Advanced Computer Architecture (0630561)

Lecture 10

Hardware and Software Parallelism

Prof. Kasim M. Al-Aubidy

Computer Eng. Dept.

Introduction:

- Modern computer architecture implementation requires special hardware and software support. This includes;
  - Distinguish between hardware and software parallelism.
  - Mismatch problem between hardware and software.
  - Fundamental concept of compilation support needed to close the gap between hardware and software.

Hardware Parallelism:

- This refers to the type of parallelism defined by the machine architecture and hardware multiplicity.
- Hardware parallelism is a function of cost and performance tradeoffs. It displays the resource utilization patterns of simultaneously executable operations. It can also indicate the peak performance of the processors.
- One way to characterize the parallelism in a processor is by the number of instruction issues per machine cycle.
- If a processor issues k instructions per machine cycle, then it is called a **k-issue processor**.
- In a modern processor, two or more instructions can be issued per machine cycle.
- A conventional processor takes one or more machine cycles to issue a single instruction. These types of processors are called **one-issue machines**, with a single instruction pipeline in the processor.
- A multiprocessor system which built n k-issue processors should be able to handle a maximum of nk threads of instructions simultaneously.

Example Detection of parallelism in a program

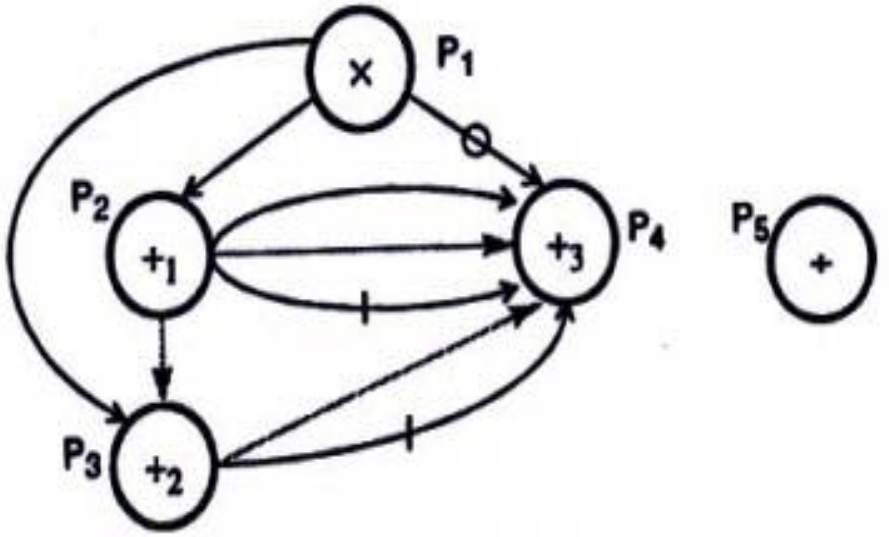
Consider the simple case in which each process is a single HLL statement. We want to detect the parallelism embedded in the following instructions P_1, P_2, P_3, P_4, and P_5.

$$
\begin{aligned}
P_1 &: \quad C = D \times E \\
P_2 &: \quad M = G + C \\
P_3 &: \quad A = B + C \\
P_4 &: \quad C = L + M \\
P_5 &: \quad F = G \div E
\end{aligned}
$$

Assume that each statement requires one step to execute.

No pipelining is considered here.

The dependence graph is →

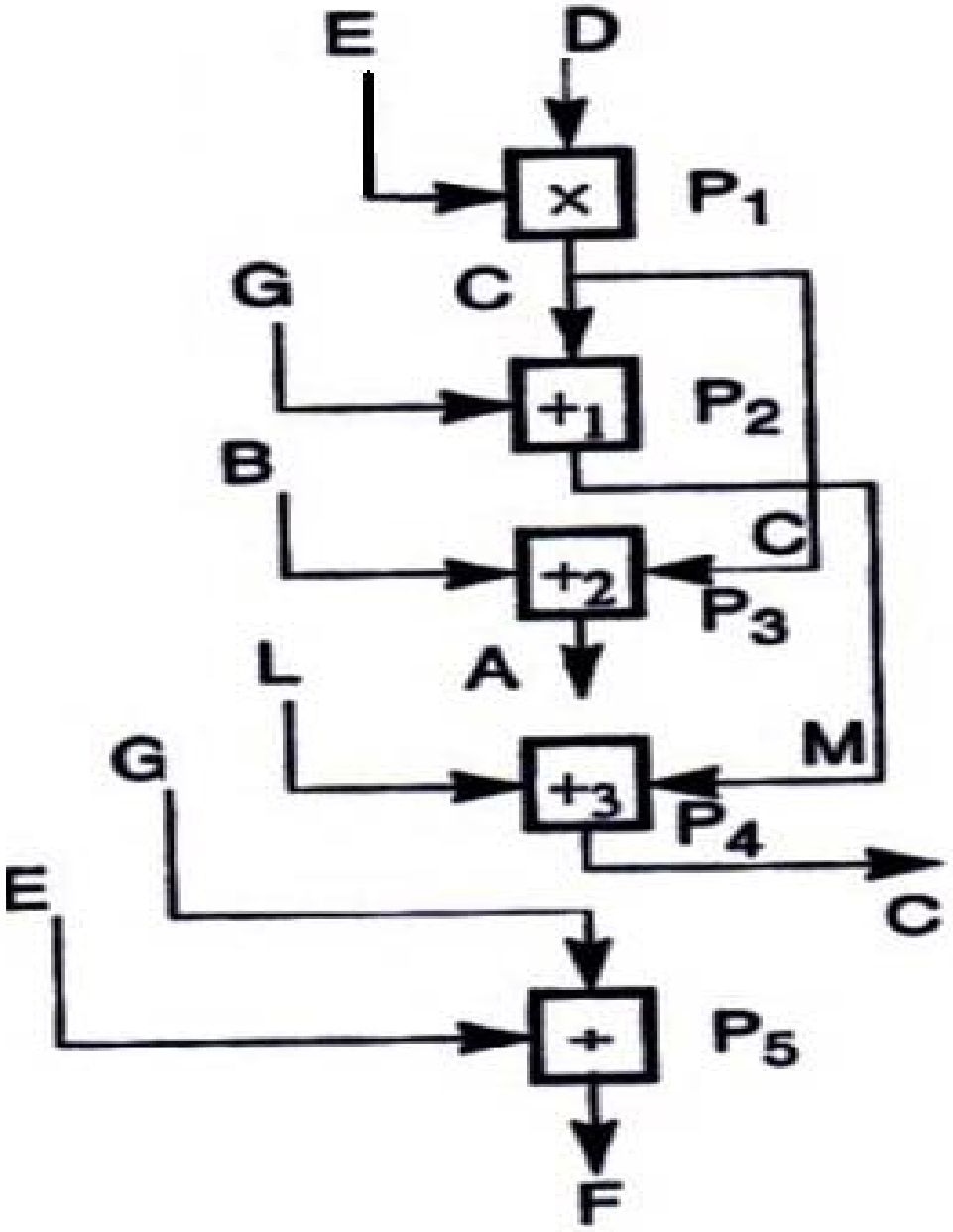


A dependence graph showing both data dependence (solid arrows) and resource dependence (dashed arrows)

Sequential Execution:

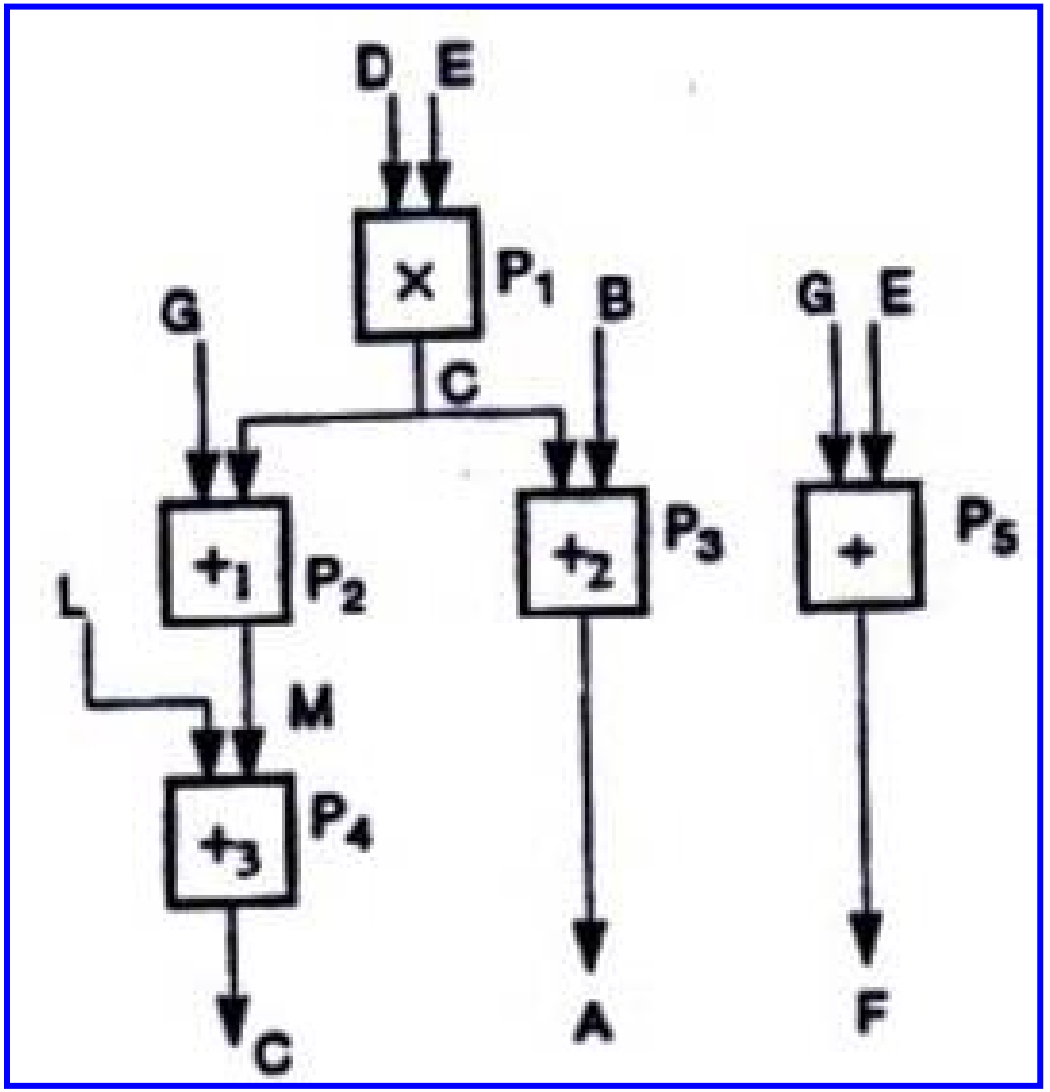
$$
\begin{array}{lllll}
P_1: & C & = & D \times E \\
P_2: & M & = & G + C \\
P_3: & A & = & B + C \\
P_4: & C & = & L + M \\
P_5: & F & = & G \div E
\end{array}
$$

Sequential execution in five steps, assuming one step per statement (no pipelining)



Parallel Execution:

$$
\begin{array}{lllll}
P_1: & C & = & D \times E \\
P_2: & M & = & G + C \\
P_3: & A & = & B + C \\
P_4: & C & = & L + M \\
P_5: & F & = & G \div E
\end{array}
$$



- Parallel execution in THREE steps, assuming TWO adders are available per step.
- If TWO adders are available, the parallel execution requires only THREE steps.
- Only 5 pairs can execute in parallel, if there is no resource conflicts.
- In this example, as shown in the fig, only P2 ‖ P3 ‖ P5 is possible.

Software Parallelism:

- It is defined by the control and data dependence of programs.
- The degree of parallelism is revealed in the program profile or in the program flow graph.
- Software parallelism is a function of algorithm, programming style, and compiler optimization.
- The program flow graph displays the patterns of simultaneously executable operations.
- Parallelism in a program varies during the execution period. It limits the sustained performance of the processor.

Mismatch between S/W & H/W Parallelism:

Example:

$A = L_1 * L_2 + L_3 * L_4$

$B = L_1 * L_2 - L_3 * L_4$

Software Parallelism:

There are 8 instructions;

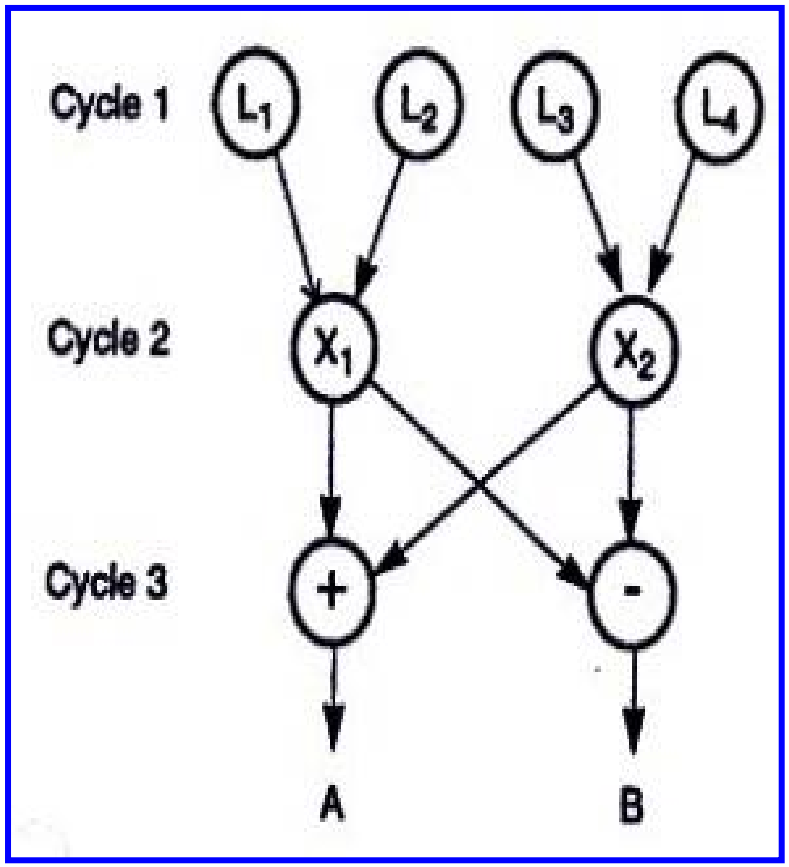
FOUR Load instructions (L1, L2, L3 & L4).

TWO Multiply instructions (X1 & X2).

ONE Add instruction (+)

ONE Subtract instruction (-)

- The parallelism varies from 4 to 2 in three cycles.

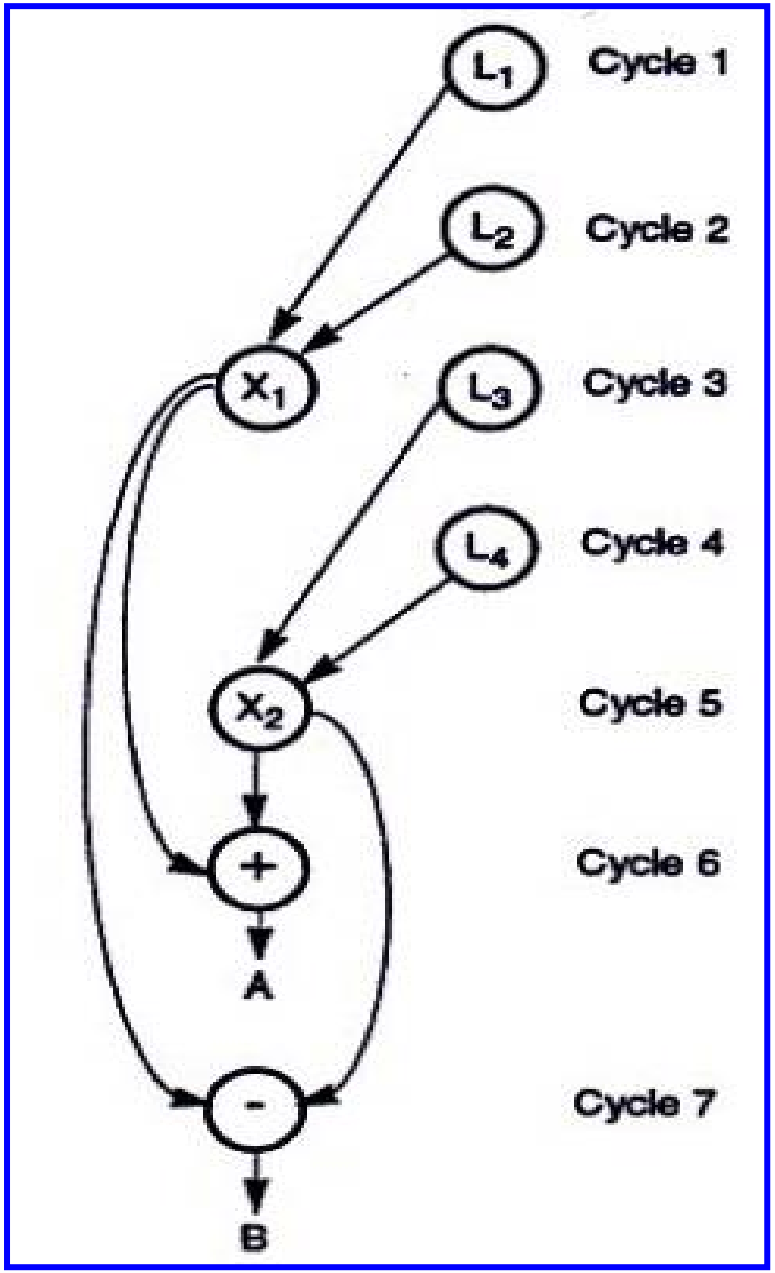


$$\text{Average S/W Parallelism} = \frac{8 \text{ cycles}}{3 \text{ cycles}} = \frac{8}{3} = 2.67$$

Hardware Parallelism:

Parallel Execution:

- Using TWO-issue processor:
- The processor can execute one memory access (Load or Store) and one arithmetic operation (multiply, add, subtract) simultaneously.
- The program must execute in 7 cycles.
- The h/w parallelism average is 8/7=1.14.
- It is clear from this example the mismatch between the s/w and h/w parallelism.



Example:

- A h/w platform of a Dual-Processor system, single issue processors are used to execute the same program.
- Six processor cycles are needed to execute the 12 instructions by two processors.
- S1 & S2 are two inserted store operations.
- L5 & L6 are two inserted load operations.
- The added instructions are needed for inter-processor communication through the shared memory.