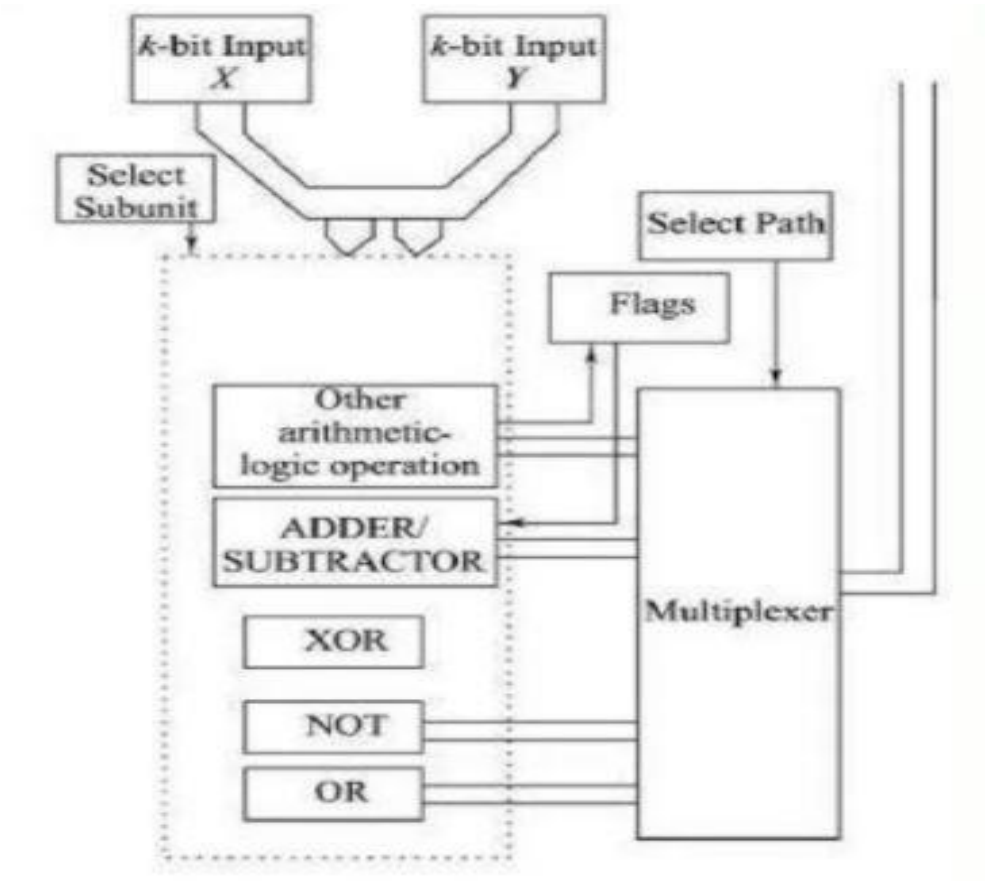
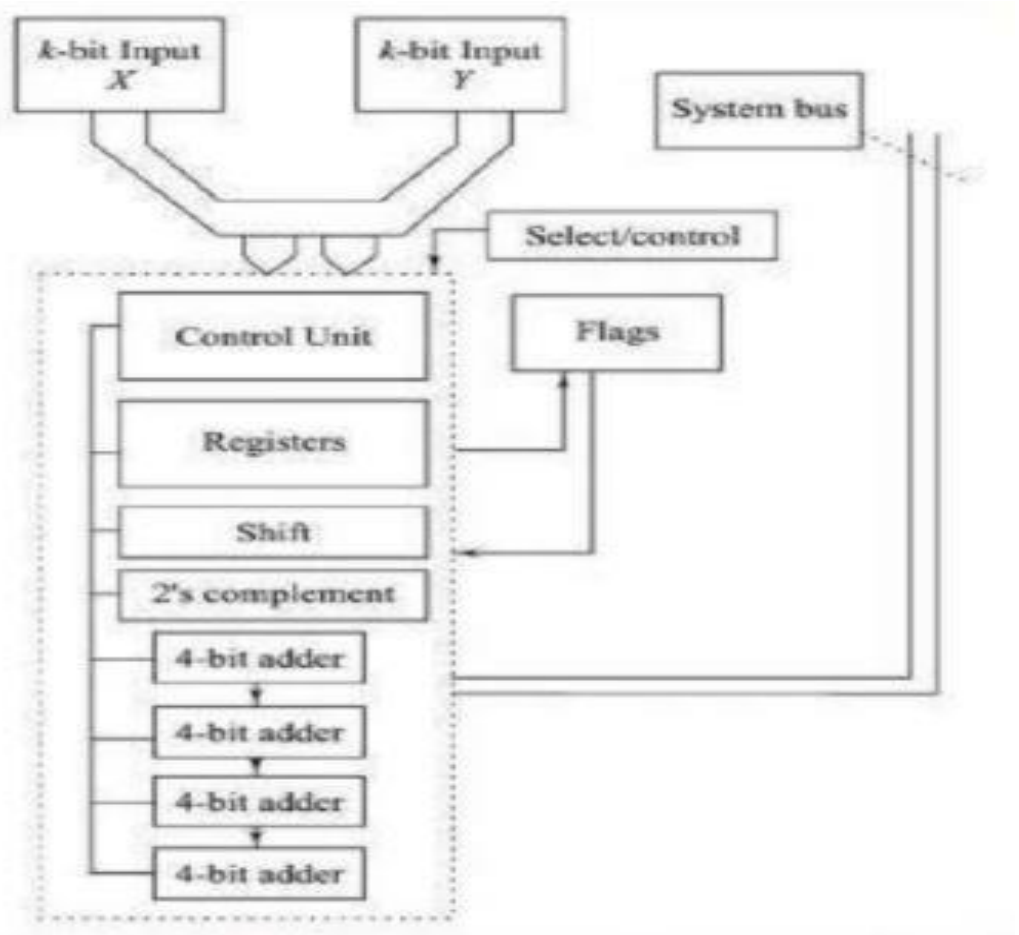
Fixed Point ALUs

- ALU is a complex digital circuit with an Arithmetic Unit (AU) and a Logic Unit (AU).
- It is integrated with the CPU.
- Sometimes the processor has more than one ALU; one for fixed point operation and other for floating point operation.
- Sometimes floating point operations are done by a separate Floating Point Unit (FPU).
- There are two types of digital logic circuits used in a ALU:
  - Combinational circuits: half adder/full adder, encoder/decoder, mux/de-mux
  - Sequential circuits: flip-flop, shift register, counter

Combinational Circuits based ALU



Sequential Circuits based ALU



Pipelining(more)

- It is already discussed that pipelining is an effective way of organizing concurrent activity in a computer system.
- **Pipeline performance:**
  - The potential increase in performance resulting from pipelining is proportional to the number of pipeline stages.
  - Performance would be increased if the there is no interrupt during the program execution. Unfortunately, this is not the case.
  - For various reason, a stage in pipeline can not be completed in the time slot allotted.

Pipelining(more)

- A simple measure of pipeline performance is determined as follows:

$$\tau = \max[\tau_i] + d = \tau_m + d \quad 1 \leq i \leq k$$

τ is the cycle time of a pipeline which is the time needed to advance a set of instructions one stage through the pipeline

where,

τ_i = time delay of the circuitry in the ith stage of the pipeline

τ_m = maximum stage delay (delay through stage which experience the largest delay)

k = number of the stages in the pipeline

d = delay of the latch, needed to advance signals and data from one stage to the next

Pipelining(more)

- Any condition that cause the pipeline stall is called a **hazard**.
- In the figure shown, instruction 2 (I_2) requires three clock cycles (4 to 6) to complete the E_2; so from cycle 5 to 6, the write stage has nothing to do but wait.
- Hence the pipeline operation has been **stalled** for two clock cycles.

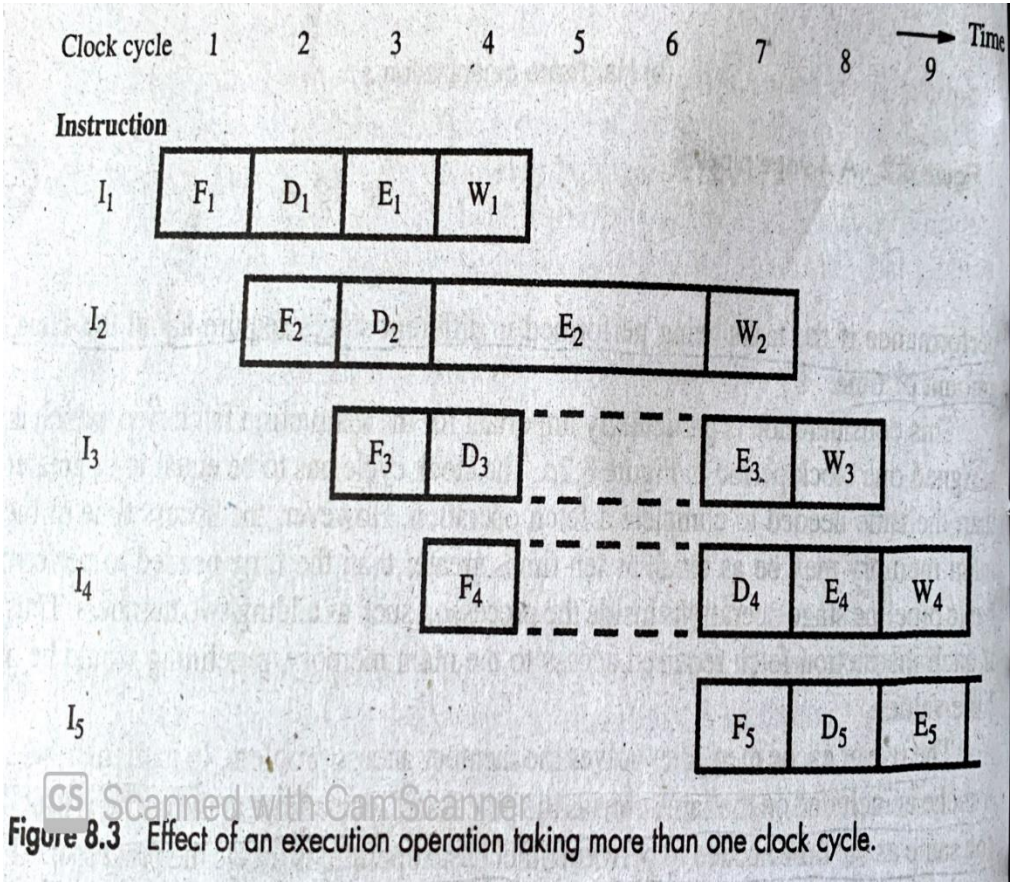


Figure 8.3 Effect of an execution operation taking more than one clock cycle.

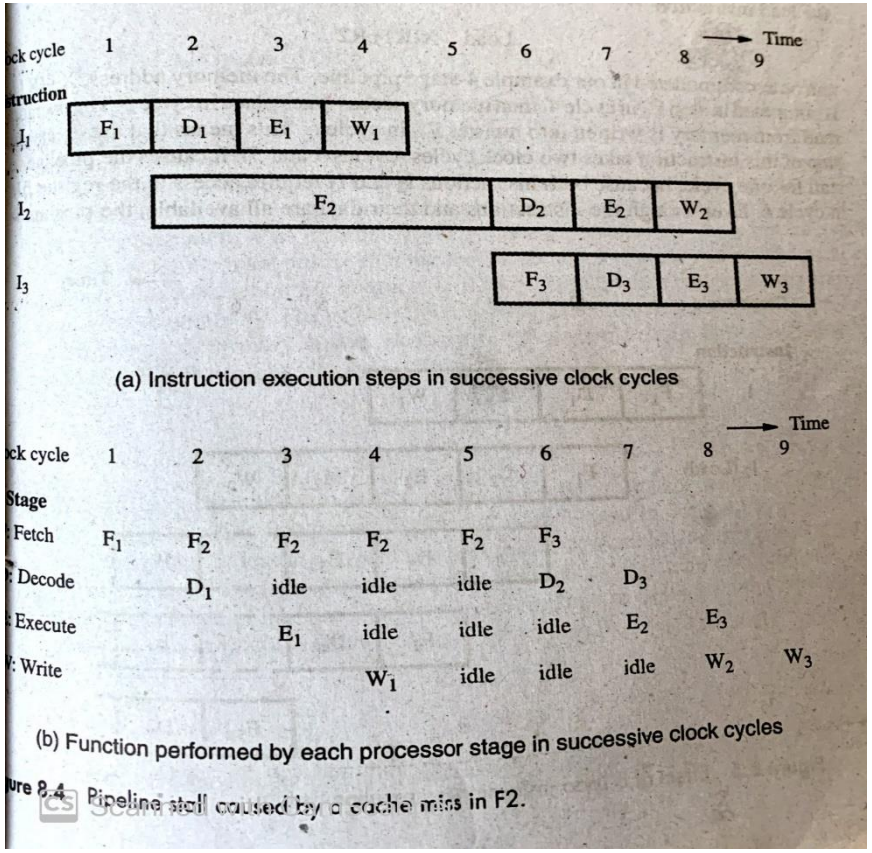
Pipelining(more)

- Pipeline hazard can be of three types: data hazard, instruction/control hazard and structural hazard.

  - A **data hazard** is any condition in which either the source or the destination operands of an instruction are not available at the time expected in the pipeline (Example: Figure 8.3).
  - The pipeline may also be stalled because of the delay in the availability of an instruction. Such hazards are known as **instruction/control hazards**. For example, delay in the availability of an instruction may be a result of a miss in the cache memory, requiring the instruction be fetched from the main memory.

Pipelining$_{(more)}$

- For example, as shown in the figure, instruction 1 (I_1) is fetched from the cache in cycle 1 and its execution proceeds normally. The fetch operation for instruction 2 (I_2) is started at cycle 2, which results in a cache miss. So the fetch stage must suspend any other fetch operation and waits (idle time) for the arrival of F_2.



Pipelining(more)

- **Structural hazard** is a situation when two instruction requires the use of a given hardware resource at the same time.
  - For example, memory access. One instruction requires to access the memory during execute or write stages while another instruction is being fetched from the memory.

    In this case, only one instruction would proceed and the other would be delayed.
  - Many processors use separate data caches and instructions to avoid this delay.
  - As shown in the figure, instructions 2 and 3 (I_2 and I_3) require the access of memory at the same time (cycle 6).

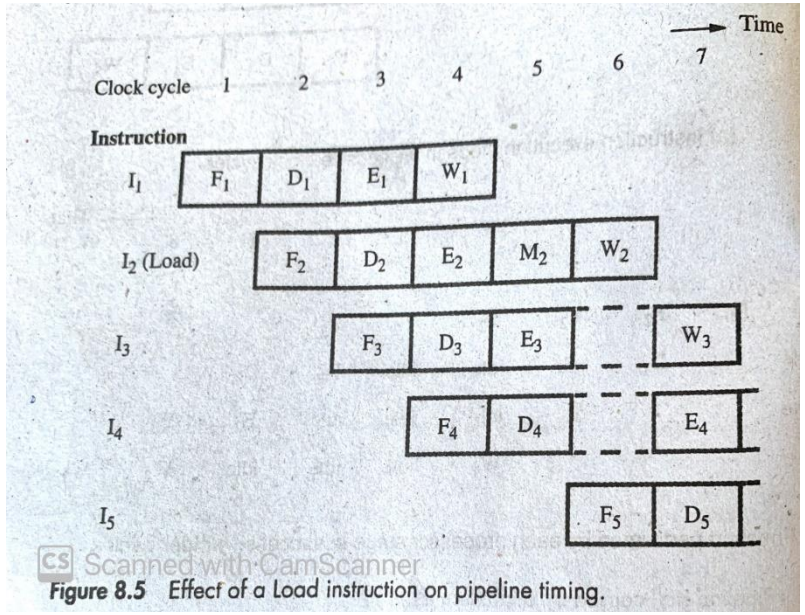


Figure 8.5 Effect of a Load instruction on pipeline timing.

See Stalling and Zaky (Chapter 4 and 8)

Systolic Array/Architecture

- It is a parallel processing approach that diverges from the traditional Von Neumann architecture.
- It is an example of pipeline processing.
- Systolic array consists of a network of interconnected identical processors or processing elements that compute and pass data through the system.
- In systolic system, data flows from the memory in a rhythmic fashion, passing through many processing elements before return to memory.
- Follows the same process of flow of blood through heart; thus called systolic.

Systolic Array/Architecture (contd.)

- Each processing elements perform one operation.
- Speeds up the system.
- Application: matrix multiplication, correlation, DNA and protein sequence analysis etc.

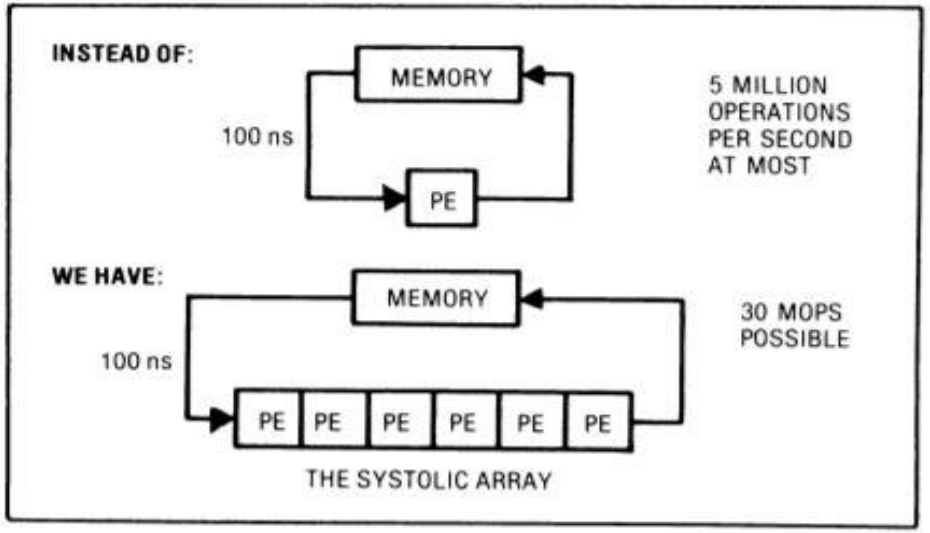


Figure: Basic principle of Systolic array

Systolic Array/Architecture (contd.)

- Data flows from left and top, and pass them to the right and bottom PEs.

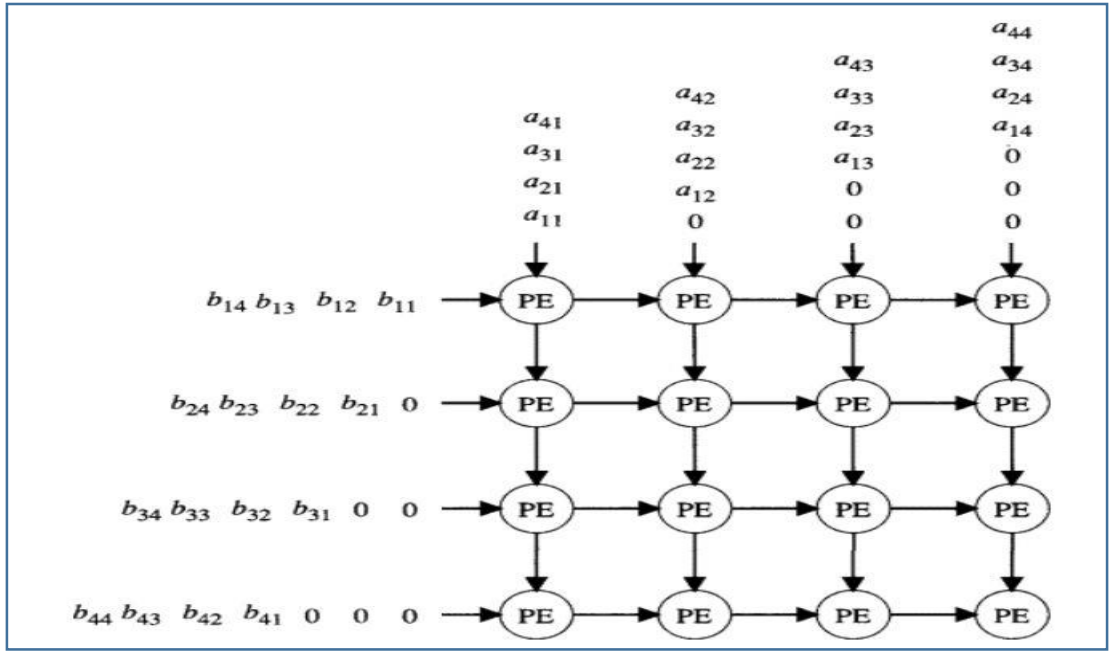


Figure: A 4x4 matrix multiplication