Central Processing Unit (CPU)

- CPU performs the data processing operations in a computer.
- Three major parts (Fig. 8-1)

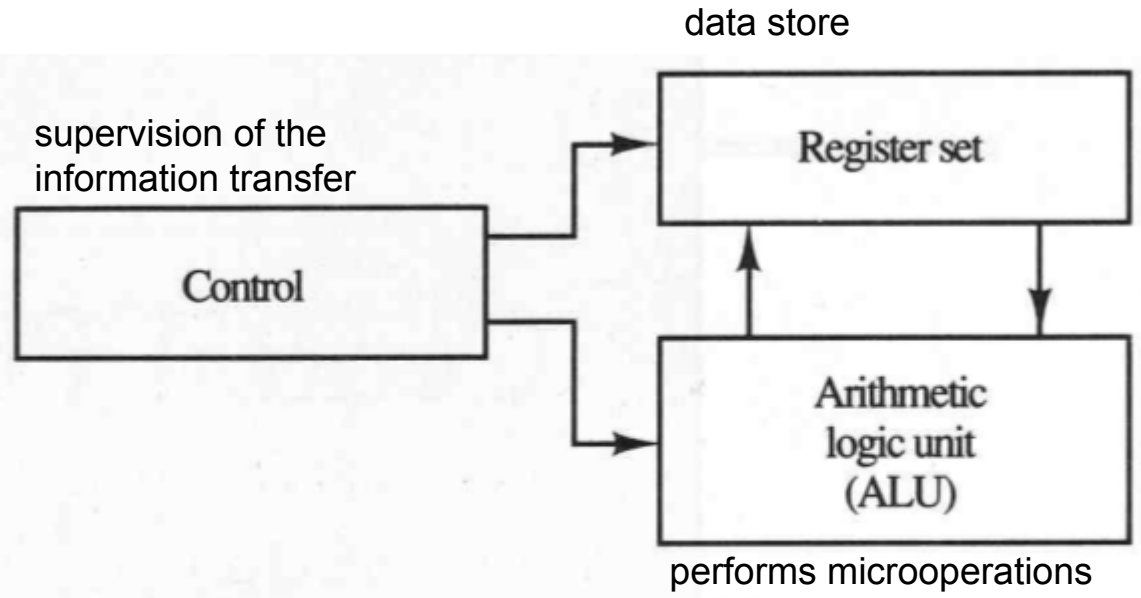
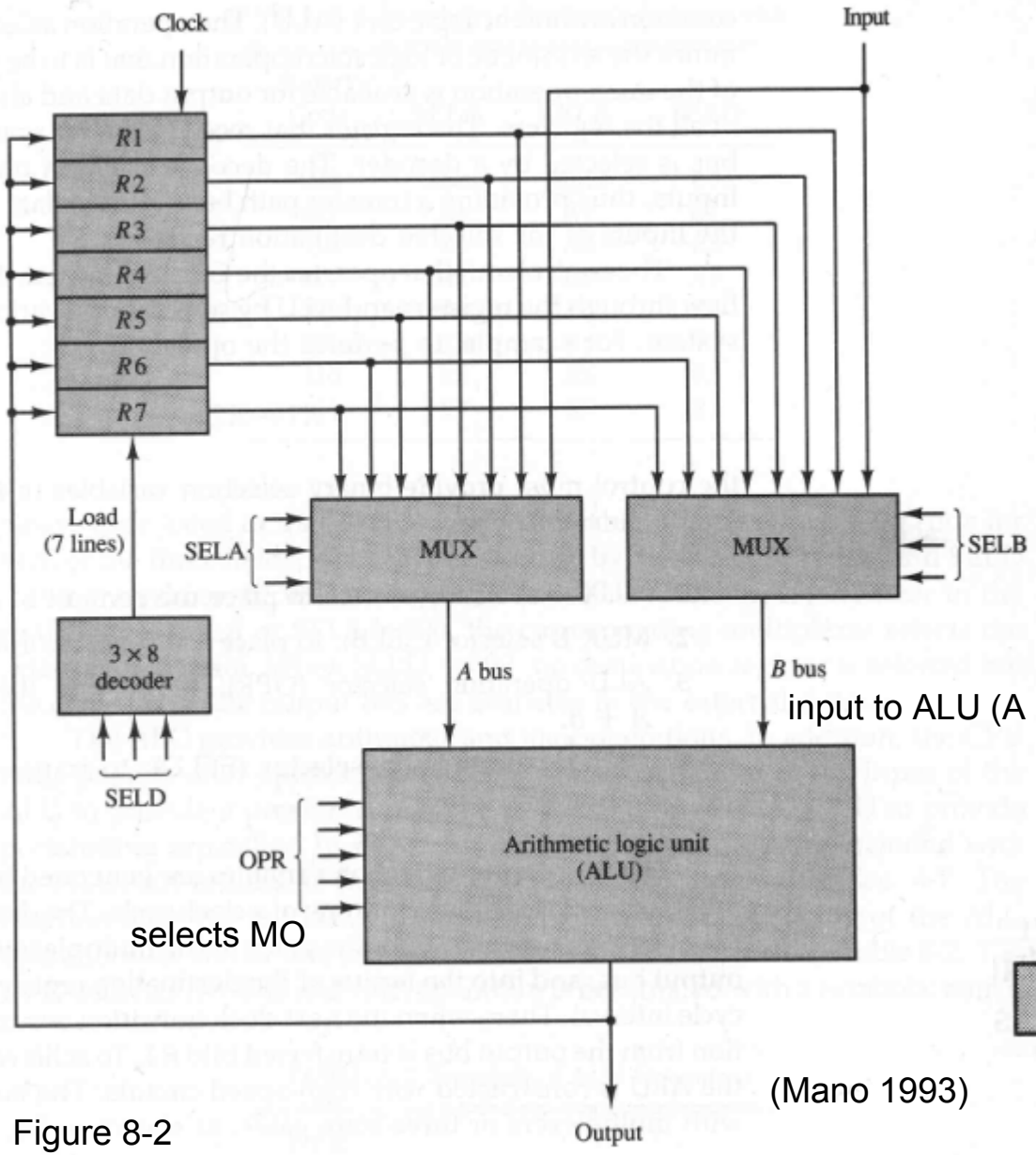


Figure 8-1 Major components of CPU.

- Instruction set provides the specification for the design of the CPU.
- The design of CPU involves choosing the hardware for implementing the machine instructions.
- A programmer using assembly language has to be aware of the register set, the memory structure, the type of data supported by the instructions, and the function of each instruction.

General Register Organization

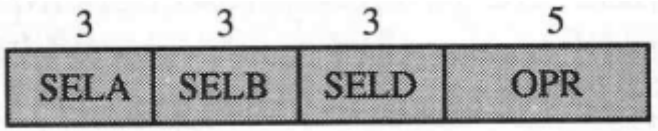
- Storing pointers, counters, return addresses, temporary results, and partial products into the memory is not efficient: referring to memory locations is time consuming: it is more efficient to use registers.
- If CPU has a large number of registers, a common bus is used to connect the registers.
- Arithmetic logic unit (ALU) is used to perform various microoperations.
- A bus organization of 7 CPU registers is shown in Fig. 8-2.



input to ALU (A bus and B bus)

selects MO

control word



(Mano 1993)

Figure 8-2

- Control unit operates the CPU bus system. For example:

  1. SELA is used to place R2 into bus A.
  2. SELB is used to place R3 into bus B.
  3. OPR selects the arithmetic addition.
  4. SELD is used to transfer the result into R1.

- The buses are implemented with multiplexers or 3-state gates.
- The state of 14 binary selection inputs specifies a control word.
- The 14-bit control word (when applied to the selection inputs specify a microoperation).
- The encoding of register selections in specified in Table 8-1.

TABLE 8-1 Encoding of Register Selection Fields

Binary Code	SELA	SELB	SELD
000	Input	Input	None
001	*R*1	*R*1	*R*1
010	*R*2	*R*2	*R*2
011	*R*3	*R*3	*R*3
100	*R*4	*R*4	*R*4
101	*R*5	*R*5	*R*5
110	*R*6	*R*6	*R*6
111	*R*7	*R*7	*R*7

Input = external input

if SELD = 000 then the content of the output bus is available in the external output

- ALU provides arithmetic and logic operations.
- Encoding of ALU operations (function table for this ALU is listed in Table 4-8):

TABLE 8-2 Encoding of ALU Operations

OPR Select	Operation	Symbol
00000	Transfer A	TSFA
00001	Increment A	INCA
00010	Add $A + B$	ADD
00101	Subtract $A - B$	SUB
00110	Decrement A	DECA
01000	AND A and B	AND
01010	OR A and B	OR
01100	XOR A and B	XOR
01110	Complement A	COMA
10000	Shift right A	SHRA
11000	Shift left A	SHLA

- Examples of microoperations and corresponding control words:

TABLE 8-3 Examples of Microoperations for the CPU

Microoperation	Symbolic Designation SELA	SELB	SELD	OPR	Control Word
$R1 \leftarrow R2 - R3$	R2	R3	R1	SUB	010 011 001 00101
$R4 \leftarrow R4 \vee R5$	R4	R5	R4	OR	100 101 100 01010
$R6 \leftarrow R6 + 1$	R6	—	R6	INCA	110 000 110 00001
$R7 \leftarrow R1$	R1	—	R7	TSFA	001 000 111 00000
Output $\leftarrow R2$	R2	—	None	TSFA	010 000 000 00000
Output $\leftarrow$ Input	Input	—	None	TSFA	000 000 000 00000
$R4 \leftarrow$ sh1 $R4$	R4	—	R4	SHLA	100 000 100 11000
$R5 \leftarrow 0$	R5	R5	R5	XOR	101 101 101 01100

000 (not used)

XOR(x,x) = 0

- The most efficient way to generate control words with a large number of bits is to store them in a memory unit, which is referred to as control memory (control store).
- By reading consecutive control words from memory, it is possible to initiate the desired sequence of microoperations for the CPU => microprogrammed control.