

Memory-Memory Architectures

Three address architecture: op destination source1 source2

$$Z = X + Y$$

ADD Z, X, Y

$$Z = (X + Y) / (Q - Z)$$

ADD T1, X, Y

SUB T2, Q, Z

DIV Z, T1, T2

Two address architecture: op destination source

(destination serves as both the second source and the destination)

$$Z = X + Y$$

MOV Z, X

ADD Z, Y

$$Z = (X + Y) / (Q - Z)$$

MOV T1, Q

SUB T1, Z

MOV Z, X

ADD Z, Y

DIV Z, T1

Memory-Register Architectures

Multiple register machine: op source register (or op register source)

(The designated register serves as both one source and the destination; on some machines (e.g. Intel 80x86), the memory location may also serve as the destination.) (Multiple register machines generally allow a register to be used instead of the memory operand, as well)

$$Z = X + Y$$

```
LOAD    R1, X
ADD     R1, Y
STORE   Z, R1
```

$$Z = (X + Y) / (Q - Z)$$

```
LOAD    R1, X
ADD     R1, Y
LOAD    R2, Q
SUB     R2, Z
DIV     R1, R2
STORE   Z, R1
```

Single accumulator machine: op source
(AC serves as both a source and destination)

$$Z = X + Y$$

LOAD	X
ADD	Y
STORE	Z

$$Z = (X + Y) / (Q - Z)$$

LOAD	Q
SUB	Z
STORE	T1
LOAD	X
ADD	Y
DIV	T1
STORE	Z

Load-store architecture:

$$Z = X + Y$$

```
LOAD    R1, X
LOAD    R2, Y
ADD     R1, R2, R1
STORE   Z, R1
```

$$Z = (X + Y) / (Q - Z)$$

```
LOAD    R1, X
LOAD    R2, Y
ADD     R1, R2, R1
LOAD    R2, Q
LOAD    R3, Z
SUB     R2, R2, R3
DIV     R1, R1, R2
STORE   Z, R1
```

Stack architecture:

$$Z = X + Y$$

PUSH X

PUSH Y

ADD

POP Z

$$Z = (X + Y) / (Q - Z)$$

PUSH X

PUSH Y

ADD

PUSH Q

PUSH Z

SUB

DIV

POP Z