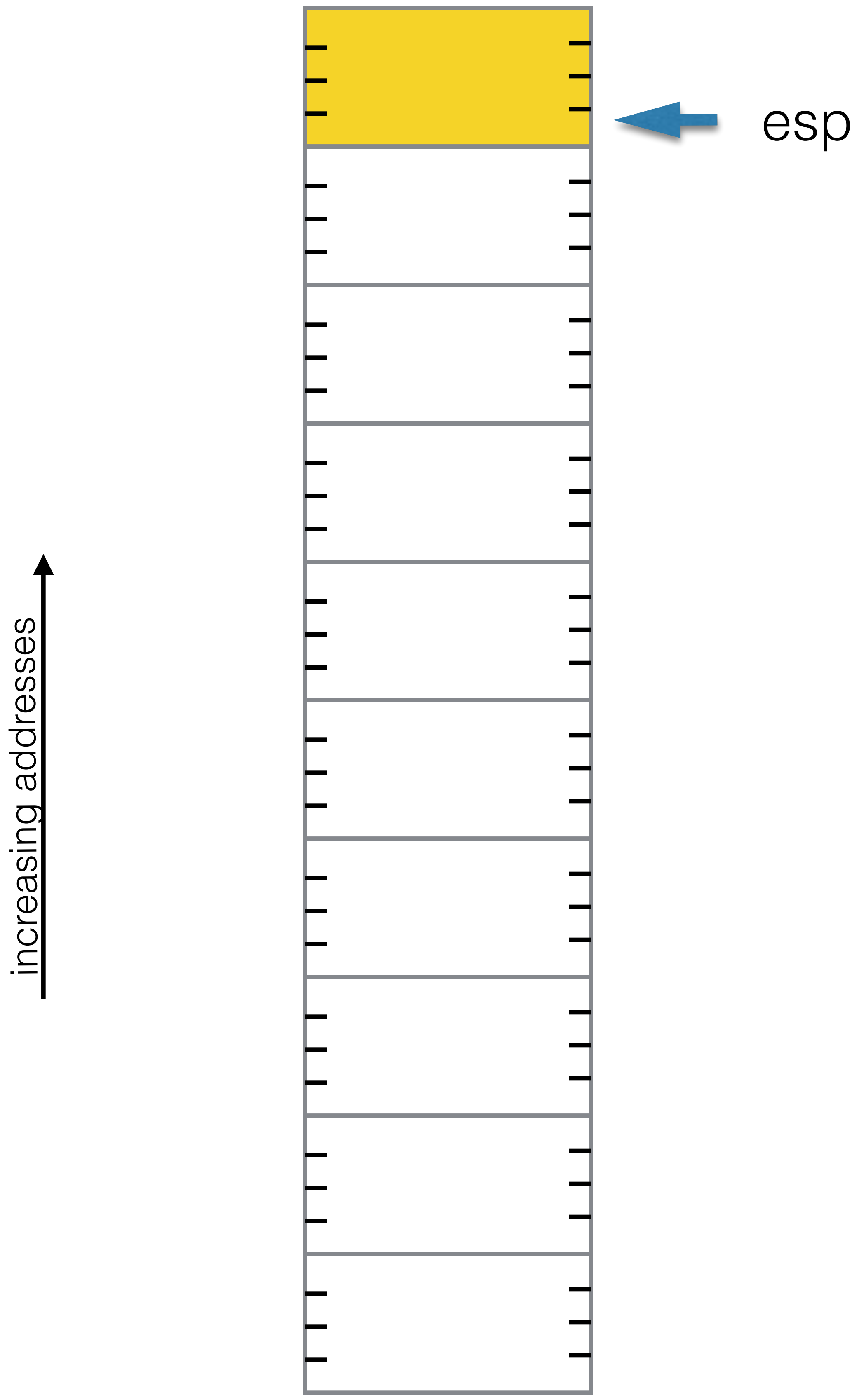
Passing Parameters Through the Stack

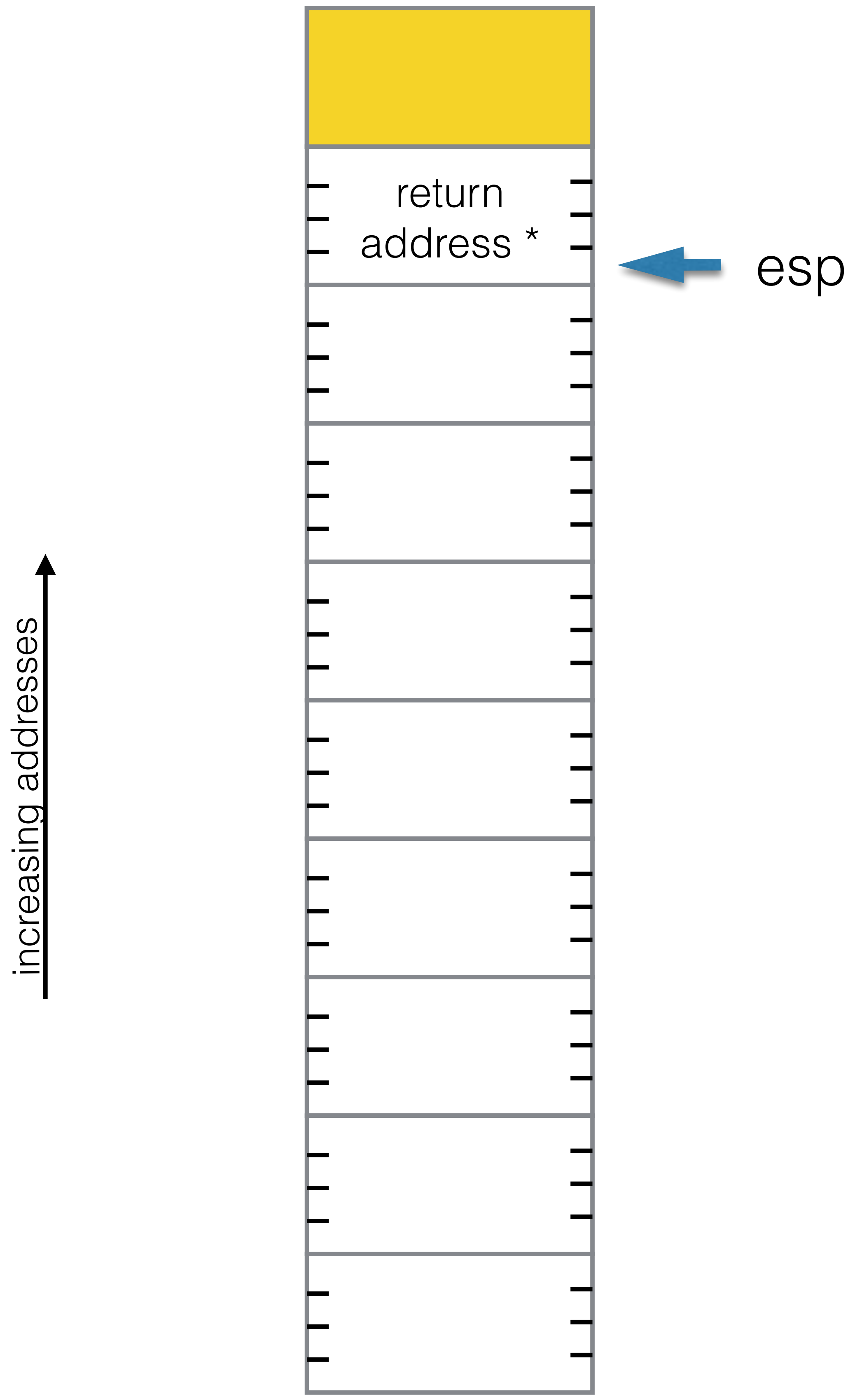
D. Thiebaut — CSC231

Pass *a* & *b* via Registers

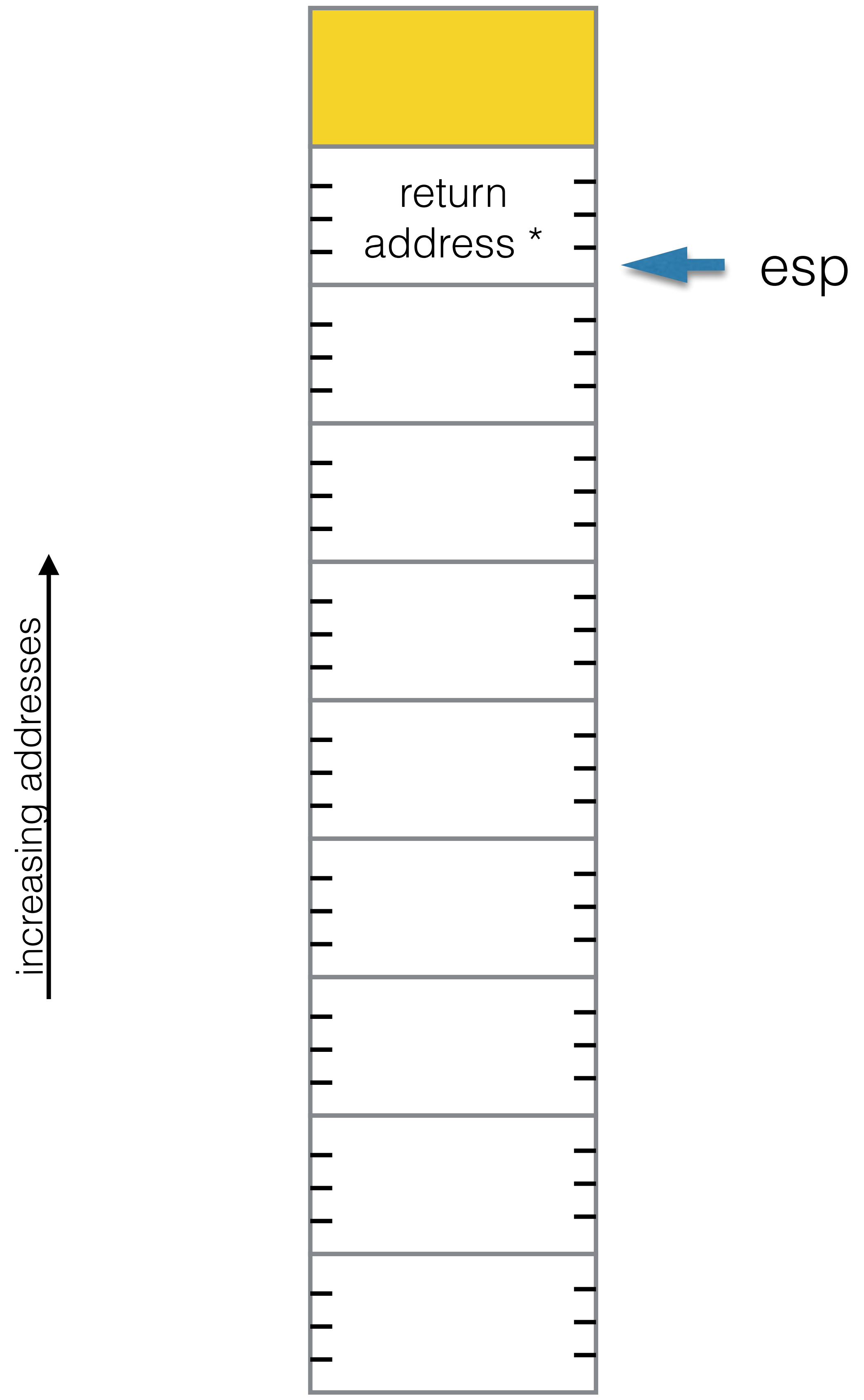
```
        section .data
a       dd      1234
b       dd      5555
result  dd      0

        section .text

        mov     eax,dword [a]
        mov     ebx,dword [b]
        call    sum
*       mov     dword [result], eax


        mov     eax,SYS_EXIT
        mov     ebx,0
        int     0x80

;;; ----------------------------
;;; sum function
;;; adds eax+ebx and return in eax
;;; registers modified:  ax
;;; ----------------------------
sum:    add     eax,ebx
        ret
```



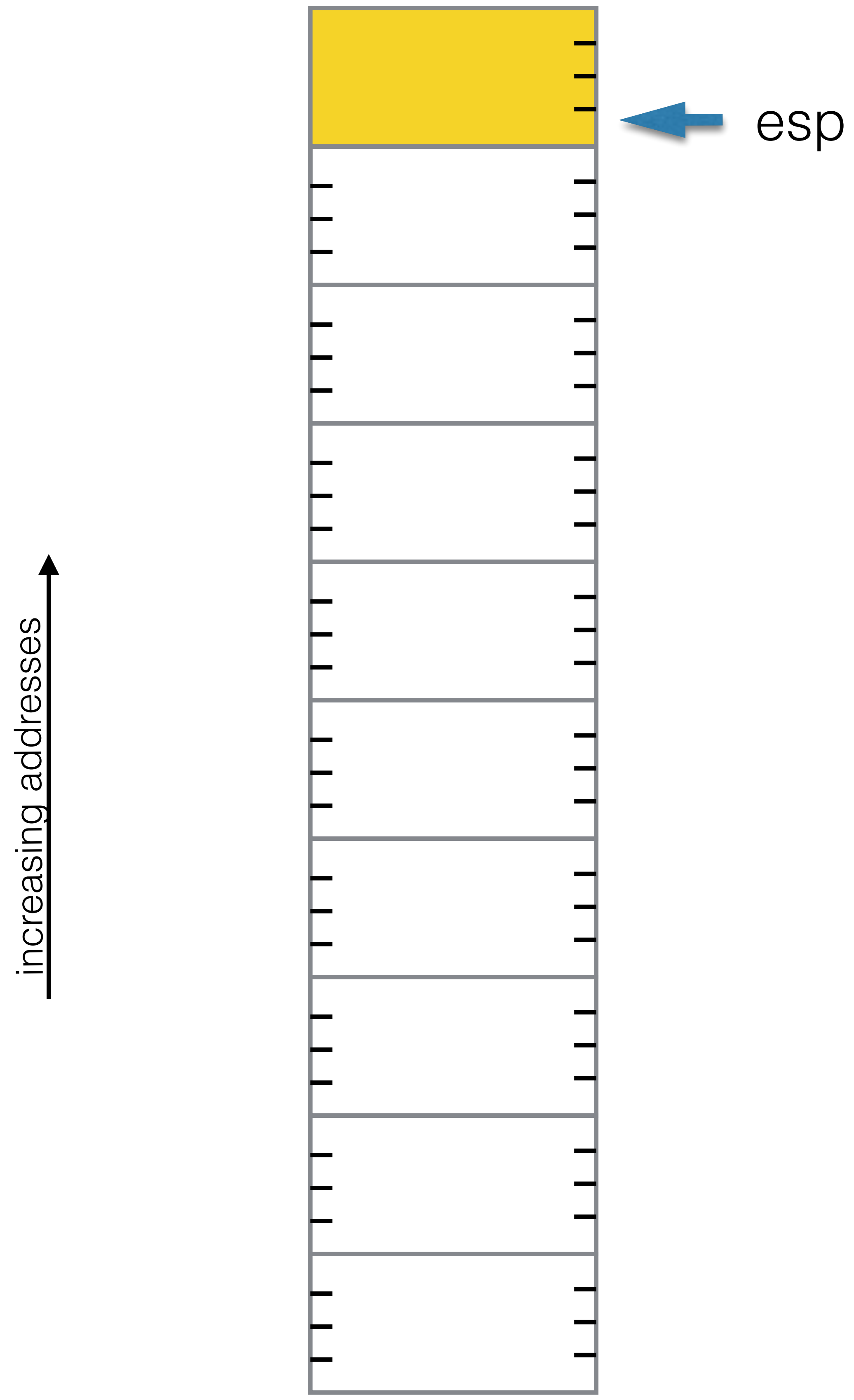
```
        section .data
a       dd      1234
b       dd      5555
result  dd      0

        section .text

        mov     eax,dword [a]
        mov     ebx,dword [b]
        call    sum
*       mov     dword [result], eax


        mov     eax,SYS_EXIT
        mov     ebx,0
        int     0x80

;;; ----------------------------
;;; sum function
;;; adds eax+ebx and return in eax
;;; registers modified:  ax
;;; ----------------------------
sum:    add     eax,ebx
        ret
```



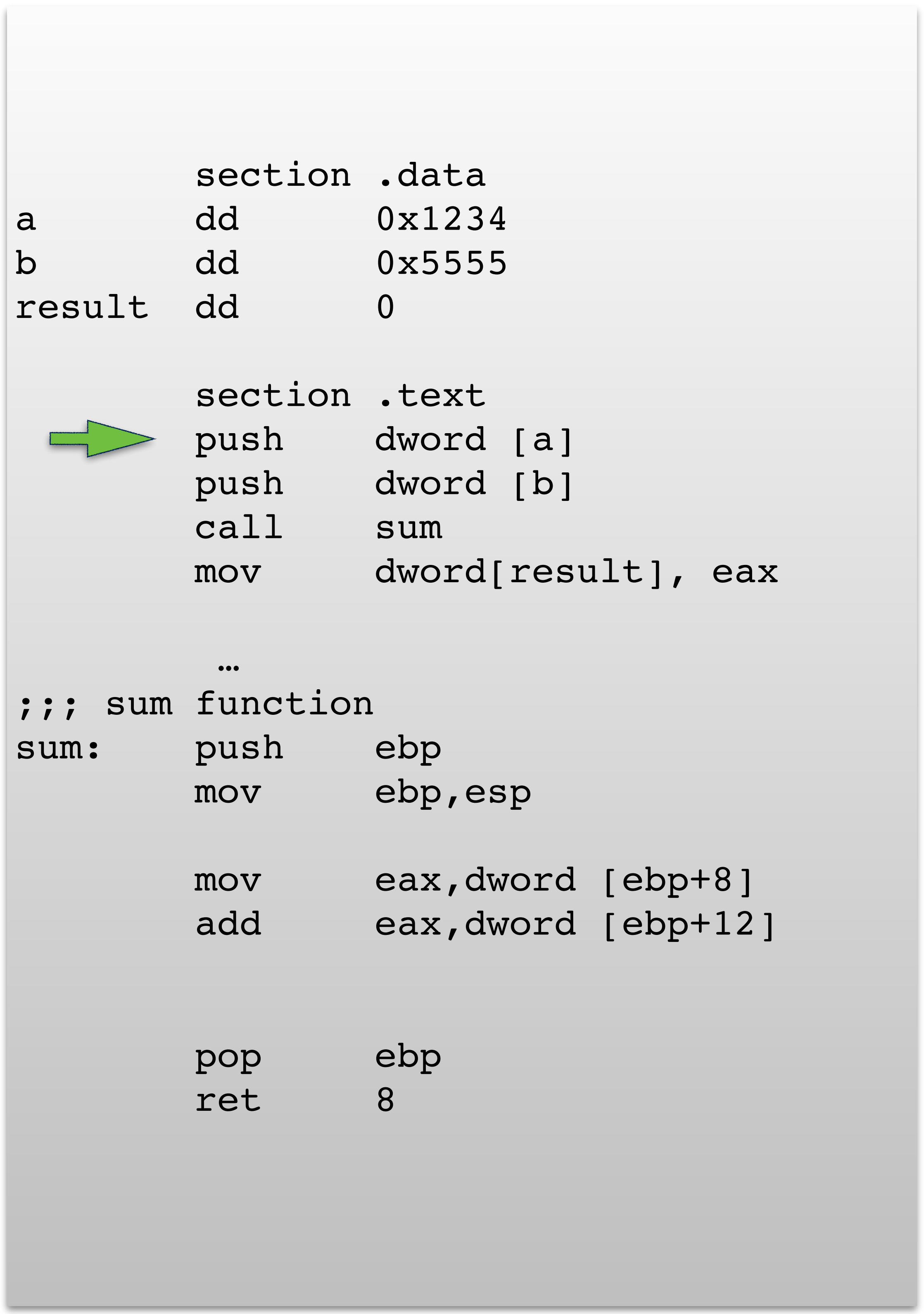
```
        section .data
a       dd      1234
b       dd      5555
result  dd      0

        section .text

        mov     eax,dword [a]
        mov     ebx,dword [b]
        call    sum
*       mov     dword [result], eax


        mov     eax,SYS_EXIT
        mov     ebx,0
        int     0x80

;;; ----------------------------
;;; sum function
;;; adds eax+ebx and return in eax
;;; registers modified:  ax
;;; ----------------------------
sum:    add     eax,ebx
        ret
```



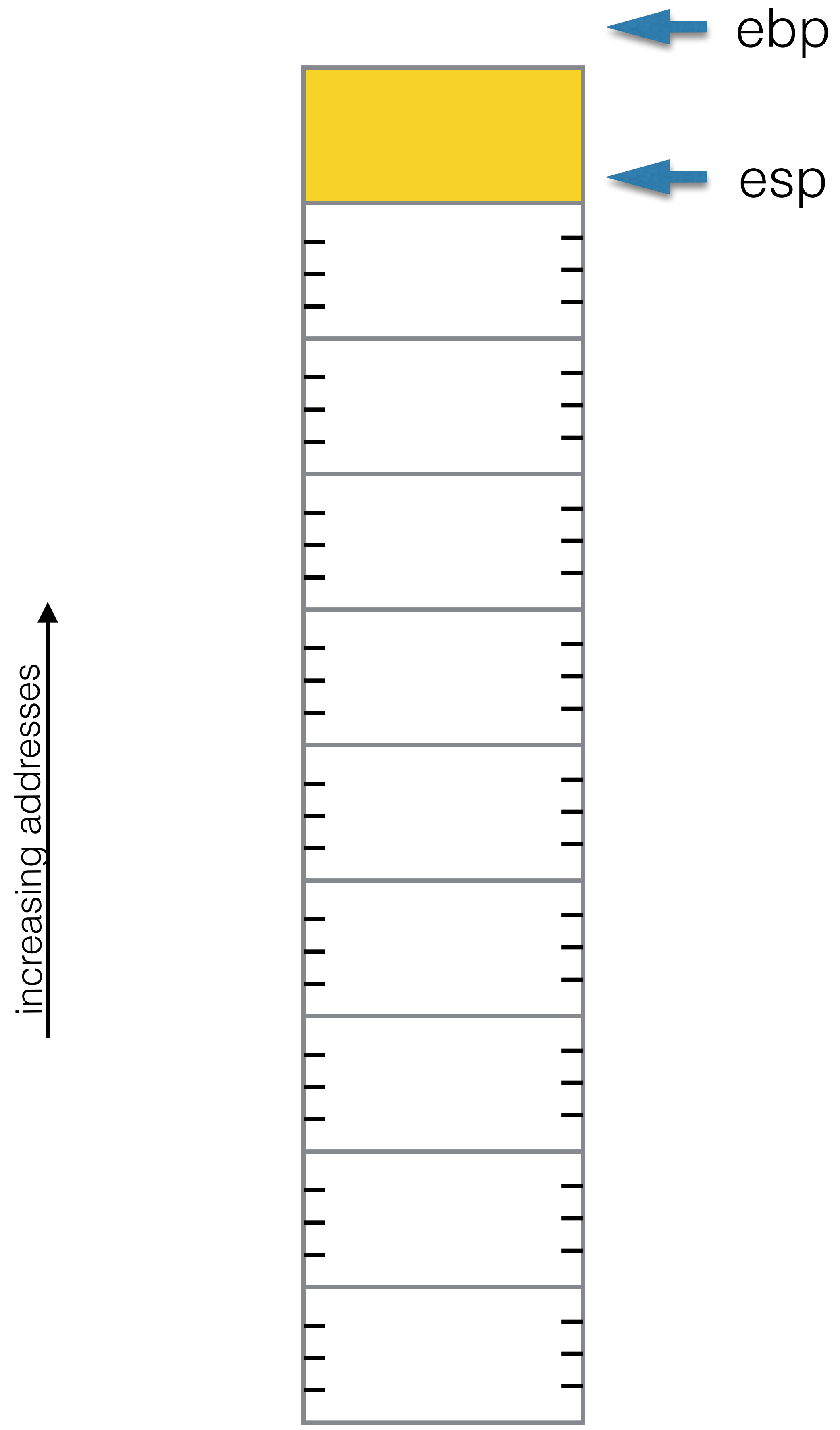
```
        section .data
a       dd      1234
b       dd      5555
result  dd      0

        section .text

        mov     eax,dword [a]
        mov     ebx,dword [b]
        call    sum
*       mov     dword [result], eax


        mov     eax,SYS_EXIT
        mov     ebx,0
        int     0x80

;;; ----------------------------
;;; sum function
;;; adds eax+ebx and return in eax
;;; registers modified:  ax
;;; ----------------------------
sum:    add     eax,ebx
        ret
```



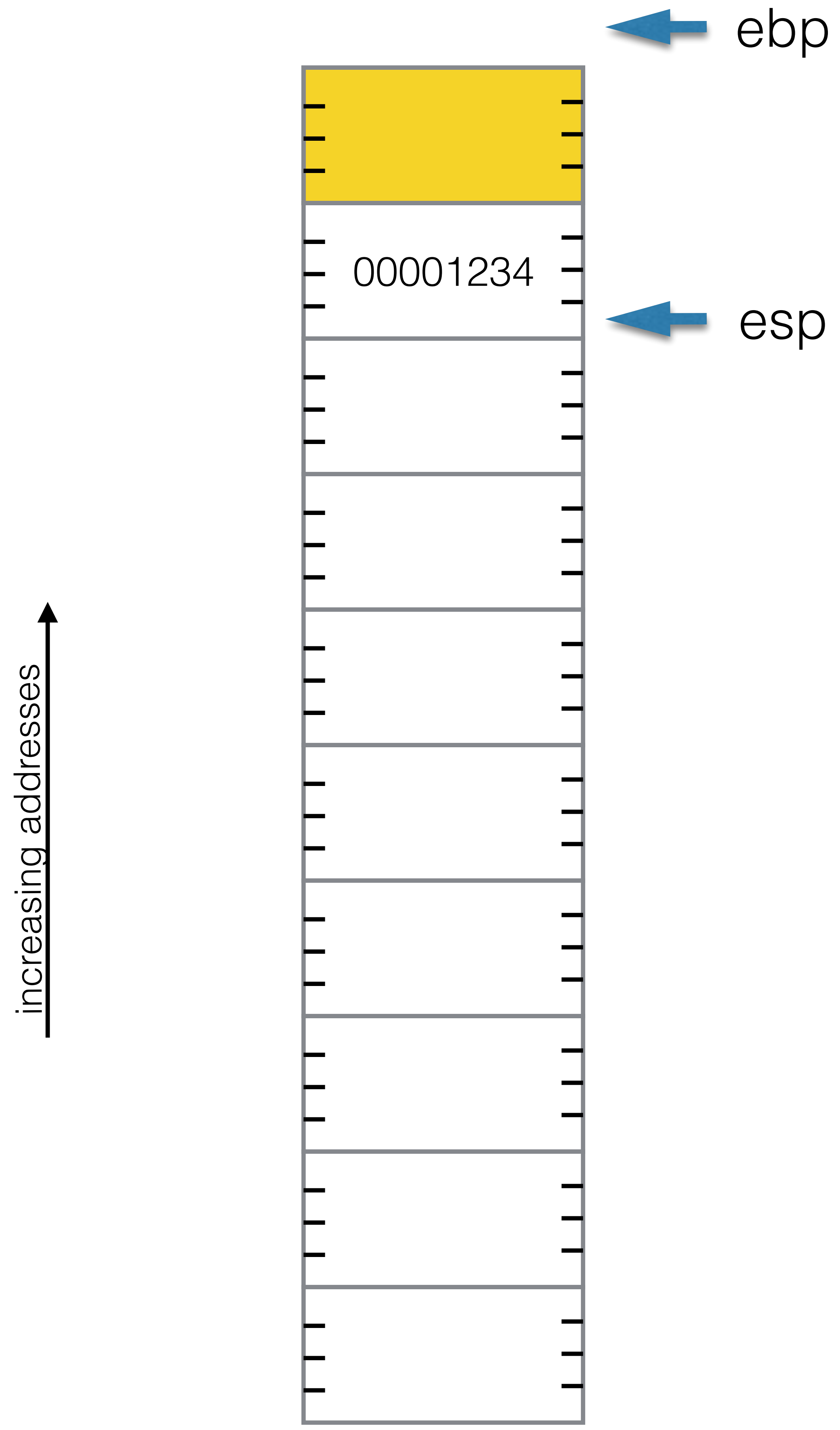
Pass *a* & *b* Through The Stack



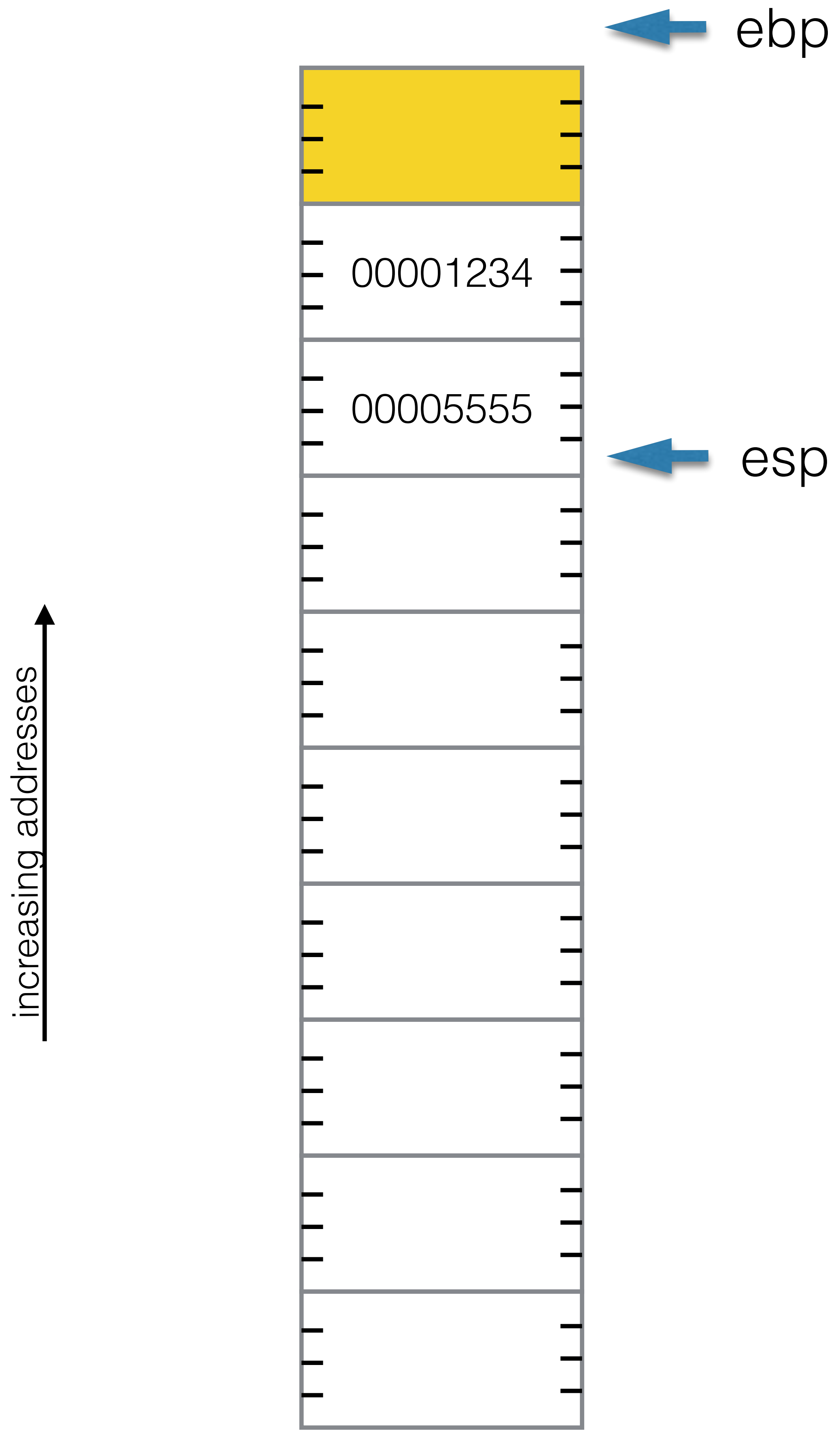
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
        mov     dword[result], eax

          …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



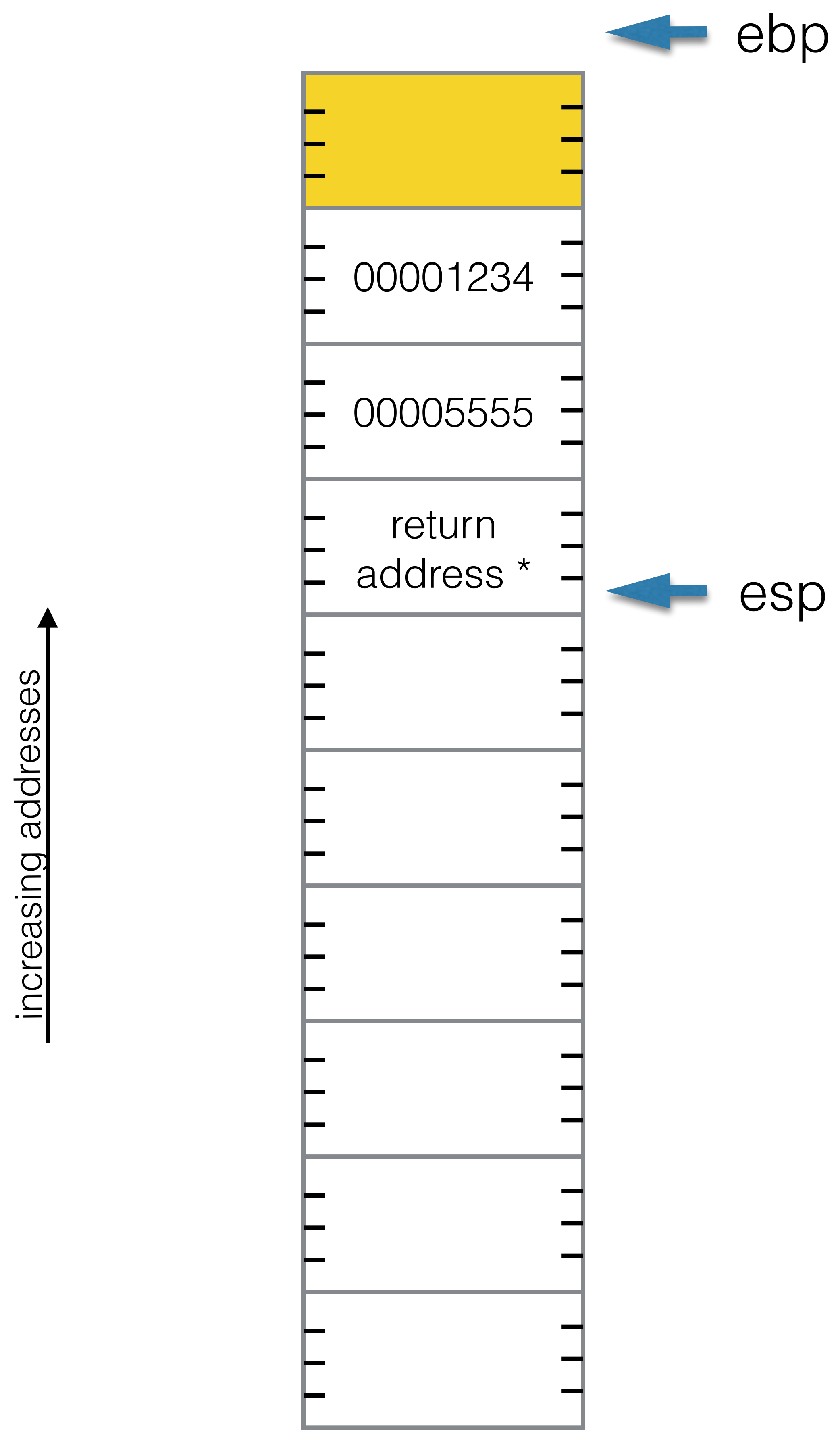
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
        mov     dword[result], eax

         …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



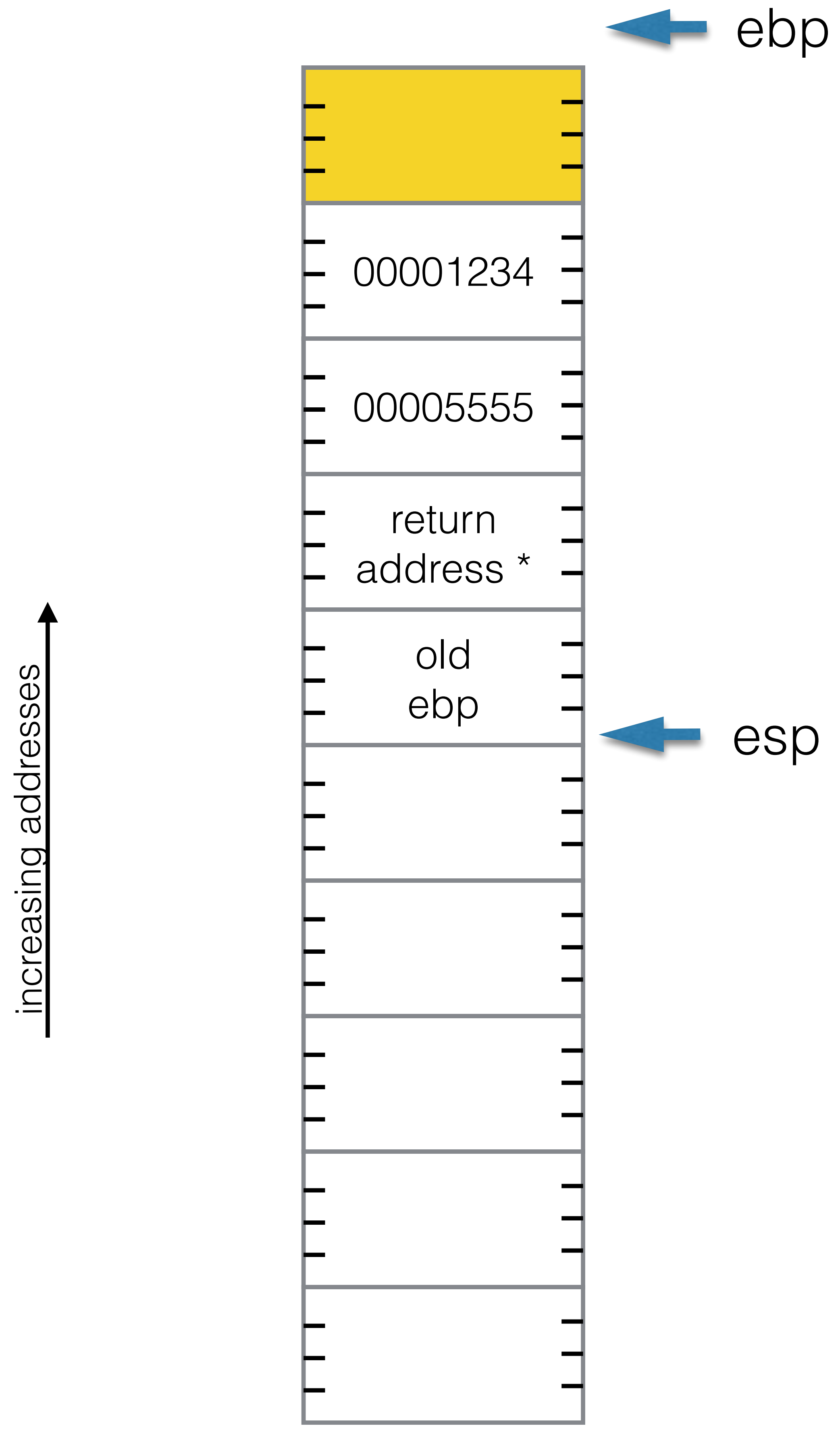
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

         …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



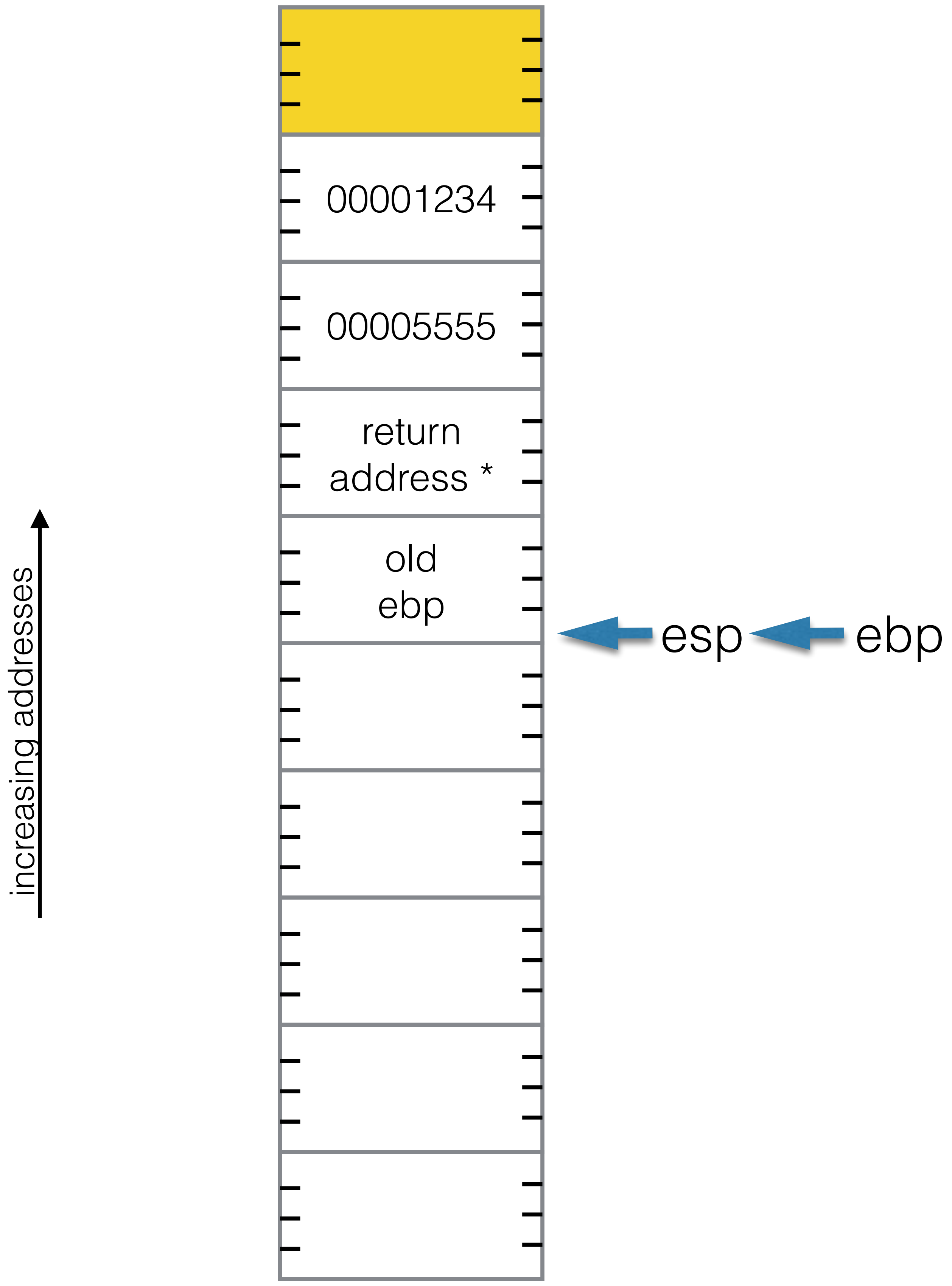
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

        …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



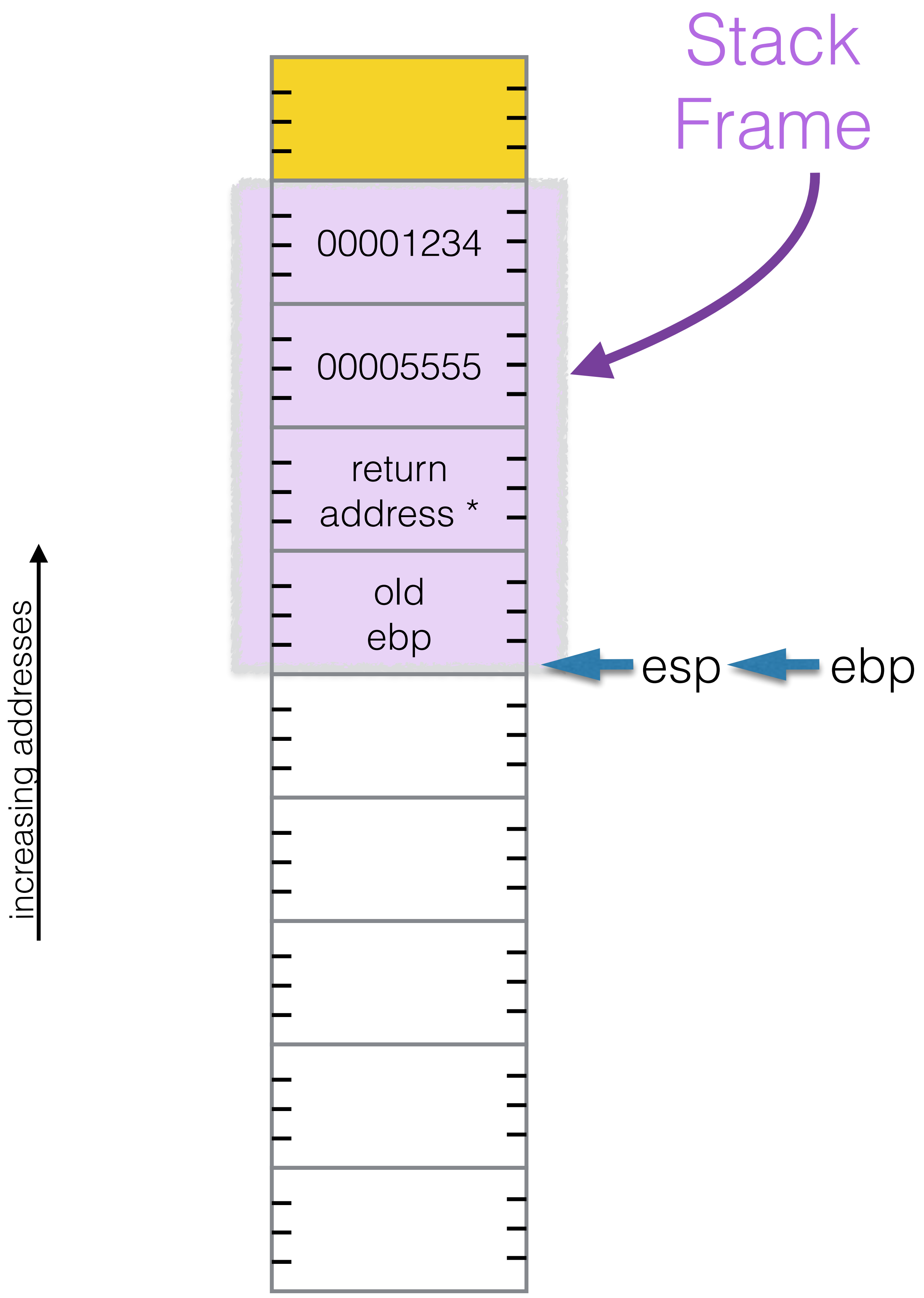
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

        …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



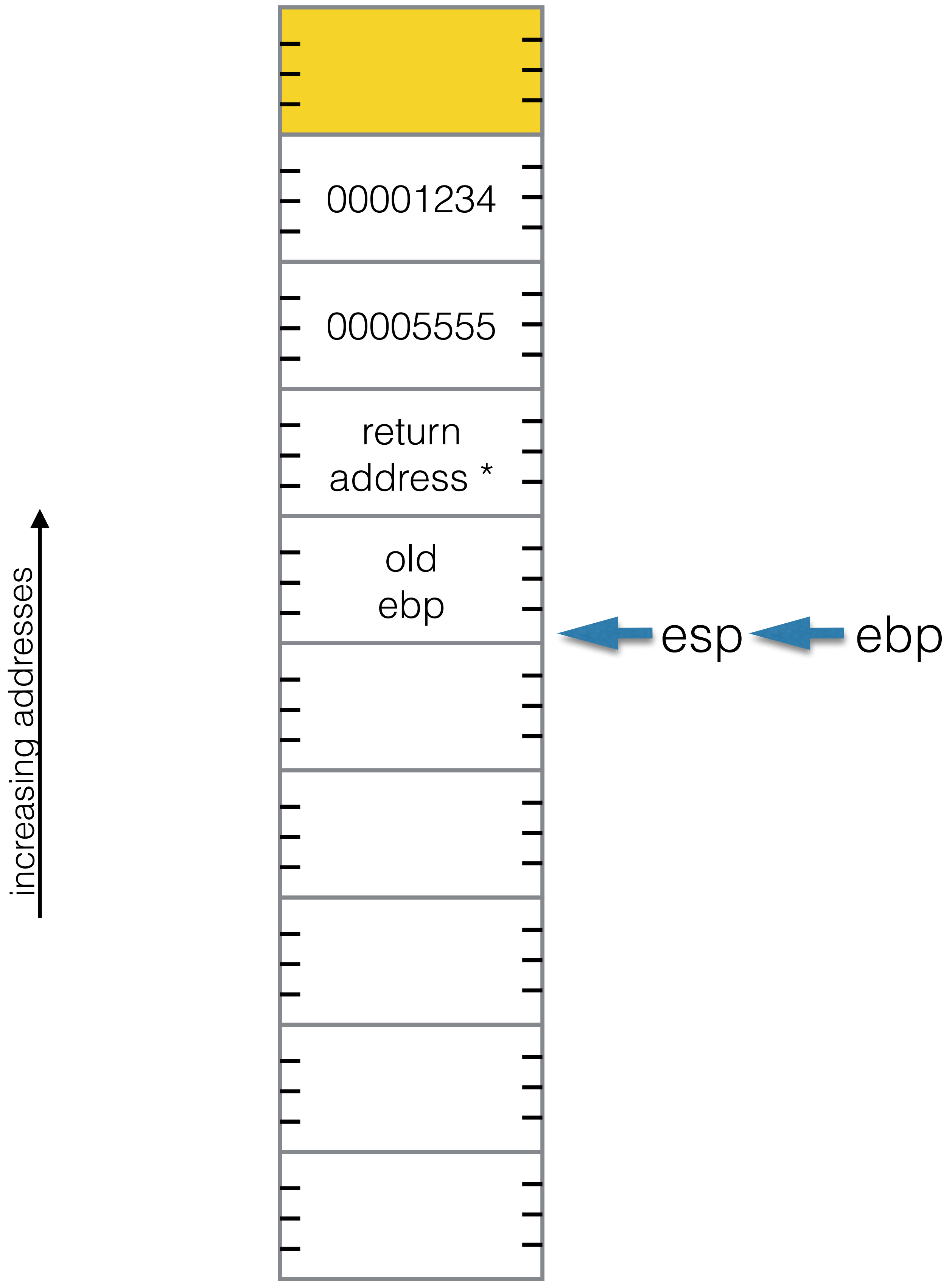
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

        …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



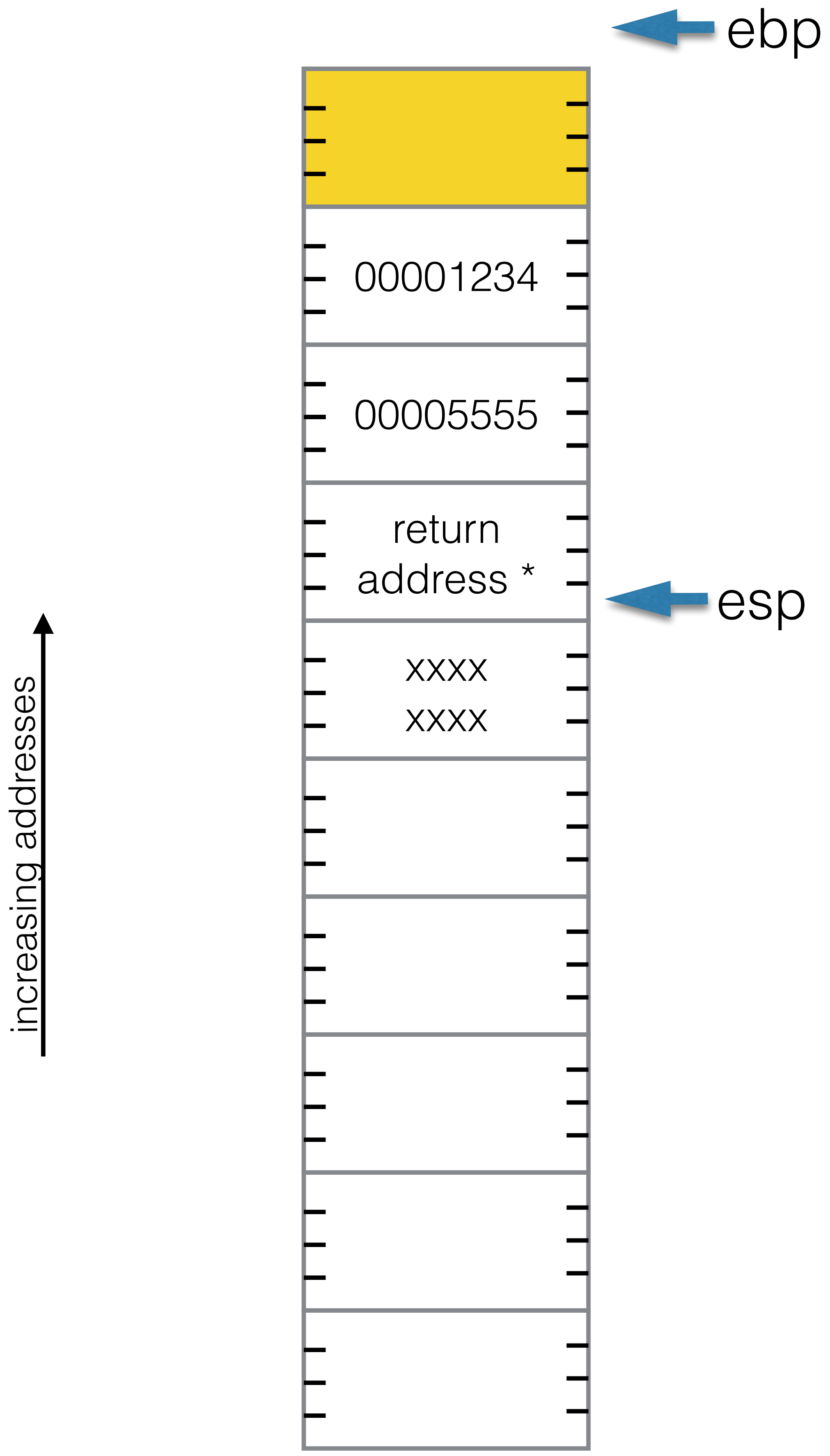
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

        …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



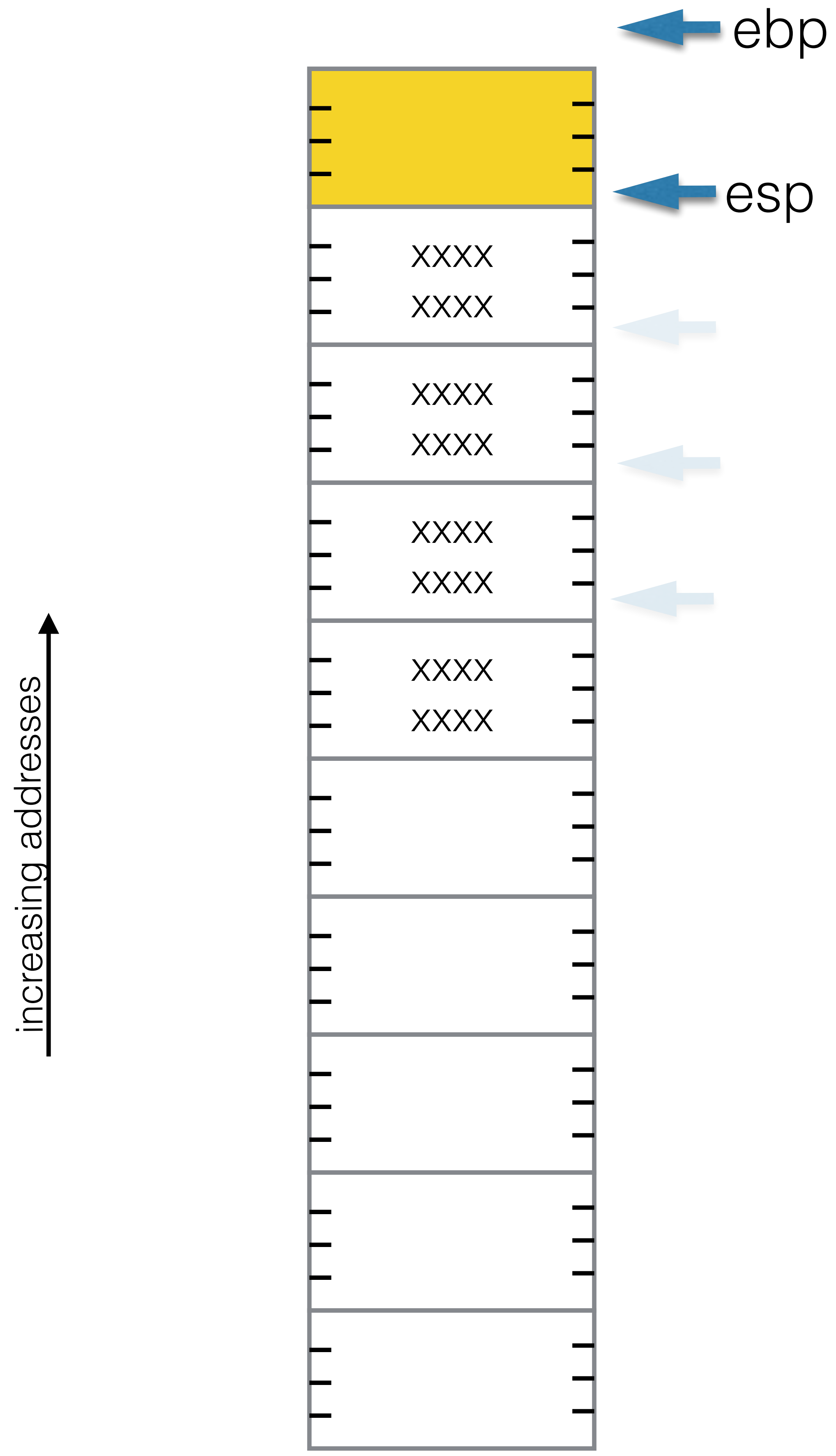
```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

        …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

        …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



```
        section .data
a       dd      0x1234
b       dd      0x5555
result  dd      0

        section .text
        push    dword [a]
        push    dword [b]
        call    sum
*       mov     dword[result], eax

        …
;;; sum function
sum:    push    ebp
        mov     ebp,esp

        mov     eax,dword [ebp+8]
        add     eax,dword [ebp+12]


        pop     ebp
        ret     8
```



Question:

- Why do we bother pushing **ebp** when the function starts?

Exercises

Exercises 1 & 2

- Write a new **printDec()** function that gets the number to print through the stack. The function should not modify any register upon its return.
- Write a new **printString()** function that prints a string, and that gets the string address and length through the stack. The function should not modify any register upon its return.

Exercise 3

- The sum function illustrated above modifies **eax** when it performs the addition. If **eax** had contained an important piece of information in the main program, the function would have overwritten it.

    - Modify the function so that it **saves** eax before using it.

    - Show the **behavior of the stack** as the function executes.

Exercise 4

- Make the **sum** function *call* your new **printString** function to make it print the sum of the two parameters before it (sum) returns to the main program.   Show the stack behavior as the program executes.