

Figure 3.2 Computer Components: Top-Level View

Von Neumann Architecture

Computer Function

- program execution

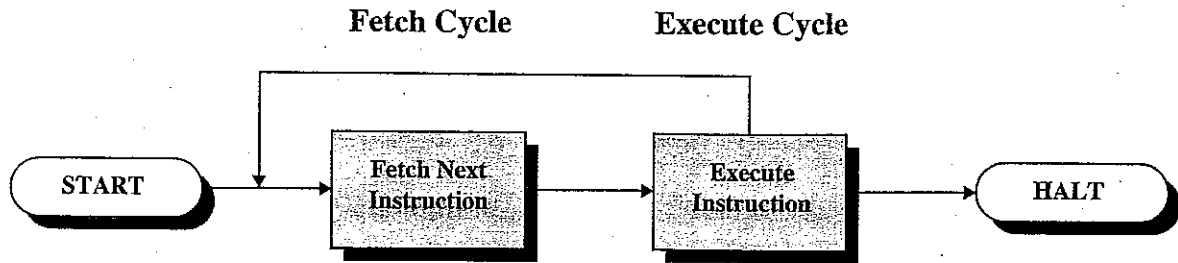


Figure 3.3 Basic Instruction Cycle



(a) Instruction format



(b) Integer format

Program Counter (PC) = Address of instruction
 Instruction Register (IR) = Instruction being executed
 Accumulator (AC) = Temporary storage

} system register
 } user register

(c) Internal CPU registers

0001 = Load AC from Memory
 0010 = Store AC to Memory
 0101 = Add to AC from Memory ($AC \leftarrow AC + M(addr)$)

0110 = subtract
 0011 = shl
 0100 = shr
 1111 = halt

(d) Partial list of opcodes

Figure 3.4 Characteristics of a Hypothetical Machine

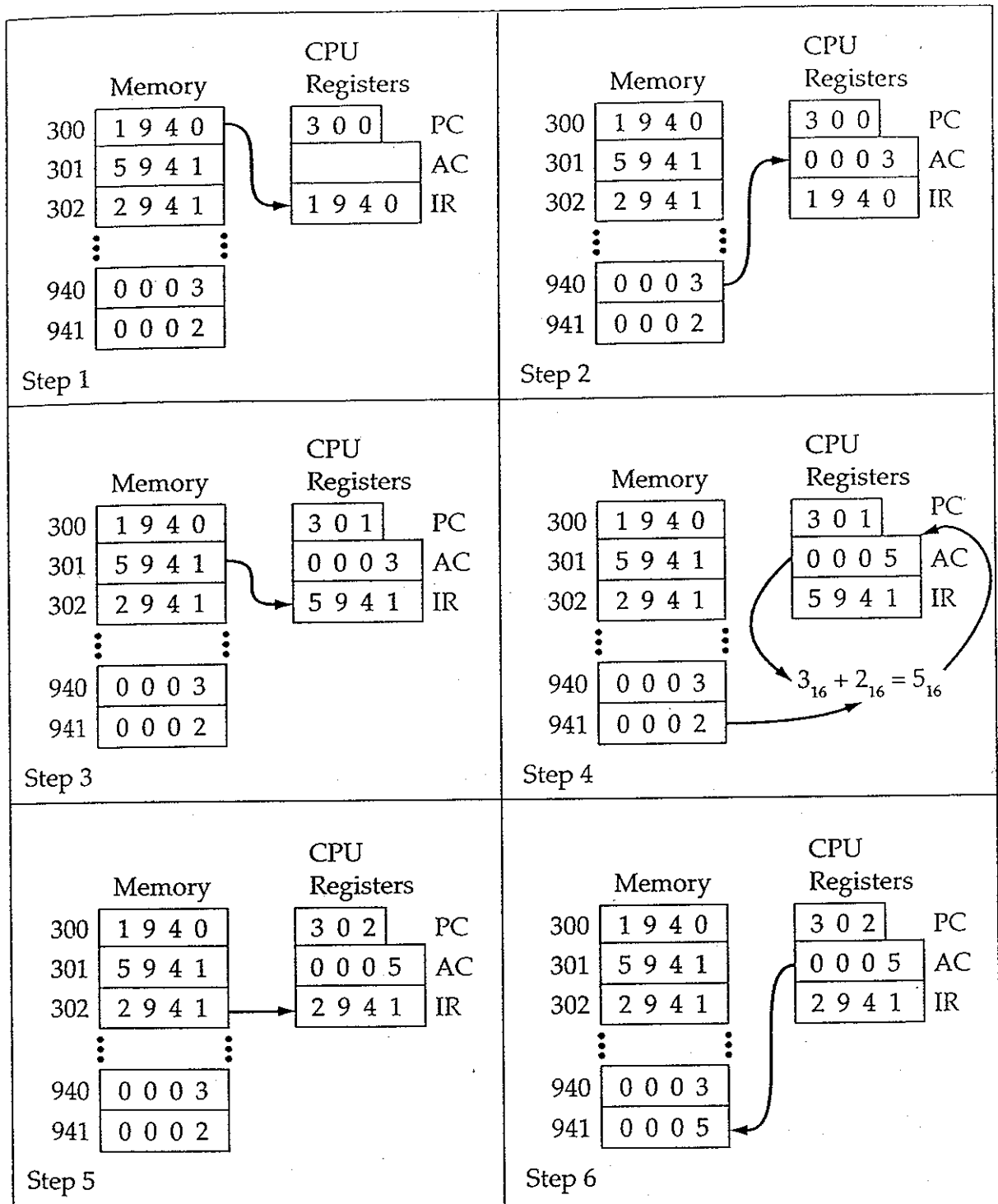


FIGURE 3.5. Example of program execution

Instruction cycle (including interrupt)

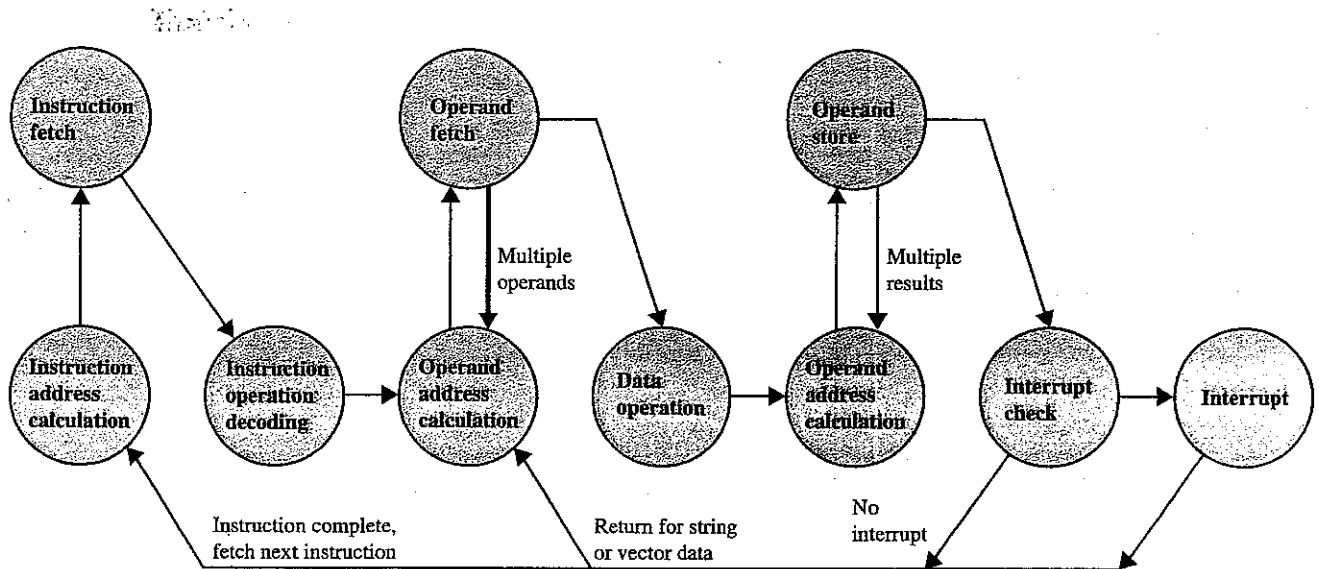


Figure 3.12 Instruction Cycle State Diagram, with Interrupts

Interrupt

- improve efficiency
- cpu speed vs. I/O speed $\approx 100 : 1$
- cpu is idle during I/O operation. With interrupt, cpu can serve other user's program \rightarrow multiprogramming

Classes of interrupt

- program interrupt: divide by zero, overflow, ...
- timer interrupt
- I/O interrupt
- machine interrupt: hardware failure

Interrupt handling routines are in the O.S.

Multiple Interrupts

- (1) FCFS - disable interrupts
- (2) Priority - preemptive

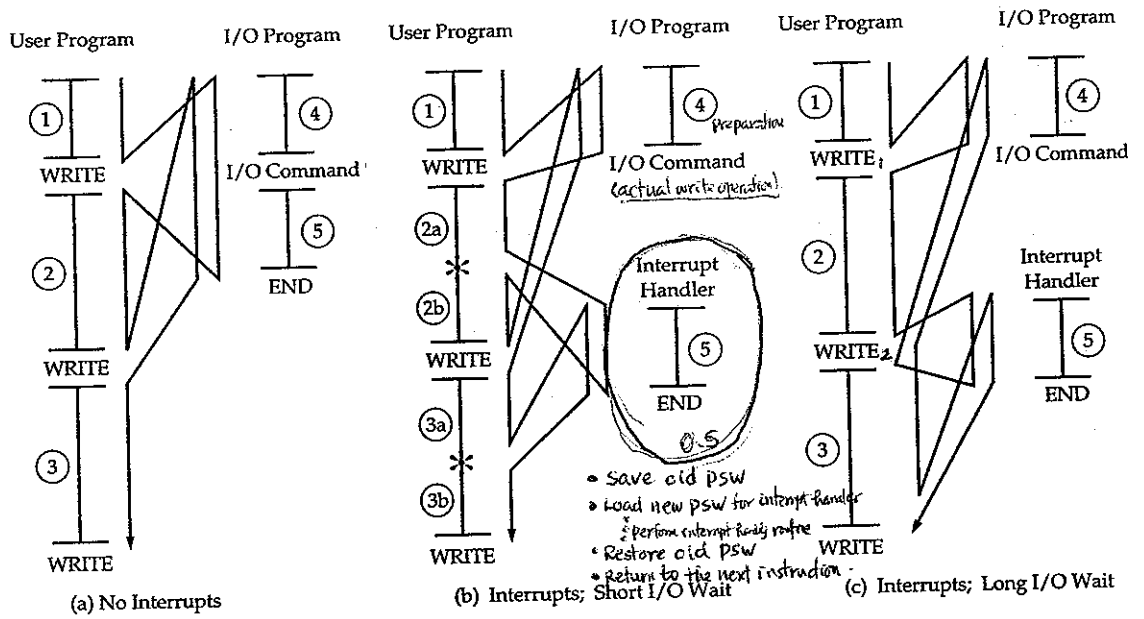


FIGURE 3.7. Program flow of control without and with interrupts

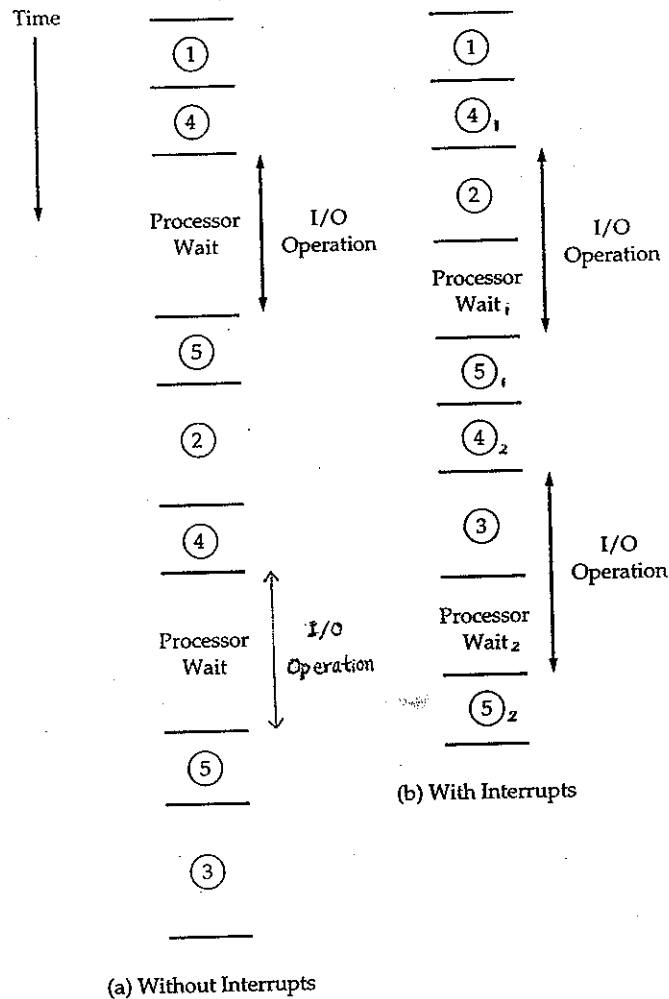


FIGURE 3.11. Program timing; long I/O wait

Interconnection Structure

bus: 50 ~ 100 lines

Ex. Load AC, A // load content of memory location A to AC //

- o Address bus: address of A
- o Data bus: content of A
- o Control bus: control signal for Load instruction

