Chapter 2 - Instructions

(Slide 7)

Accumulator Arch

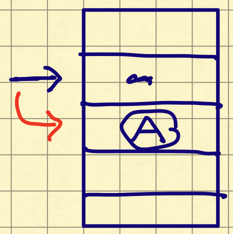
add $\underline{A}$ $\rightarrow$ acc $\leftarrow$ acc + Memory [A]

load A acc $\leftarrow$ Memory [A]

Stack Arch

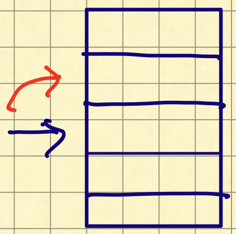
Operands are pushed on the stack from memory or popped off the stack into memory.
Operations take their operands from the stack and then place the results back onto the stack.

push A



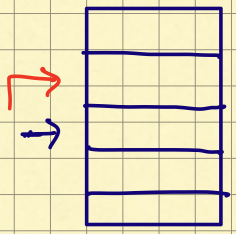
Top $\leftarrow$ Top + 4
Stack [Top] $\leftarrow$ Memory [A]
(To get proper byte addr we adjust stack by 4)

Pop A



Memory [A] $\leftarrow$ stack [Top]
Top $\leftarrow$ Top - 4

add



Stack [Top-4] = stack [Top] + Stack [Top-4]

Top = Top - 4

(Blue arrow Top of the stack before operation)
(red arrow Top of the stack after operation)

General purpose Register Architectures

a) Register-Memory (one operand in Memory)

add Ra, C   Reg[Ra] ← Reg[Ra] + Memory[C]

b) Load-store (register-register) (operands in registers)

add Ra, Rb, Rc   Reg[Ra] ← Reg[Rb] + Reg[Rc]

RISC & CISC Architectures

RISC (Reduced Instruction Set Computers)
CISC (Complex Instruction Set Computers)

CISC

Minimize # of instructions per program by sacrificing # of cycles per instruction.

(longer programs needs more storage thus increasing memory cost.

(embedding operations in a single instr → make instructions more complex)

RISC

RISC does the opposite.
It attempts to reduce the cycles per instruction at the cost of # of instructions per program.

Exec Time = IC × CPI × CCT

RISC	CISC
* Single cycle exec.	* Many multi-cycle ops
* Hardwired Control	* Microcoded multi-cycle ops.
* Load/store Arch.	* Reg-Mem & Mem-Mem
* Few Memory addressing Modes	* Many Modes
* Fixed-length instr format	* Many formats & lengths

Hardwired Control = expressed as a finite state machine
(state diagrams & transitions between states)
Good if # of states is small

Microcoded/microprogramed Control - expressed as a "micro" program
(micro instructions to derive signals needed to execute instructions in the ISA)

ISA Takeaways

CISC $\rightarrow$ complex; small # instr necessary to fit the program into memory but greatly increase complexity of ISA as well.