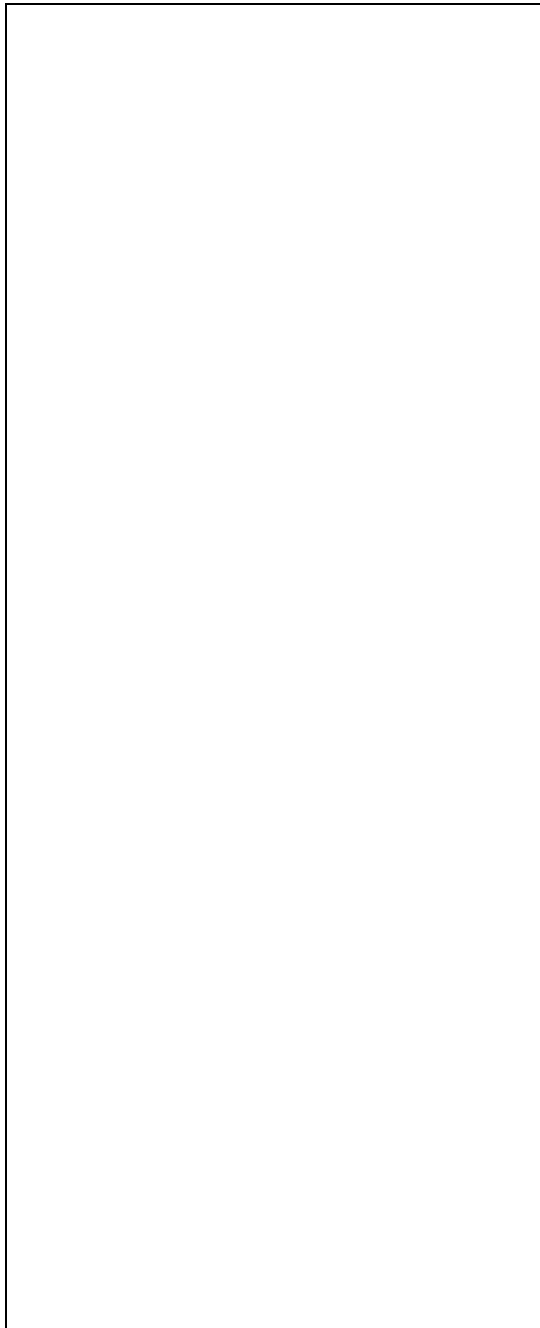


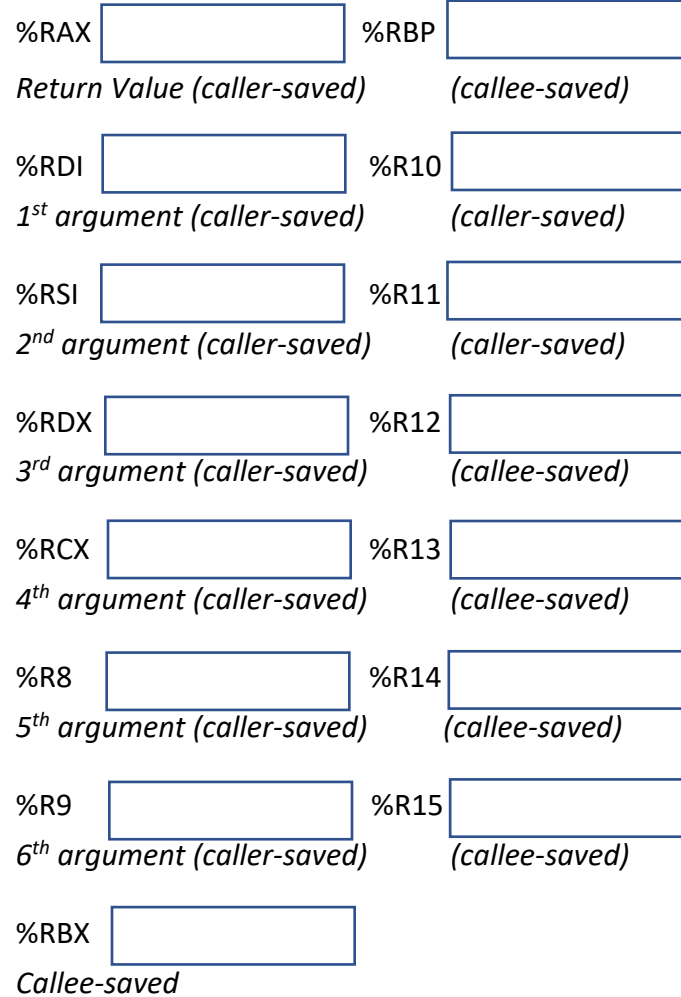
## MEMORY DIAGRAM

Higher Addresses

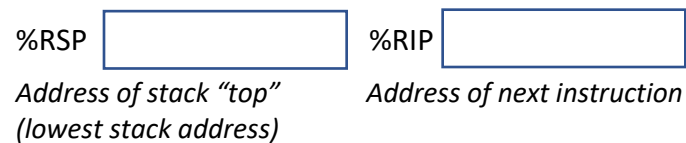


Lower Addresses

## REGISTERS DIAGRAM



## SPECIAL REGISTERS



## COMMON INSTRUCTIONS

### Data movement:

**mov a, b** – copy a into b  
**movs a, b** – copy sign-extended a into b  
**movz a, b** – copy zero-extended a into b  
**lea a, b** – save address of memory addressing expression a into b

### Stack:

**push a** – push a on stack  
**pop a** – pop value from top of stack into a

### Procedures:

**call target** - push return address on stack and jump to *target*  
**ret** – pop return address from stack and jump to return address

### Arithmetic

**add a, b** – save sum (a+b) into b  
**sub a, b** – save difference (b-a) into b  
**imul a, b** – save signed product (a\*b) into b  
**and a, b** – save bitwise AND (a&b) into b  
**or a, b** – save bitwise OR (a|b) into b  
**Shift:** save b shifted \_\_\_\_ by a into b  
**shl/sal a, b** – to the left (b<<a)  
**shr a, b** – logically to the right (b>>a)  
**sar a, b** – arithmetically to the right (b>>a)

### Compare/test: set condition codes/flags...

**cmp a, b** – based on difference (b-a)  
**test a, b** – based on bitwise AND (a&b)

### Conditional jump: jump to target if...

**jg target** – greater than (zero)  
**je target** – equal to (zero)  
**jne target** – not equal to (zero)  
**jle target** – less than or equal to (zero)

### Unconditional jump:

**jmp target** – jump to *target*

## MEMORY ADDRESS SYNTAX

$D(R_b, R_r, S) \Rightarrow \text{Mem}[\text{Reg}[R_b] + S * \text{Reg}[R_r] + D]$   
 S can be 1, 2, 4, or 8 only