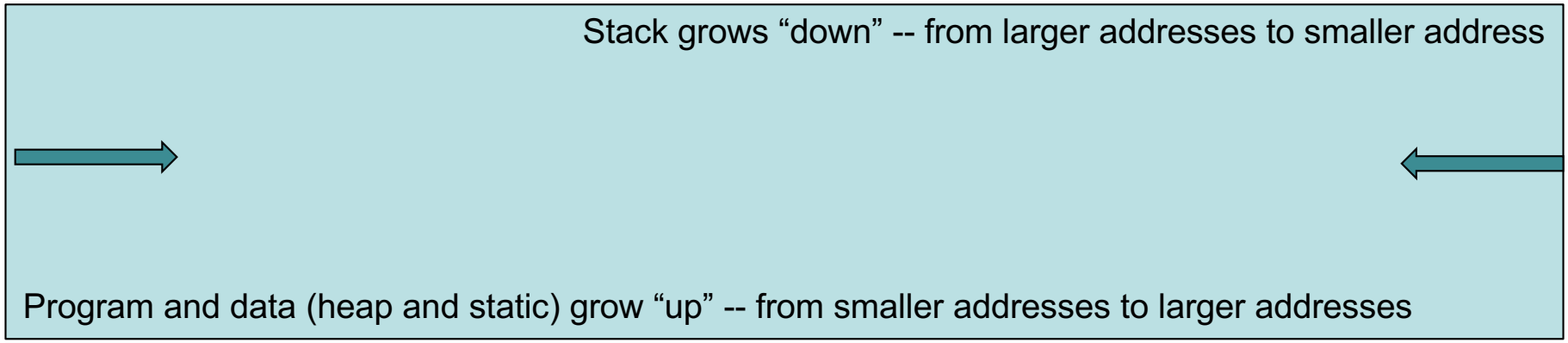
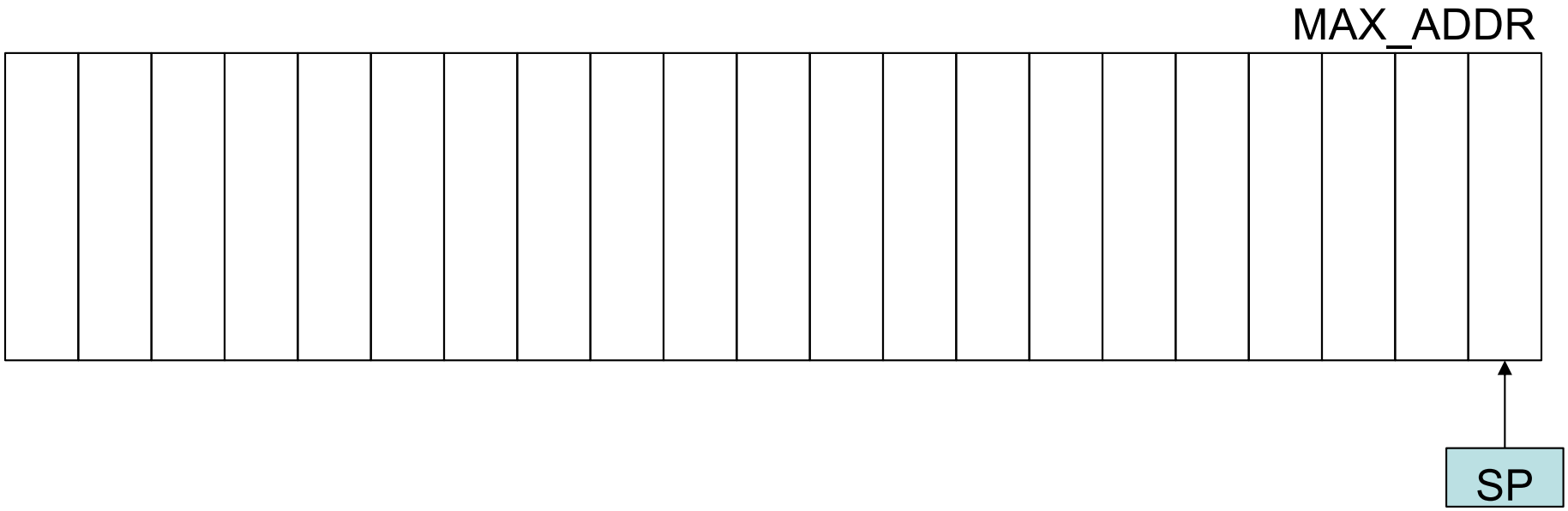
Y86 Stack

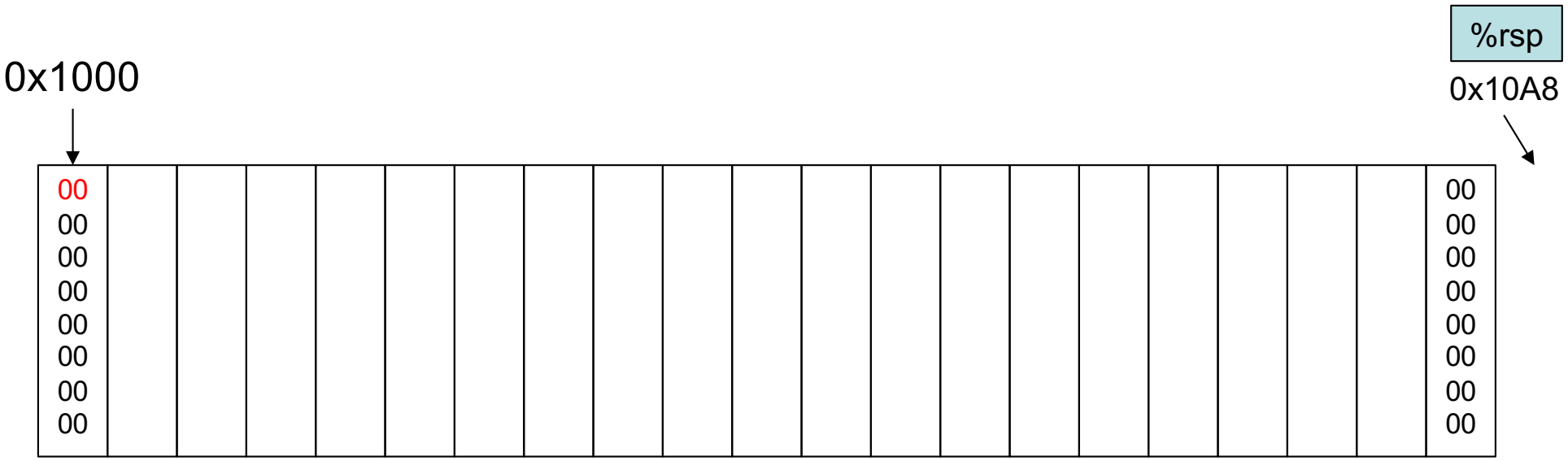
- Topic:
  - Using stack instructions on y86
- Learning Objectives
  - Draw stack diagrams
  - Use PUSHQ/POPQ instructions
  - Explain how PUSHQ and POPQ are implemented
  - Allocate space for the stack
  - Initialize the stack pointer
- Reading:
  - No new reading

Stacks (in HW)


Stack grows “down” -- from larger addresses to smaller address
Program and data (heap and static) grow “up” -- from smaller addresses to larger addresses


MAX_ADDR
SP

y86 Stack


%rsp
0x10A8
0x1000
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00

y86 Stack (PUSHQ)

2-byte instruction

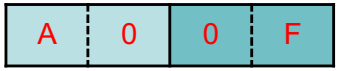
PUSHQ rA

R[%rsp] = R[%rsp] – 8

M_8[R[%rsp]] <- R[%rA]

PUSHQ %rax

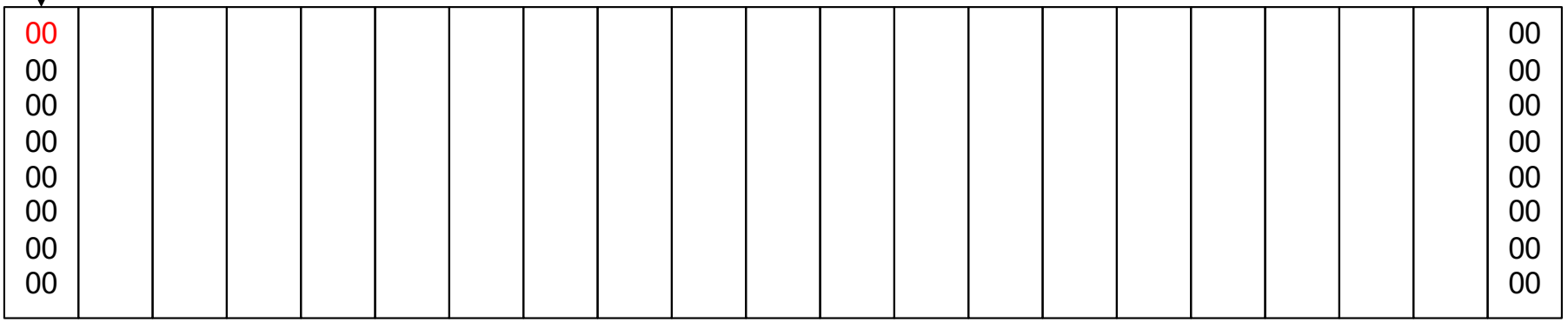
2-byte instruction



%rax	%rsp	%r8	%r12
%rcx	%rbp	%r9	%r13
%rdx	%rsi	%r10	%r14
%rbx	%rdi	%r11	

0x1000

0x10A8



y86 Stack (PUSHQ)

2-byte instruction

PUSHQ rA

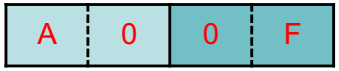
R[%rsp] = R[%rsp] – 8

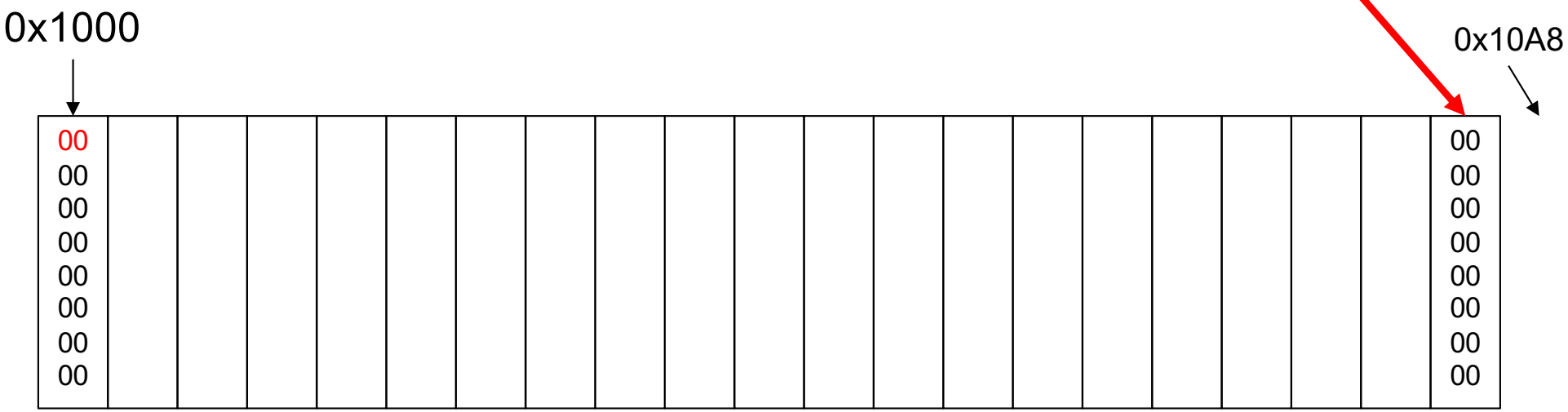
M_8[R[%rsp]] <- R[%rA]

%rax	%rsp	%r8	%r12
%rcx	%rbp	%r9	%r13
%rdx	%rsi	%r10	%r14
%rbx	%rdi	%r11	

PUSHQ %rax

2-byte instruction





y86 Stack (PUSHQ)

2-byte instruction

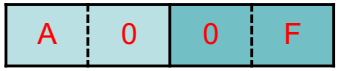
PUSHQ rA

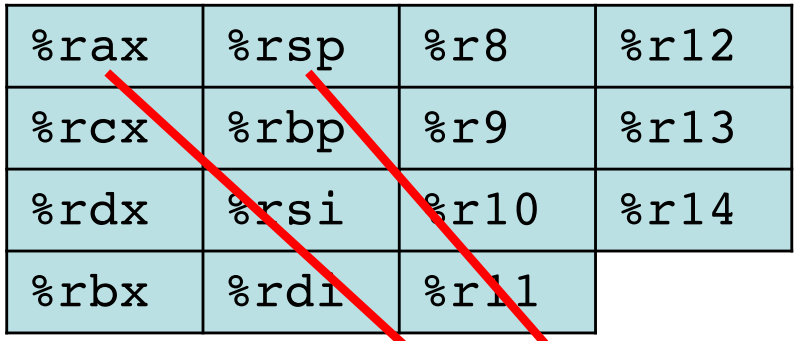
R[%rsp] = R[%rsp] – 8

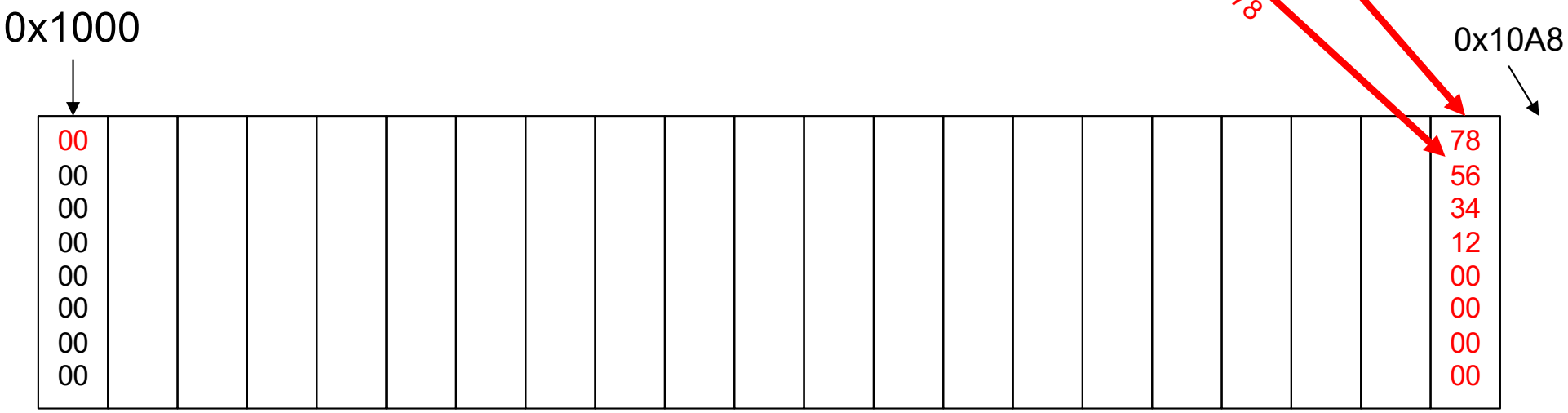
M_8[R[%rsp]] <- R[%rA]

PUSHQ %rax

2-byte instruction







y86 Stack (POPQ)

2-byte instruction

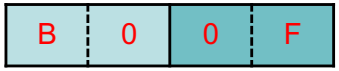
POPQ rA

R[%rA] <- M_8[R[%rsp]]

R[%rsp] = R[%rsp] + 8

POPQ %rax

2-byte instruction



%rax	%rsp	%r8	%r12
%rcx	%rbp	%r9	%r13
%rdx	%rsi	%r10	%r14
%rbx	%rdi	%r11	

0x1088

0x1000

0x10A8

00																01	ED	FE	78
00																00	FE	CA	56
00																00	00	00	34
00																00	00	00	12
00																00	00	00	00
00																00	00	00	00
00																00	00	00	00
00																00	00	00	00

y86 Stack (POPQ)

2-byte instruction

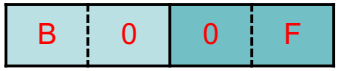
POPQ rA

R[%rA] <- M_8[R[%rsp]]

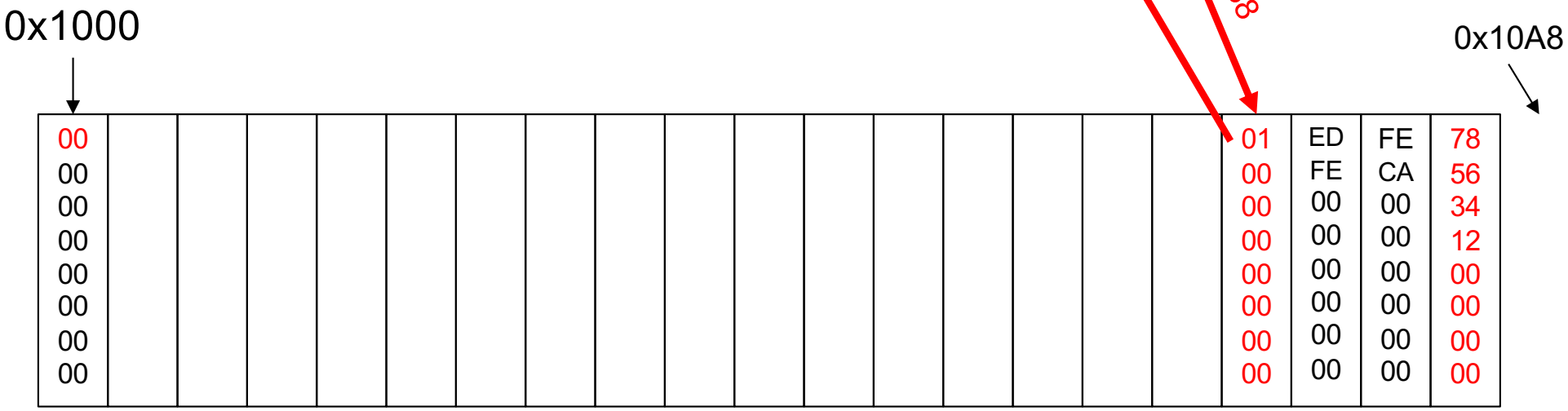
R[%rsp] = R[%rsp] + 8

POPQ %rax

2-byte instruction



%rax	%rsp	%r8	%r12
%rcx	%rbp	%r9	%r13
%rdx	%rsi	%r10	%r14
%rbx	%rdi	%r11	



y86 Stack (POPQ)

2-byte instruction

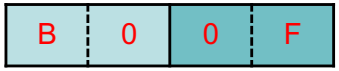
POPQ rA

R[%rA] <- M_8[R[%rsp]]

R[%rsp] = R[%rsp] + 8

POPQ %rax

2-byte instruction



%rax	%rsp	%r8	%r12
%rcx	%rbp	%r9	%r13
%rdx	%rsi	%r10	%r14
%rbx	%rdi	%r11	

0x1090

0x1000

0x10A8

00																~~01~~	ED	FE	78
00																~~00~~	FE	CA	56
00																~~00~~	00	00	34
00																~~00~~	00	00	12
00																~~00~~	00	00	00
00																~~00~~	00	00	00
00																~~00~~	00	00	00
00																~~00~~	00	00	00

Swap Using the Stack

```
#Initialize the stack
irmovq 0x2000, %rsp

#Initialize rax and rbx
irmovq 0x1234, %rax
irmovq 0xCAFE, %rbx

#Now, do the swap
pushq %rax
rrmovq %rbx, %rax
popq %rbx
```