

Princeton University
COS 217: Introduction to Programming Systems
The BigInt_add Function

```
enum {MAX_DIGITS = 32768}; /* Arbitrary */

...

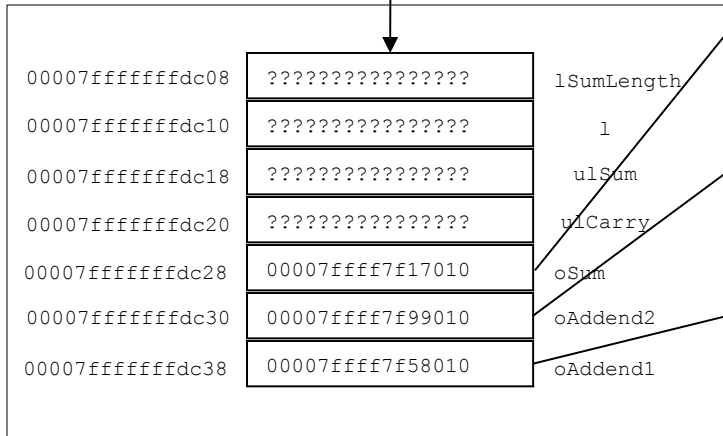
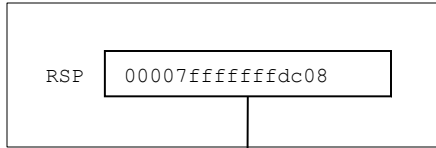
struct BigInt
{
    long lLength;
    unsigned long aulDigits[MAX_DIGITS];
};

...

int BigInt_add(BigInt_T oAddend1, BigInt_T oAddend2, BigInt_T oSum)
{
    unsigned long ulCarry;
    unsigned long ulSum;
    long l;
    long lSumLength;
    ...
}
```

Your addresses may differ

Registers



Stack



Heap

Princeton University
COS 217: Introduction to Programming Systems
The BigInt_add Function: Code: Normal Pattern

Example Code: Access `oAddend2->aulDigits[2]`

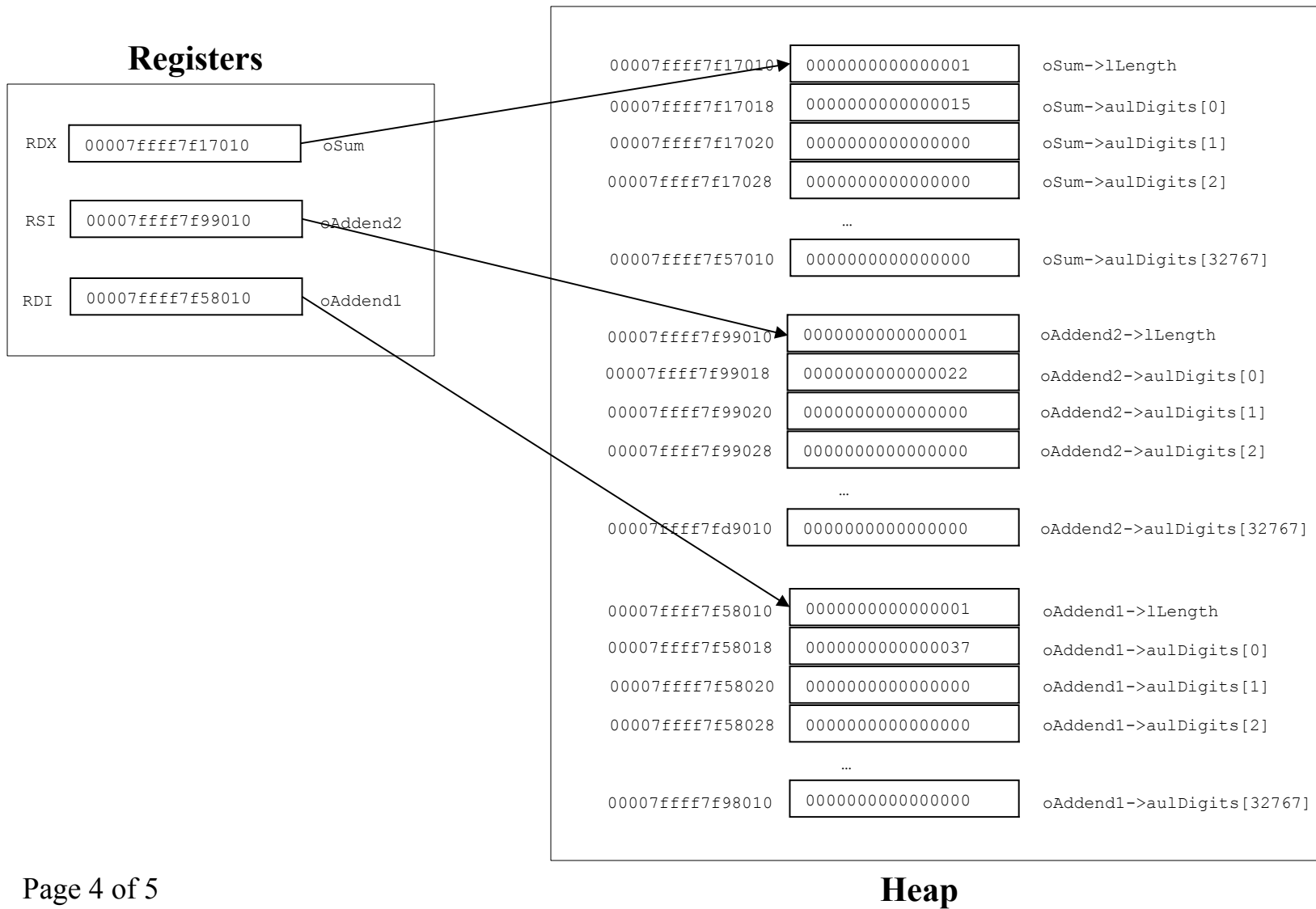
Using indirect addressing:

```
movq %rsp, %rax      # RAX contains 00007fffffffdc08 (hex)
                    # RAX contains the addr of the top of stack
addq $40, %rax       # RAX contains 00007fffffffdc30
                    # RAX contains &oAddend2
movq (%rax), %rax    # RAX contains 00007ffff7f99010 (hex)
                    # RAX contains oAddend2
addq $8, %rax        # RAX contains 00007ffff7f99018 (hex)
                    # RAX contains oAddend2->aulDigits
movq $2, %r10        # R10 contains 0000000000000002 (hex)
                    # R10 contains the index
salq $3, %r10        # R10 contains 0000000000000010 (hex)
                    # R10 contains a byte offset
addq %r10, %rax      # RAX contains 00007ffff7f99028 (hex)
                    # RAX contains oAddend2->aulDigits + 2
movq (%rax), %rax    # RAX contains 0000000000000000 (hex)
                    # RAX contains *(oAddend2->aulDigits + 2)
                    # RAX contains oAddend2->aulDigits[2]
```

Using scaled-indexed addressing:

```
movq 40(%rsp), %rax  # RAX contains 00007ffff7f99010 (hex)
                    # RAX contains oAddend2
movq $2, %r10        # R10 contains 0000000000000002 (hex)
                    # R10 contains the index
movq 8(%rax, %r10, 8), %rax # RAX contains 0000000000000000 (hex)
                    # RAX contains oAddend2->aulDigits[2]
```

Your addresses may differ



Princeton University
COS 217: Introduction to Programming Systems
The BigInt_add Function: Code: Optimized Pattern

Example Code: Access `oAddend2->aulDigits[2]`

Using indirect addressing:

```
movq %rsi, %rax      # RAX contains 00007ffff7f99010 (hex)
                    # RAX contains oAddend2
addq $8, %rax        # RAX contains 00007ffff7f99018 (hex)
                    # RAX contains oAddend2->aulDigits
movq $2, %r10       # R10 contains 0000000000000002 (hex)
                    # R10 contains the index
salq $3, %r10       # R10 contains 0000000000000010 (hex)
                    # R10 contains a byte offset
addq %r10, %rax     # RAX contains 00007ffff7f99028 (hex)
                    # RAX contains oAddend2->aulDigits + 2
movq (%rax), %rax   # RAX contains 0000000000000000 (hex)
                    # RAX contains *(oAddend2->aulDigits + 2)
                    # RAX contains oAddend2->aulDigits[2]
```

Using scaled-indexed addressing:

```
movq $2, %r10       # R10 contains 0000000000000002 (hex)
                    # R10 contains the index
movq 8(%rsi, %r10, 8), %rax # RAX contains 0000000000000000 (hex)
                    # RAX contains oAddend2->aulDigits[2]
```