Lecture TWO (not complete)
Parallel Computers

1. Synchronous
   - Vector-Array
   - SIMD
   - Systolic

2. Asynchronous
   - MIMD
Parallel–computers contain more than 1 processor

Why there is a need for parallel computers ?

• “Speed of light” problem for the chip designers. The speed of the circuits (hardware–chip) can not be increased indefinitely.

• Speed up execution of programs

• Deal with large amount of data
  (Large amount of data ⇒ More execution time)
Applications of Parallel Computers

- Image Processing
  (Large amount of data, for example:
  1024 × 1024 × 8 × 1 \(PIXELS\) : Monochrome
  1024 × 1024 × 8 × 3 \(PIXELS\) : Color

- Weather Prediction
  (Apply Partial Differential Equations in 3–D space, uses statistical – historic data)
• Robotics
  (Develop machines to behave like humans – Vision Image Processing)

• Simulation of Physical Phenomena
  (Molecule interaction–large amount of data due to the large amount of molecules)

• Aircraft testing
  (Discretize surface – Large amount of data)

• Real Time Speech Processing
  (Time Consuming Operations)

• Artificial Intelligence
  (Create trees and perform a lot of searching. Do it in Parallel)
• Simulate evolution of galaxies  
  (Has been done)
• Data Bases  
  (Search = most critical part – In order to delete a record, first find it)
• Stock market  
  (Prediction and research)
• Air line reservations  
  (Searching)
• Bioinformatics (DNA Processing and Analysis)
Goal of the U.S government

High performance directive from White House
To achieve 12.3 Teraflop performance by 2002

\[ 12.3 \, TFLOP = 12.3 \times 10^{12} \, FLOPS \]

(FLOP = FLOating Point operations per second)

SPARC \approx 50 \, MFLOPS or \( 5 \times 10^7 \) \, FLOPS

Therefore there is a big distance \( 10^7 - 10^{12} \).
The performance has to be sustained ALL the time and not in peaks.
Today ++

- 2005: 360 TFLOPS, “BlueGene/Lite”, IBM–USA (will)
Important Questions

1. How do you put the processors together?
   (Need coordination or synchronization)

2. Do you have laeders or all PE’s do the same thing?
   (Parallel system behaves like a society)

Let us give a refined definition of Parallel Computers – Processing
Parallel Computer

**Definition:**
“A Parallel Computer is a collection (connection) of processing units, usually of the same type, that communicate and coordinate their activities to solve large problems fast”

**Questions about a new system**

1. How the PE are Connected?
2. How the PE Communicate?
3. How the PE are Coordinated?
4. How the PE can solve large problems fast?
1. A Collection of Processing Elements

- How many?
  - For a small degree of parallelism a number of 10 P.E is fine (application oriented).
  - Use ALU’s = primitive, in a chip to perform in parallel, ADD, SUB,...., logic operations.
  - Use CPU’s for large volume system.

- How powerfull is each P.E.?
  Use ALU’s or CPU’s
• What operations do they perform?
  To perform smoothing in image processing use an ALU, because you just need add’s and sub’s.

• How much memory do we need?
  If I/O (secondary storage) is “stupid” = disk, you don’t need a parallel system. Only need large amount of memory (RAM).

• How much secondary (disk) storage do they need? do they perform I/O access?
  (A very fast car with a gas tank for only 5 miles is not a good design)
2. That can Communicate

- How do they communicate?
  In a company a worker can’t go to the president before he talks to the supervisor or other people. Therefore an order has to be in place. Therefore there is a need to establish a protocol.

- What sort of interconnection network will they need?
  (If we do not need a highway why build it?)
  If all the P.E are connected with all the other P.E’s, the performance is excellent but the cost is high. Therefore may not be possible to build the system.
3. And Coordinate

- How do they coordinate or synchronize their efforts?
  (A supervisor tells to the workers that they meet every other day to discuss various aspects of the design. If the worker meets with the supervisor every year, the meetings and the project will fail)
  (If you have to do a large programming project, give parts of the program to many programmers to do the job faster, under your supervision)
In parallel computing we need many “systems” to exchange data and to communicate with a coordinator.
To synchronize the operation, there is a need for a common clock.
• How large should each processor task be?
  (Give all the tasks to one P.E or assign tasks in pieces to different processors)
• How autonomous can each processor be?
  (Ask the permission of a controller to do something by itself)
• How will the operating system coordinate their efforts?
  (Uses UNIX for every processor to monitor the execution of the program)
• Which problems (applications) are amendable to parallel processing?
• What computational model will be used?
• How general-purpose versus special purpose our approach be?
• Which is the best algorithm for that problem?
• What is the programmers view on the machine?
  (sequential trained programmers are biased)
  (Today NEW programmers, AT LEAST, should think about a parallel solution)
• Reliability
  (Should know the probability of failure for the system from the hardware point of view)
• Serviability
  (Build a system that finds the fault and isolates the fault processor and also gives you a message to replace the faulty component with a new one).
  (It is not smart to stop the system to repair it)

“A System with no reliability and serviability is not a good design”
4. To Solve Large Problems Fast

- Can I use parallel processing for everything?

Example 1: Given the code:

```plaintext
FOR i = 1 to 5 DO
    A[i] = A[i-1] + 1
```

The sequential code can not be parallelized because the current computation $A[i-1]$ depends on the result of the previous.

Example 2: Given the code:

\[
\text{FOR } \ i = 1 \ \text{ to } 5 \ \text{ DO} \\
\]

The sequential code can be parallelized

\[
\]