Komputasi Paralel

Kudang B. Seminar

Example 1: Weather Prediction

- Area, segments
  - 3000*3000*11 cubic miles
  - .1*.1*.1 cubic mile: ~ 10^{11} segments
- Two day prediction
  - half hour periods: ~ 100 periods
- Computation per segment
  - Temp, Pressure, Humidity, Wind speed, Wind direction
  - Assume ~ 100 FLOPs

Performance: Weather Prediction

- Computational requirement: 10^{15}
- Serial supercomputer: 10^{9} instr/sec
- Total serial time: 10^{6} sec = 280 hours
- Not too good for 48 hour weather prediction

Parallel Weather Prediction

- 1 K workstations, grid connected
  - 10^{8} segment computations per processor
  - 10^{8} instructions per second
  - 100 instructions per segment computation
  - 100 time steps: 10^{4} seconds = ~3 hours
    - Much more acceptable
  - Assumption: Communication not a problem here
- More workstations:
  - finer grid
  - better accuracy

Now you need to program it!

Parallel programming introduces:

- Task partitioning, task scheduling
- Data partitioning
- Synchronization
- Load balancing
- Latency issues
  - hiding
  - tolerance

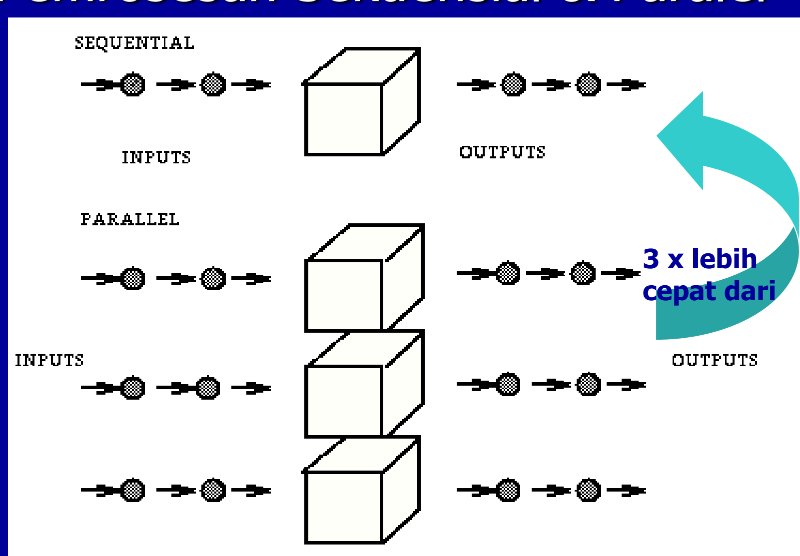
Sequential vs Parallel Algorithms

- Efficient Parallel Algorithms
  - Maximize parallelism
  - Minimize synchronization, remote accesses
  - Efficiency is Architecture Dependent
- Efficient Sequential Algorithms
  - Minimize time, space
  - Efficiency is portable
    - Efficient C program on Pentium ~ Efficient C program on Alpha

Speedup

- Ideal: n processors → n fold speed up
  - Ideal not always possible. WHY?
  - Tasks are data dependent
  - Not all processors are always busy
  - Remote data
- Super linear speedup: >n speedup
  - Nonsense! Because we can execute the faster parallel program sequentially
  - No nonsense!! Because parallel computers do not just have more processors, they have more caches

Pemrosesan Sekuensial & Paralel

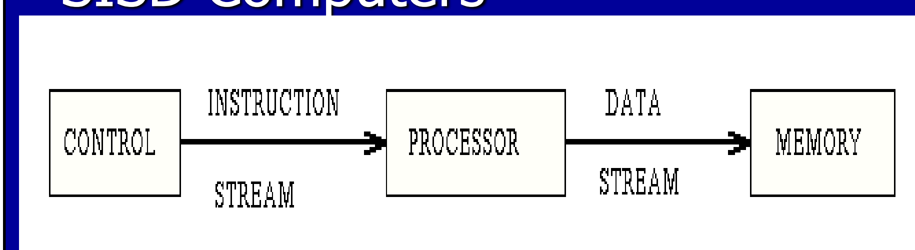


Klasifikasi Mesin Paralel

Models of Computation (Flynn 1966)

1. **Single Instruction Stream, Single Data Stream : SISD.**
2. **Multiple Instruction Stream, Single Data Stream : MISD.**
3. **Single Instruction Stream, Multiple Data Stream : SIMD.**
4. **Multiple Instruction Stream, Multiple Data Stream : MIMD.**
5. **Single Program Multiple Data: SPMD.**

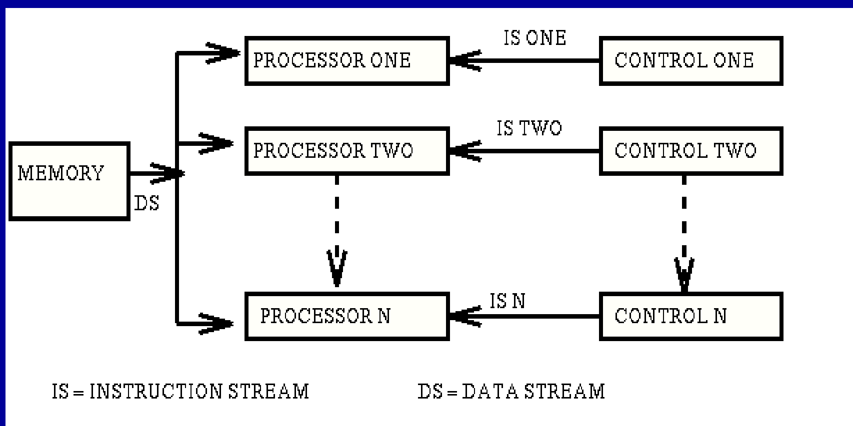
SISD Computers



Untuk operasi $a_1 + a_2 + a_3 + \ldots + a_n$ memerlukan sebanyak n akses ke memori oleh prosesor dan sebanyak $n-1$ operasi penjumlahan. Jadi kompleksitas waktu operasi adalah $O(n)$.

MISD Computers

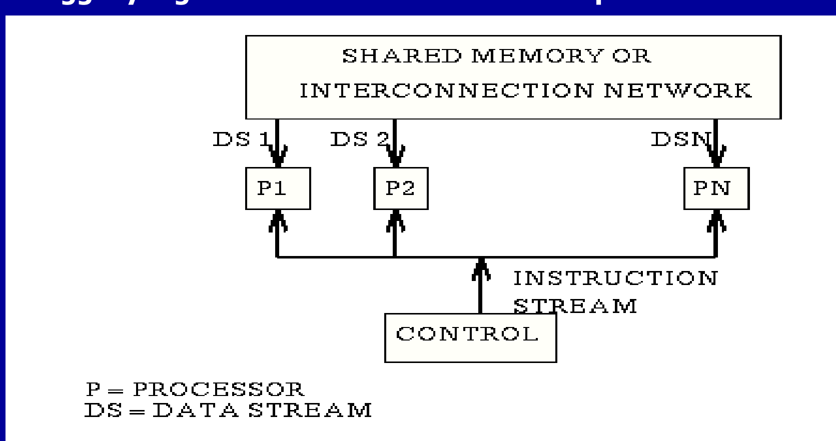
N prosesor yang memiliki unit kontrol pribadi, berbagi guna memori bersama (*shared memori*).



Parallelisme diperoleh dengan menugaskan semua prosesor mengerjakan operasi/tugas yang berbeda secara simultan pada data yang sama.

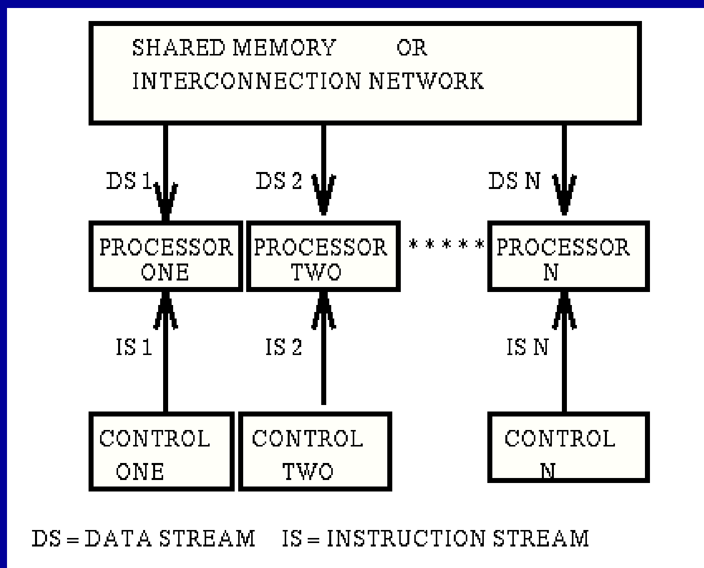
SIMD Computers

N prosesor beroperasi dibawah kendali aliran instruksi tunggal yang dikeluarkan oleh unit kontrol pusat.

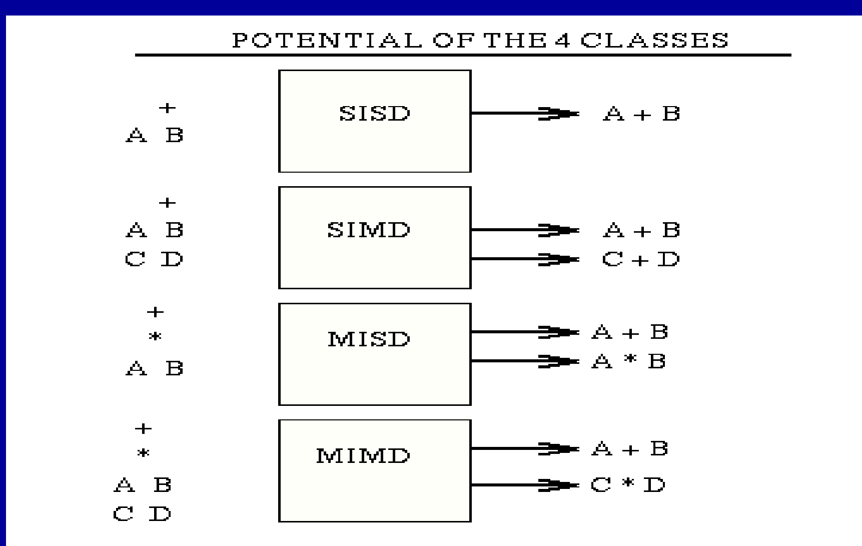


The processors operate **synchronously** and a **global clock** is used to ensure **lockstep** operation.

MIMD Computers



Potensi dari 4 kelas komputer



SPMD Computers

Program yang sama dieksekusi pada prosesor komputer MIMD.

SPMD bukan merupakan paradigma hardware, ini adalah software ekuivalen dari SIMD, namun bersifat *asynchronous*.

Perhatikan instruksi IF X = 0 THEN S1 ELSE S2

Asumsikan X = 0 pada prosesor P1, dan untuk X != 0 pada prosesor P2

Proses P1 mengeksekusi S1 paralel dengan prosesor P2 **mengeksekusi S2 (ini tidak dapat terjadi pada SIMD)**