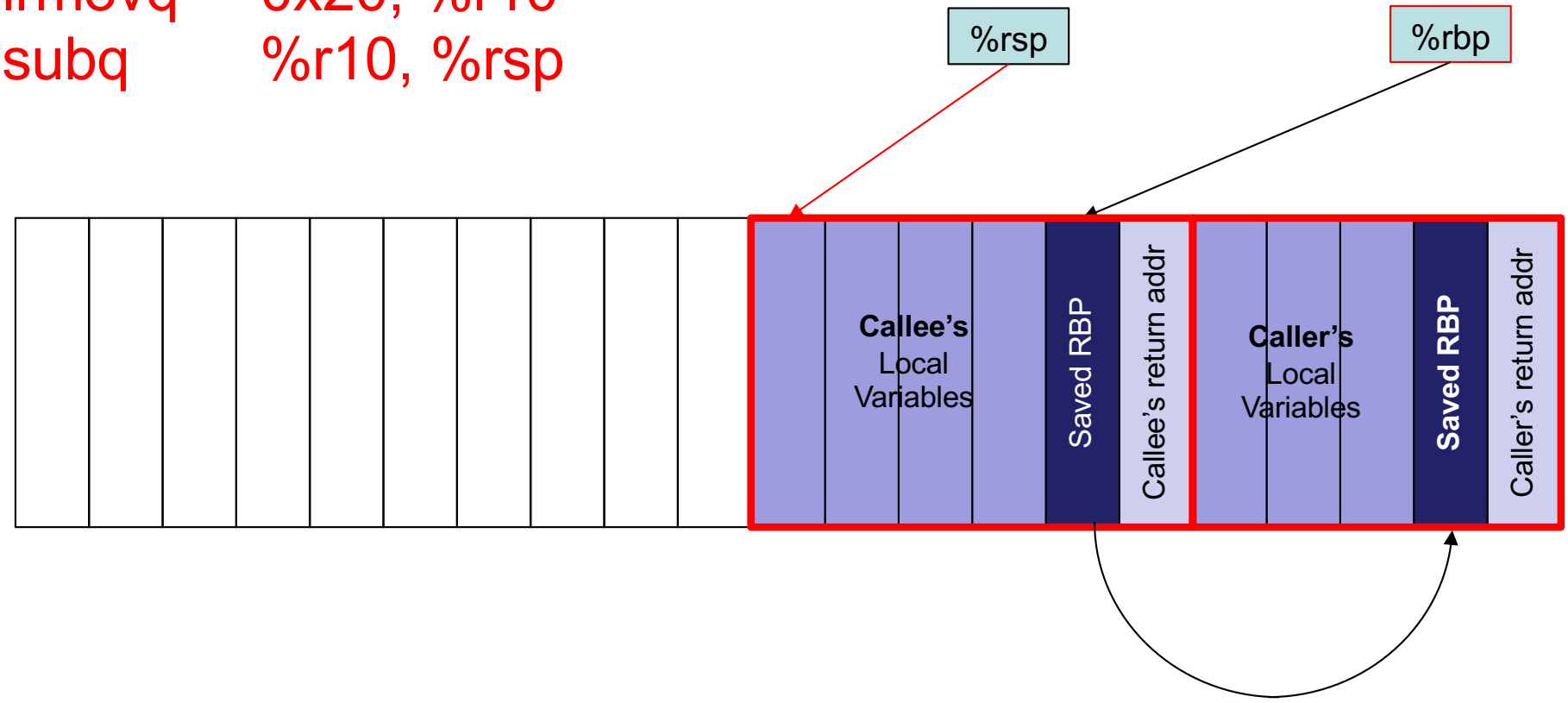
pushq %rbp

rrmovq %rsp, %rbp



Stack Frame Setup: with Base Pointer (5)

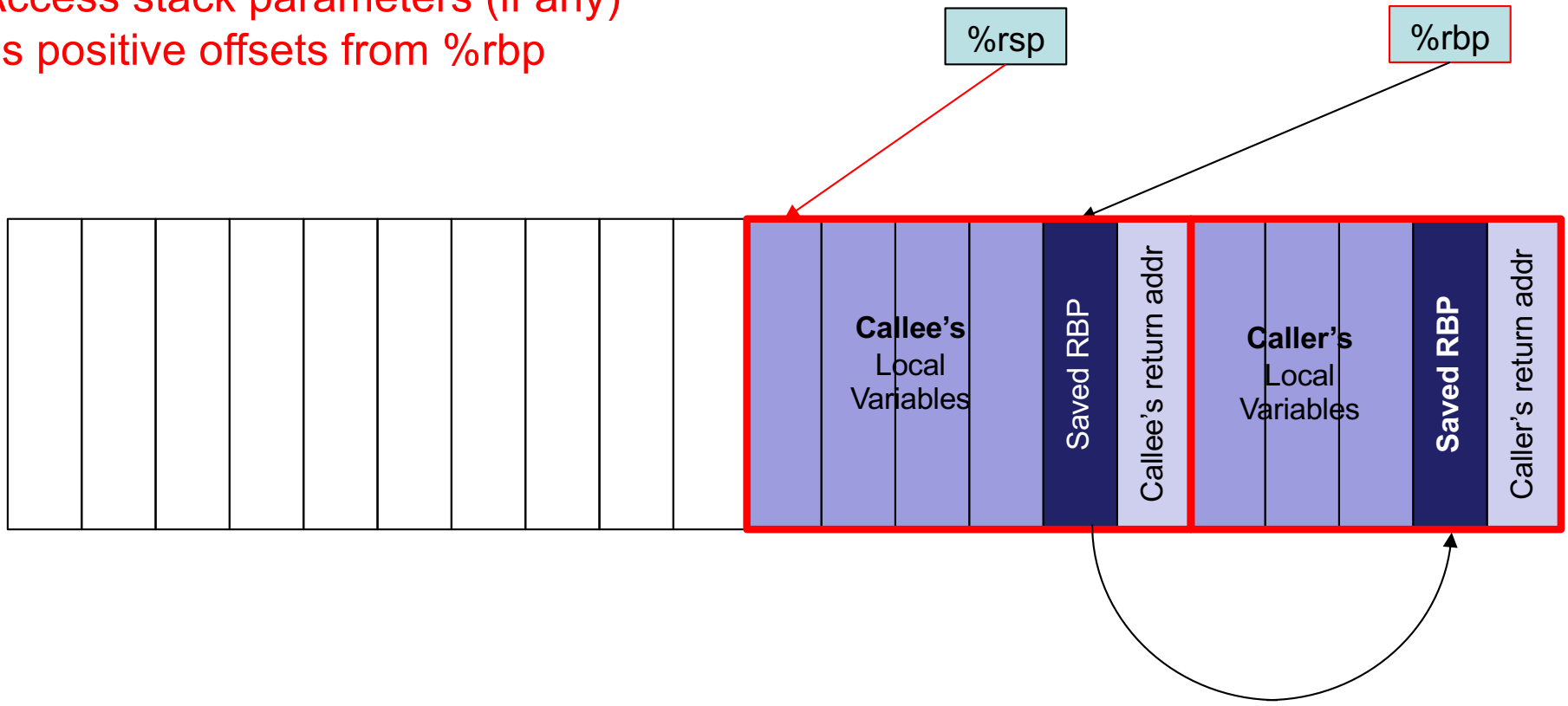
```
pushq     %rbp
rrmovq    %rsp, %rbp
irmovq    0x20, %r10
subq      %r10, %rsp
```



Stack Frames: with Base Pointer (6)

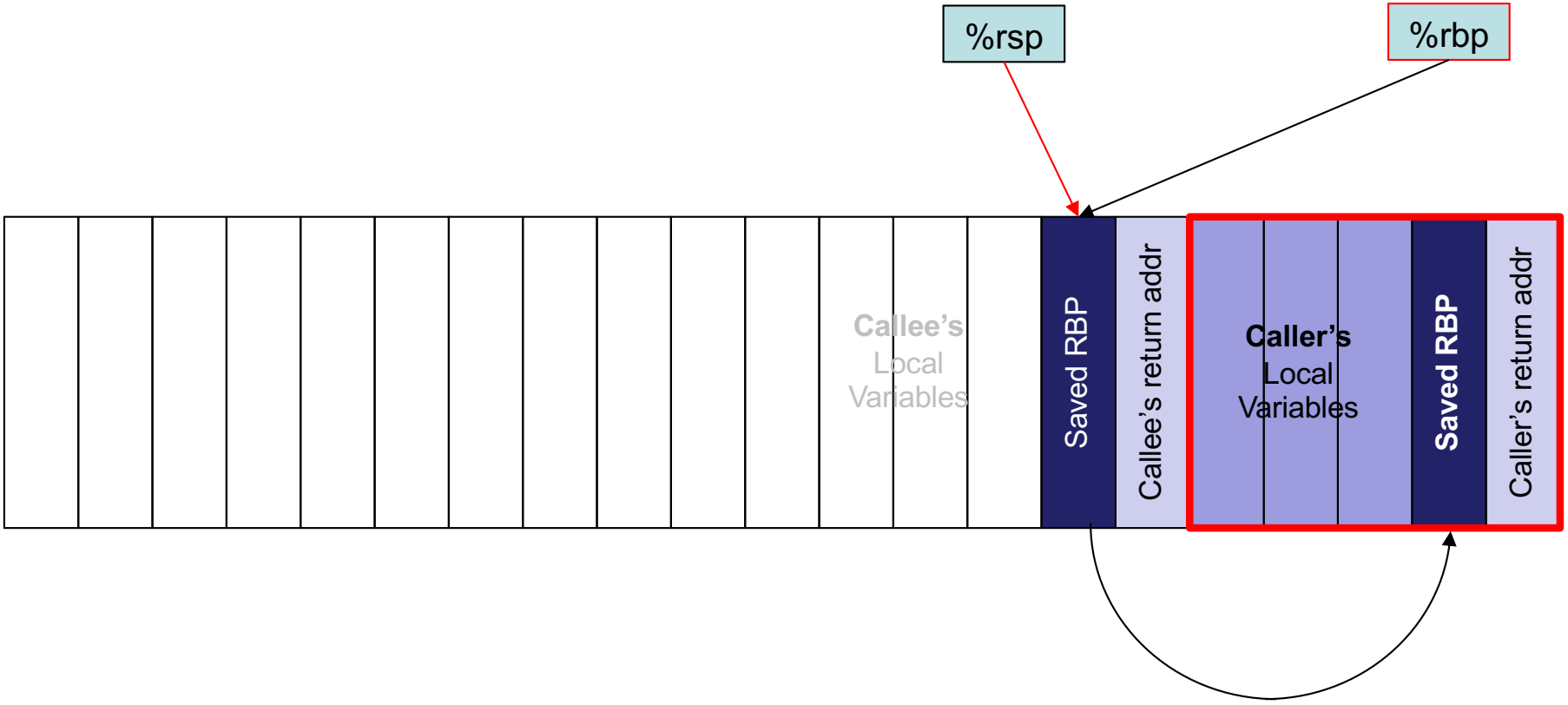
Access locals as negative offsets from %rbp

Access stack parameters (if any) as positive offsets from %rbp



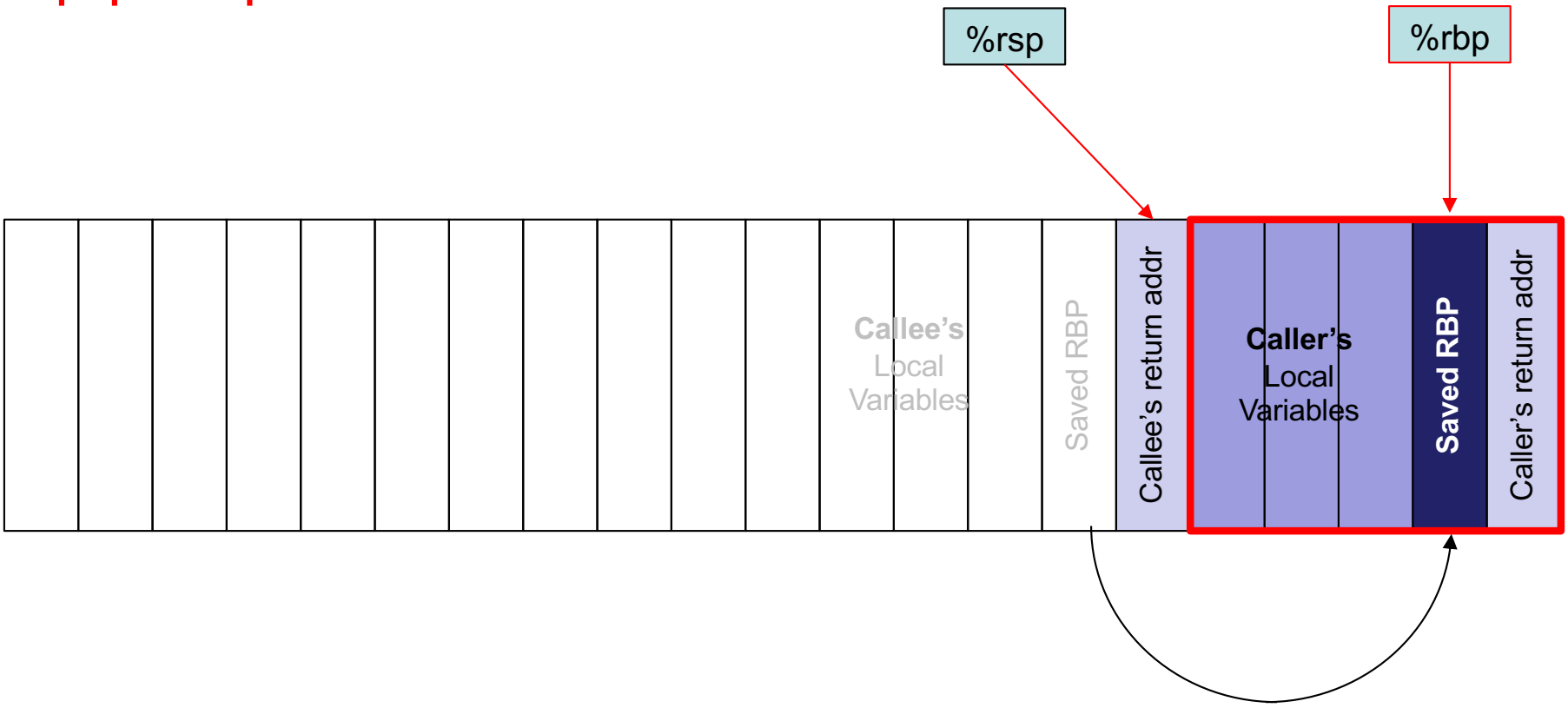
Stack Frame Teardown: with Base Pointer (7)

irmovq 0x20, %r10
addq %r10, %rsp



Stack Frame Teardown: with Base Pointer (8)

```
irmovq    0x20, %r10
addq   %r10, %rsp
popq %rbp
```



Stack Frame Teardown: with Base Pointer (9)

```
irmovq    0x20, %r10
addq   %r10, %rsp
popq %rbp
ret
```