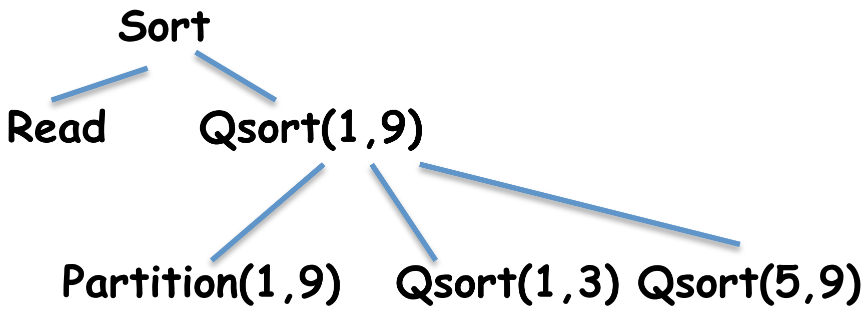
Function Activations & Activation Trees

```
Sort(list)
{
Read {}
Qsort (low,high){
    int x
      Partition (low,high){}
    x = Partition(low,high)
    call Qsort(low, x-1 }
    call Qsort(x+1,high)
```



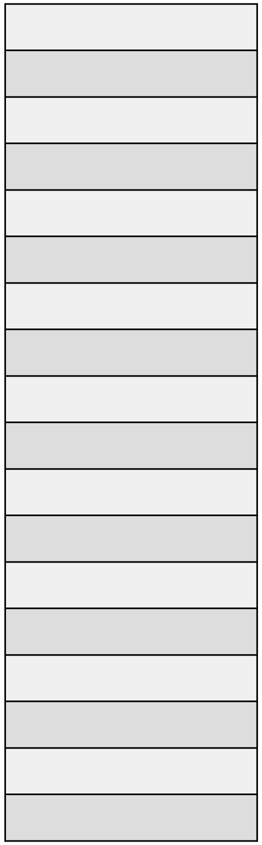
Exercise

```
Class Main {
    g() : Int { 1 };
    f(x:Int):  Int { if x = 0 then g() else f(x - 1) fi};
    main(): Int {{f(3); }};
}
```

What is the activation tree for this example?

Runtime Stack in MeggyJava (with recursion)

```
class PA4raindrop {
public static void main(String[] whatever){
   while (true) {
     new Cloud().rain((byte)3,(byte)7); …}   }}
class Cloud {
   public void rain(byte x, byte y) {
  // light up x,y if is in bounds and continue recursion
   if (this.inBounds(x, y)) {
      Meggy.setPixel(x, y, Meggy.Color.BLUE)
      if (this.inBounds(x,(byte)(y+(byte)1))) {
         Meggy.setPixel(x, (byte)(y+(byte)1), Meggy.Color.DARK);
         } else {}
     Meggy.delay(100);
     this.rain(x, (byte)(y-(byte)1));
    } else {}  }
public boolean inBounds(byte x, byte y) {
   return ((byte)(0-1) < y) && (y < (byte)8);  }
}
```



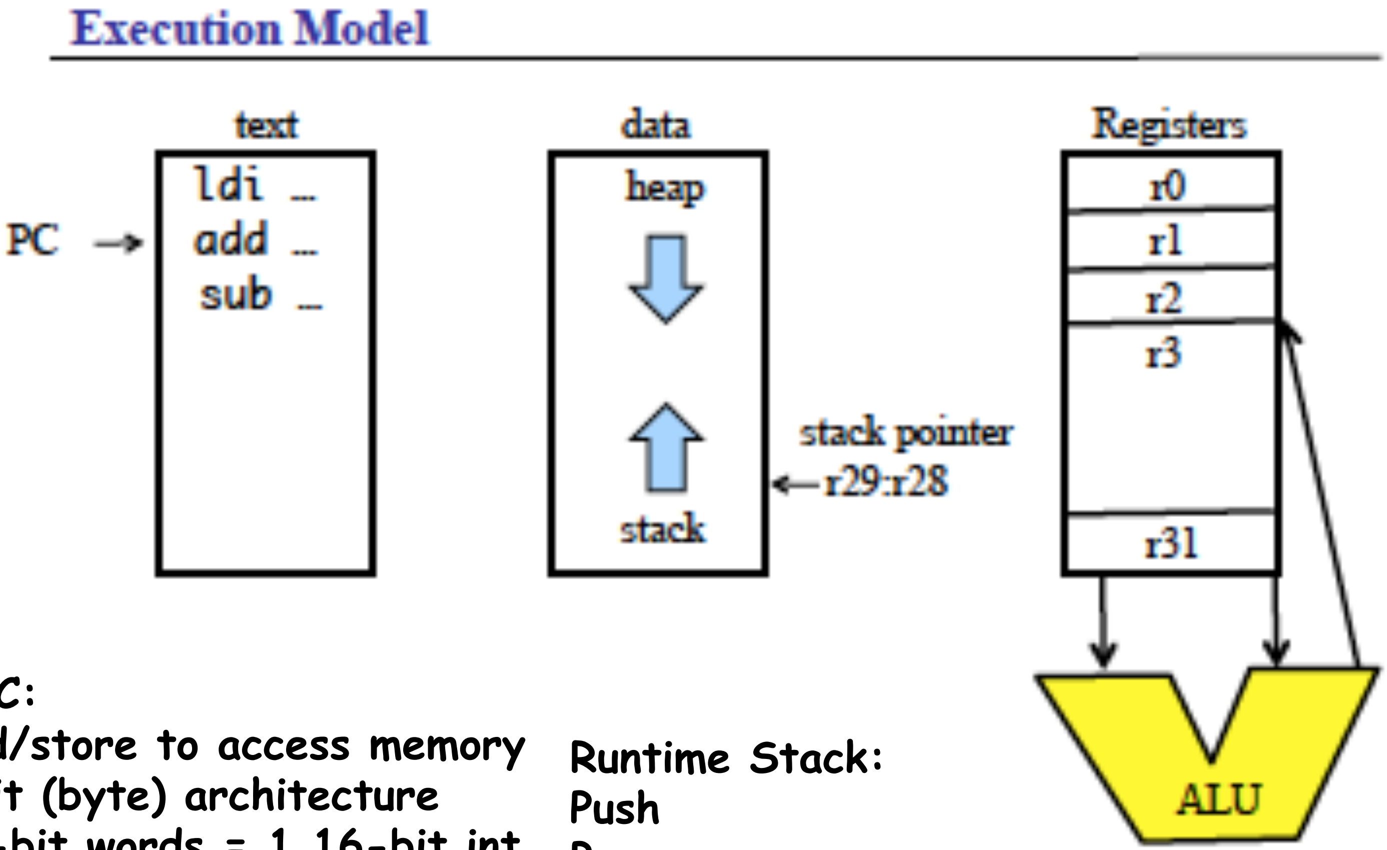
Notes

- The activation tree depends on run-time behavior

- The activation tree may be different for every program input

- Need to keep track of procedure activations during execution
- Details depend on machine architecture

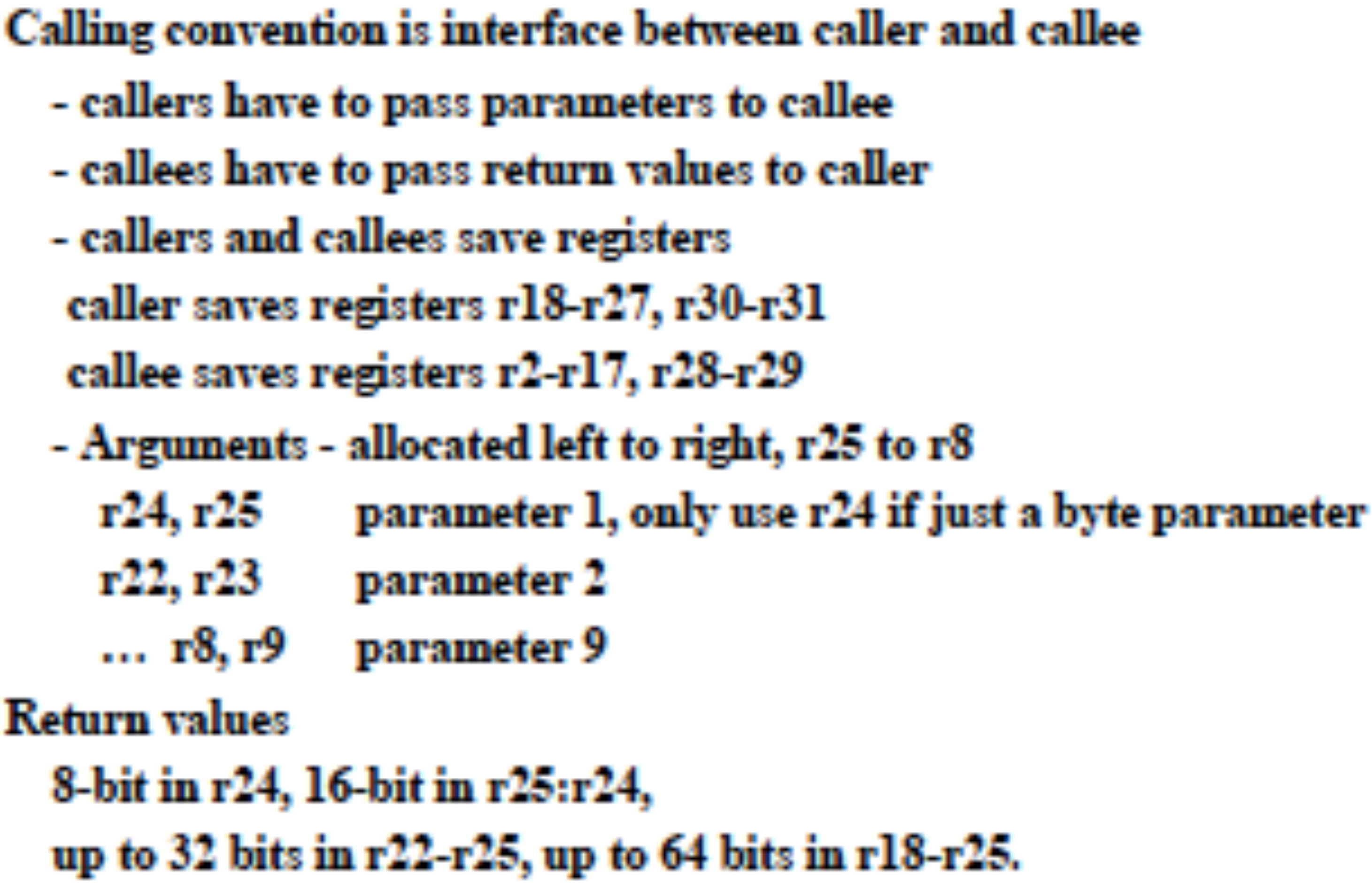
AVR Instruction Set Architecture



RISC:
Load/store to access memory
8-bit (byte) architecture
2 8-bit words = 1 16-bit int
Paired regs = 1 16-bit int

Runtime Stack:
Push
Pop
Stack pointer

Calling Convention



Calling convention is interface between caller and callee

- callers have to pass parameters to callee
- callees have to pass return values to caller
- callers and callees save registers

caller saves registers r18-r27, r30-r31

callee saves registers r2-r17, r28-r29

- Arguments - allocated left to right, r25 to r8

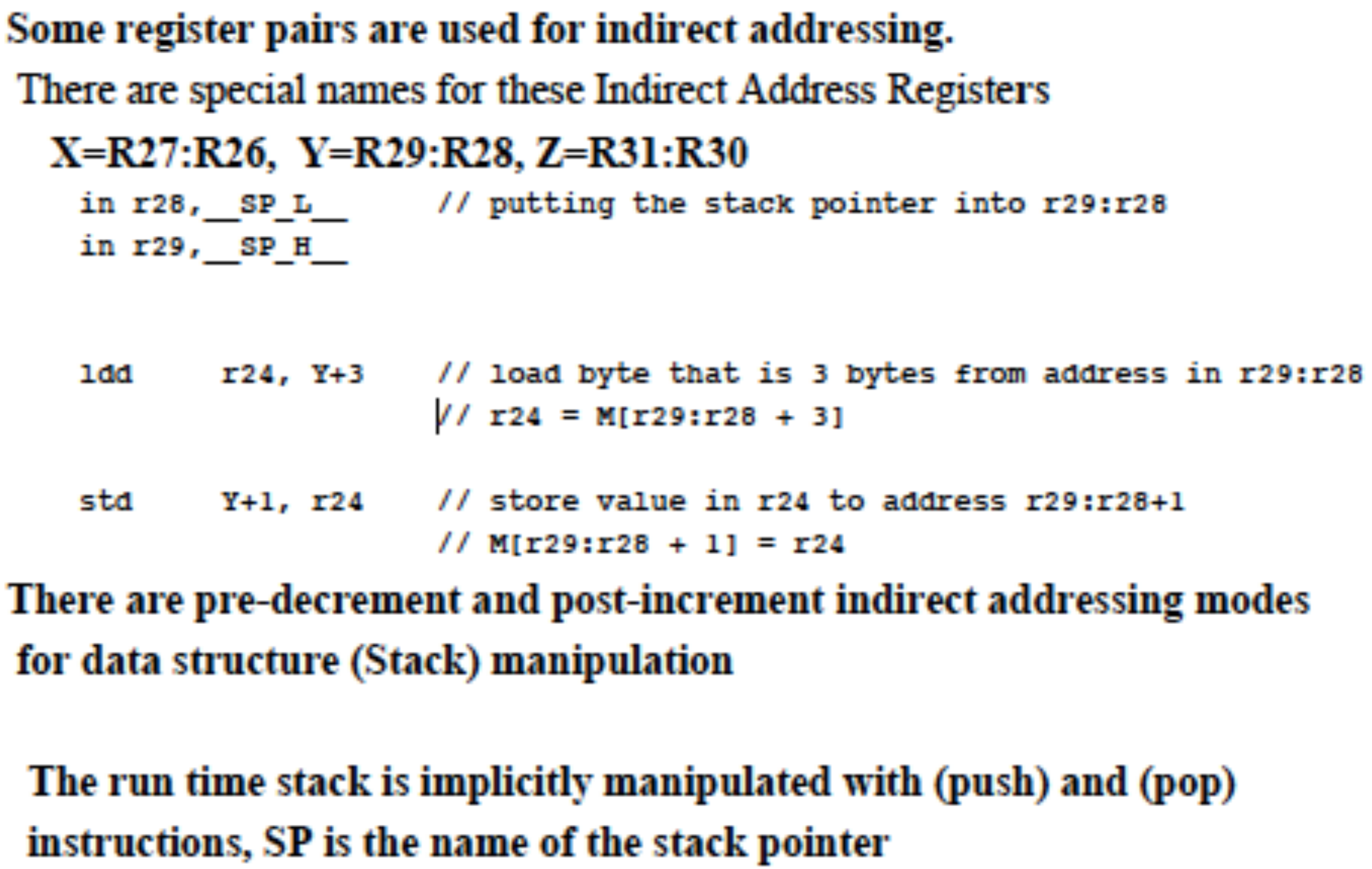
r24, r25	parameter 1, only use r24 if just a byte parameter
r22, r23	parameter 2
… r8, r9	parameter 9

Return values

8-bit in r24, 16-bit in r25:r24,

up to 32 bits in r22-r25, up to 64 bits in r18-r25.

Stack Pointer and Frame Pointer



Some register pairs are used for indirect addressing.

There are special names for these Indirect Address Registers

X=R27:R26, Y=R29:R28, Z=R31:R30

```
in r28,__SP_L__      // putting the stack pointer into r29:r28
in r29,__SP_H__


ldd     r24, Y+3     // load byte that is 3 bytes from address in r29:r28
                     // r24 = M[r29:r28 + 3]

std     Y+1, r24     // store value in r24 to address r29:r28+1
                     // M[r29:r28 + 1] = r24
```

There are pre-decrement and post-increment indirect addressing modes for data structure (Stack) manipulation

The run time stack is implicitly manipulated with (push) and (pop) instructions, SP is the name of the stack pointer

Team Exercise

Examine an assembly code example.

Answer the following:

1. What actions need to happen **at a method call site** at runtime?
2. What code is generated to perform those actions at runtime?

3. What actions need to happen **at the prologue of a method?**
4. What code is generated to perform those actions at runtime?

5. What actions need to happen **at the epilogue of a method**
6. What code is generated to perform those actions at runtime?

So, how does this affect Symbol Table?

Parameters: base, offset

- What should be the base?
- What is the offset? (how computed?)

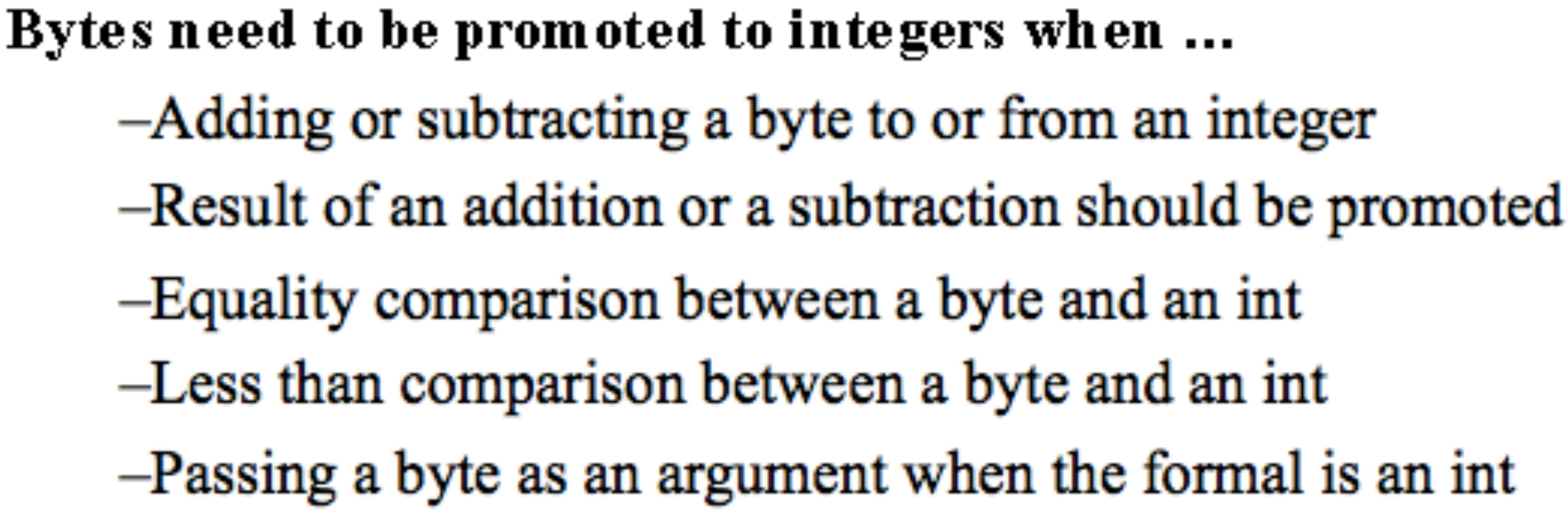
Method: size for parameters on stack

Exercise with PA4raindrop.java.noObjs.s

At each call site, draw the runtime stack and register contents.

At the end of each epilogue, draw the runtime stack and register contents.

Promoting Bytes to Ints



Bytes need to be promoted to integers when …

–Adding or subtracting a byte to or from an integer

–Result of an addition or a subtraction should be promoted

–Equality comparison between a byte and an int

–Less than comparison between a byte and an int

–Passing a byte as an argument when the formal is an int