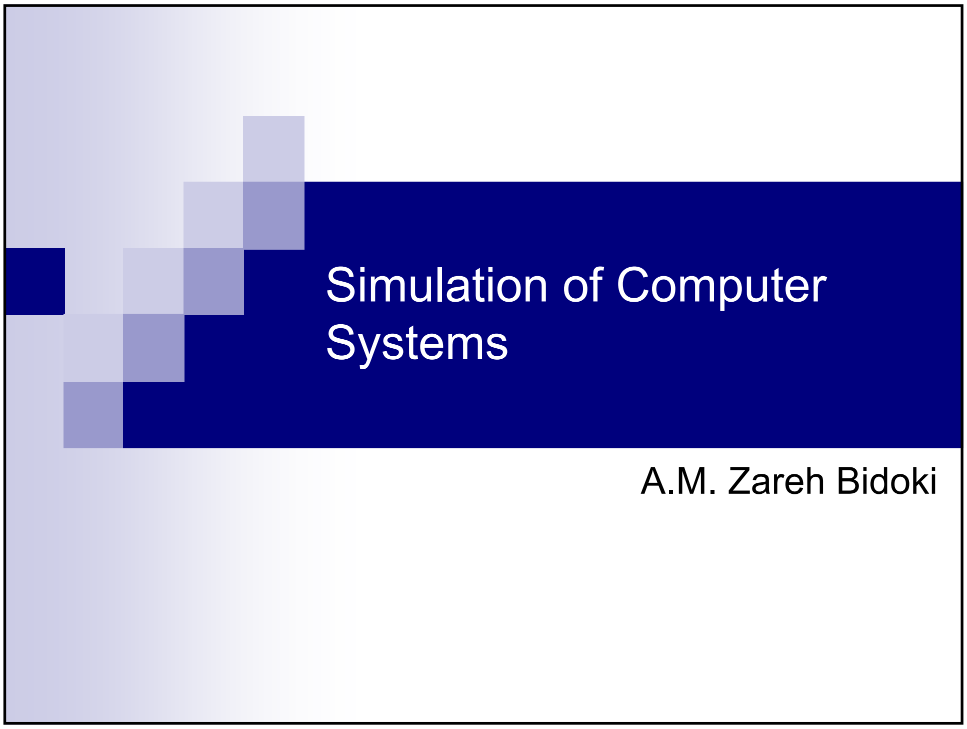
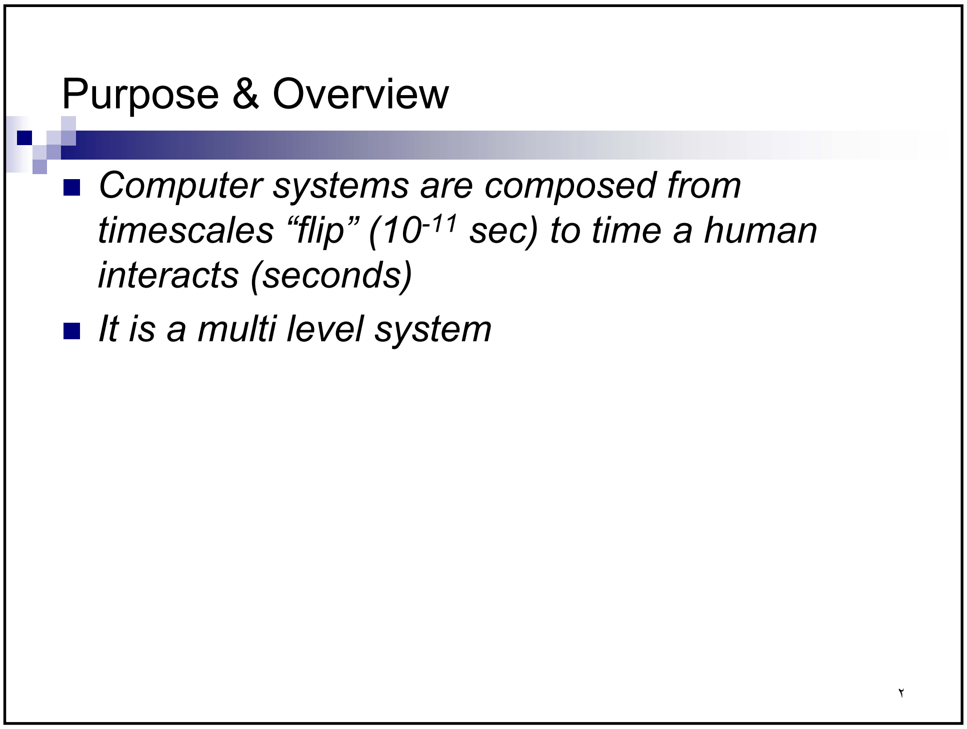


Simulation of Computer Systems

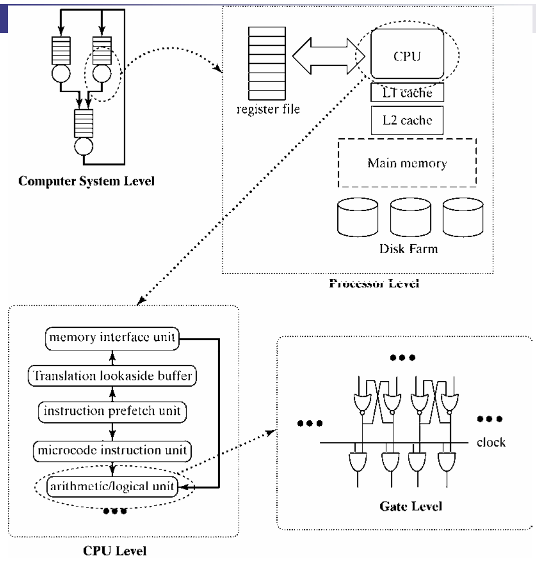
A.M. Zareh Bidoki



Purpose & Overview

- *Computer systems are composed from timescales “flip” (10^{-11} sec) to time a human interacts (seconds)*
- *It is a multi level system*

Different Level abstractions



Gate Level

- Clock (Delay)
- Test Vectors (Boundary scan, BIST)
  - □ Evaluate response of the circuit
  - □ Find problems like hazards
- Number of gates
- Number of Pins

Functional Abstraction

- RTL (Register Transfer Language)
- For example Memory (An indexed array)
  - □ R3=m[R6]
  - □ R3=R3-1
  - □ R6=R6+1
  - □ M[R6]=R3
- Time is result of gate level

I/O System Behavior

- Execution of computer program
  - □ The program execution should be modeled
  - □ Markov Chains are used for modeling inputs
- The program execution is modeled with randomly sampled CPU and I/O service time

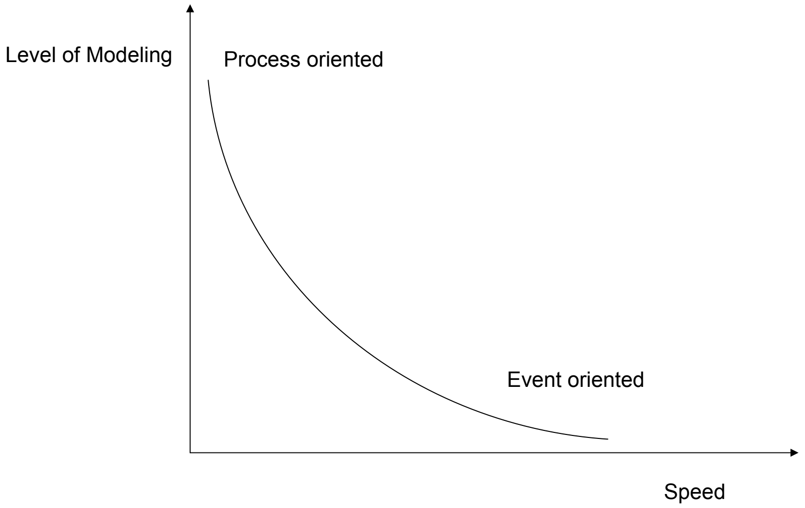
Simulation Tools

- Different simulation tools exist for each level
- VHDL
  - □ AT low level of abstraction
  - □ Modular design
  - □ Process based or event based?
- SMPL, CSIM,…

Process and Event Oriented Simulation

- DES
  - □ Trace Driven
  - □ Event Based
  - □ Process based
    - It is like OS environment
      - □ Resource sharing
      - □ Mutual exclusion
      - □ Semaphore
      - □ Process communication

Process and Event Oriented Simulation



Model Input

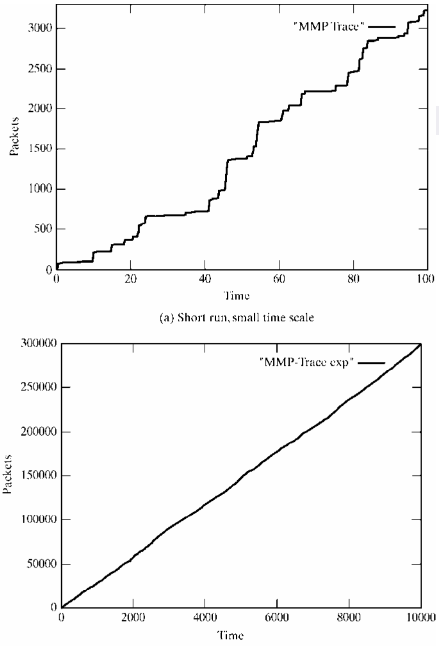
- CPU (Instructions)
- Memory (References)
- Gate (signals)

Modulated Poisson Process

- Some time input rate is burstiness (Traffic is much higher than normal)
- Modeling this state mathematically is called MPP
- The underlying framework is a continuous Markov chain (CTMC)

١١

Generating MPP trace



(a) Short run, small time scale

(b) Long run, large time scale

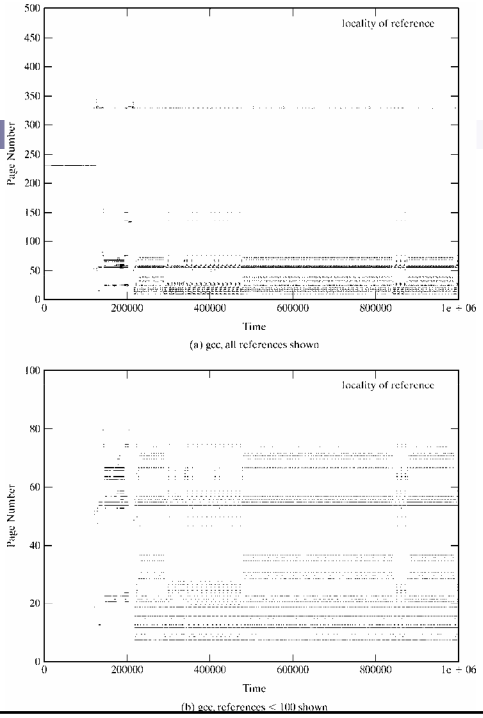
١٢

Virtual Memory Referencing

- Why VM is good?
- Program is organized on units called pages
- Physical memory is divided into page frames
- Mapping is done by OS
- Page fault?
- Replacement policy (hit ratio)
  - Use simulation to find hit ratio fro some polices
- Why VM work well?
  - Working set (Finding them is OS challenges)

١٣

Working set is line



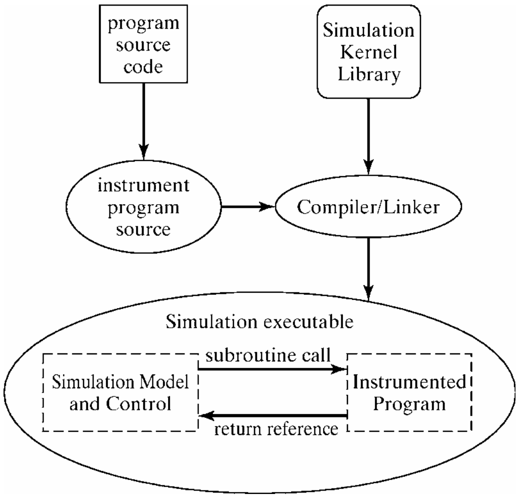
(a) gcc, all references shown

(b) gcc, references < 100 shown

١٤

Generating reference trace

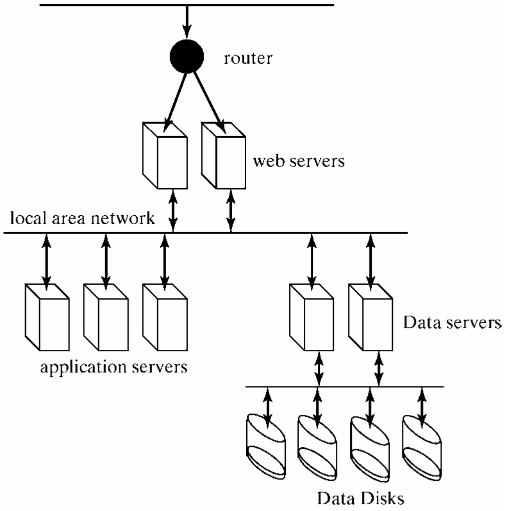
- Stochastically ?
- Direct execution



High Level Computer Simulation

- A good response time
- We should find bottleneck (delay)

Web site Server System



Web site Server System

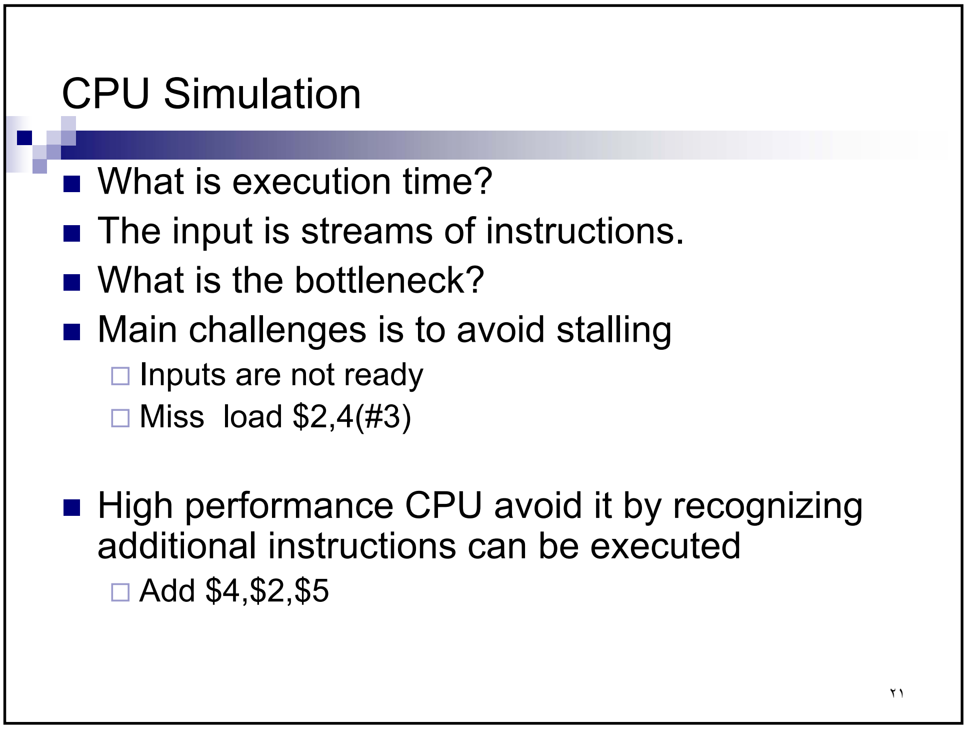
- Router have table of sessions
- Web server has three queues of threads
- Application server has two queues of threads
- Goal is to find response time distribution
- First we find bottleneck and then look how to reduce load at bottleneck during change of scheduling policy, biding applications to servers, increasing CPU and I/O devices

Web site Server System parameters

Subsystem	Specification
Router	Load balancing policy, execution tomes
Web server	Server count, queuing policy, execution tomes
Application Server	Server count, queuing policy
Data Server	Server count, Disk count, queuing policy, Disk time

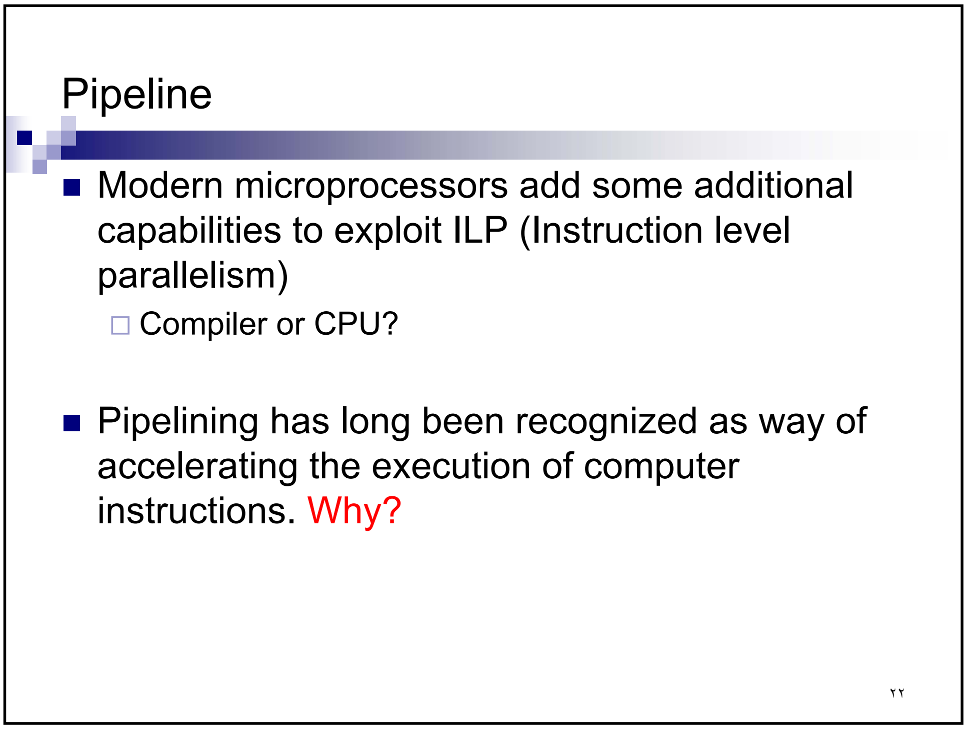
Process or event based

- The website model is an excellent candidate for process oriented approach!!
- How can we model with event based simulation?
- Event based is focused on queries.
- Process based is focused on servers.



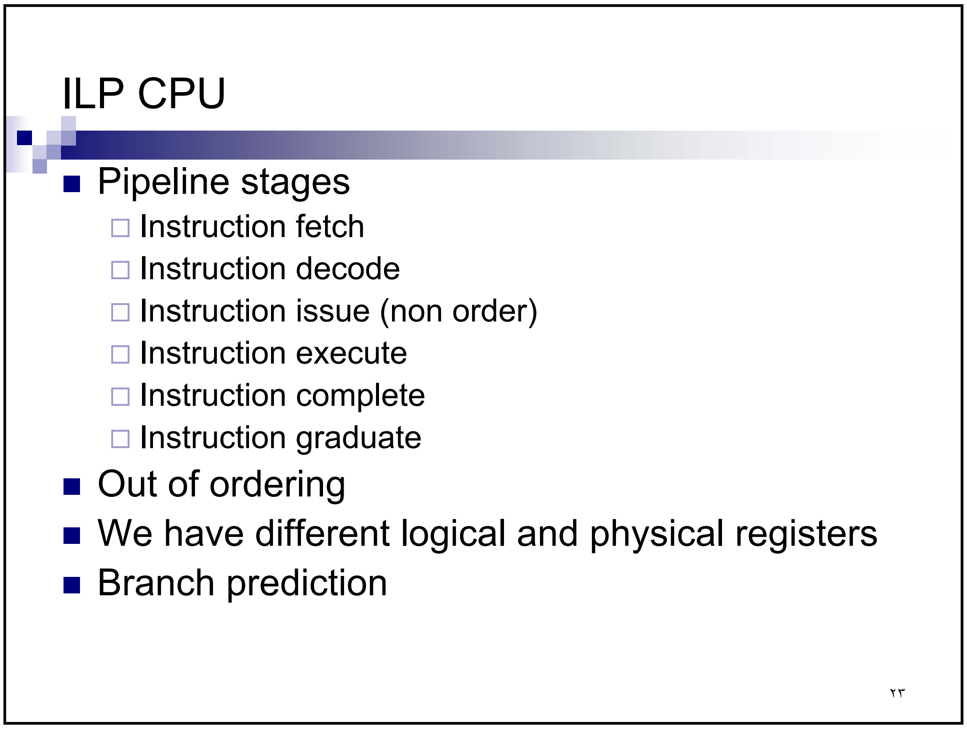
CPU Simulation

- What is execution time?
- The input is streams of instructions.
- What is the bottleneck?
- Main challenges is to avoid stalling
  - □ Inputs are not ready
  - □ Miss load $2,4(#3)
- High performance CPU avoid it by recognizing additional instructions can be executed
  - □ Add $4,$2,$5



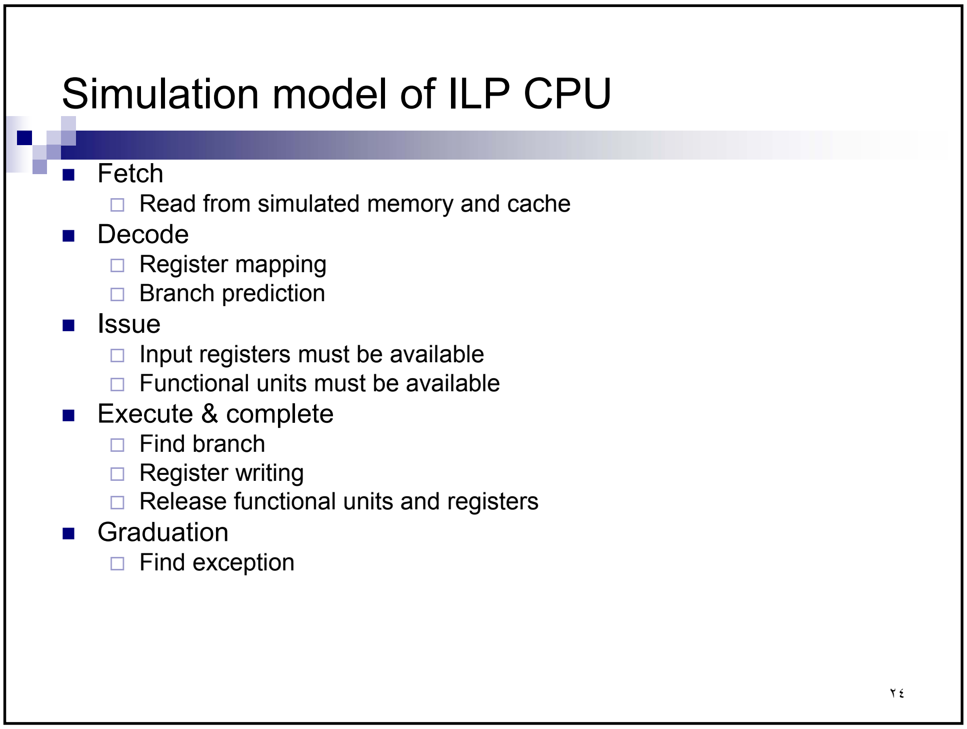
Pipeline

- Modern microprocessors add some additional capabilities to exploit ILP (Instruction level parallelism)
  - □ Compiler or CPU?
- Pipelining has long been recognized as way of accelerating the execution of computer instructions. Why?



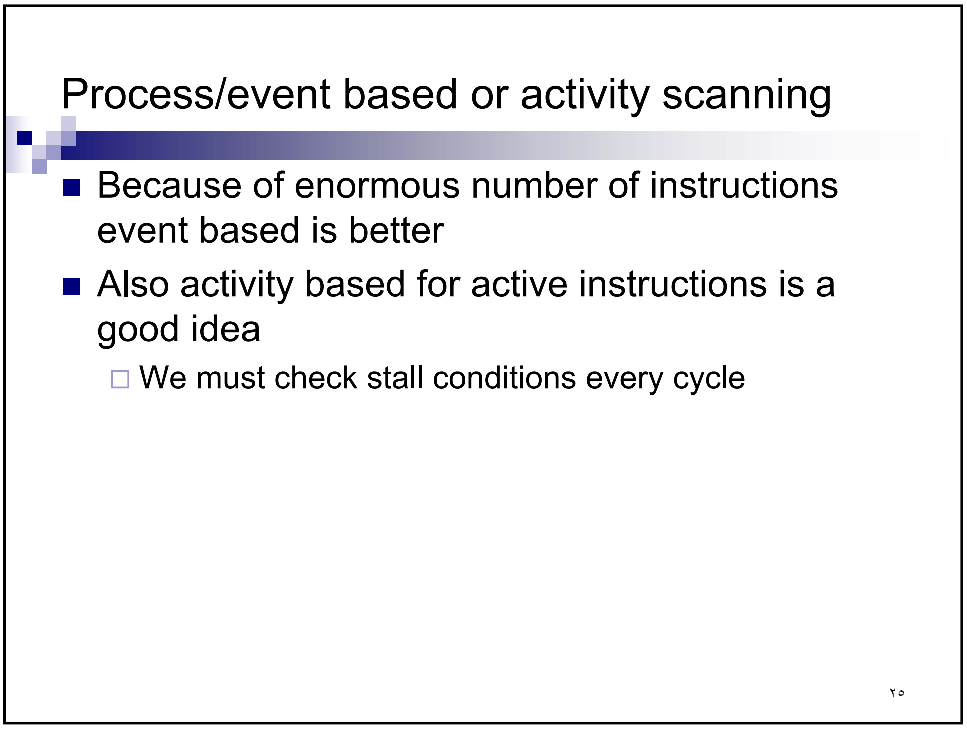
ILP CPU

- Pipeline stages
  - □ Instruction fetch
  - □ Instruction decode
  - □ Instruction issue (non order)
  - □ Instruction execute
  - □ Instruction complete
  - □ Instruction graduate
- Out of ordering
- We have different logical and physical registers
- Branch prediction



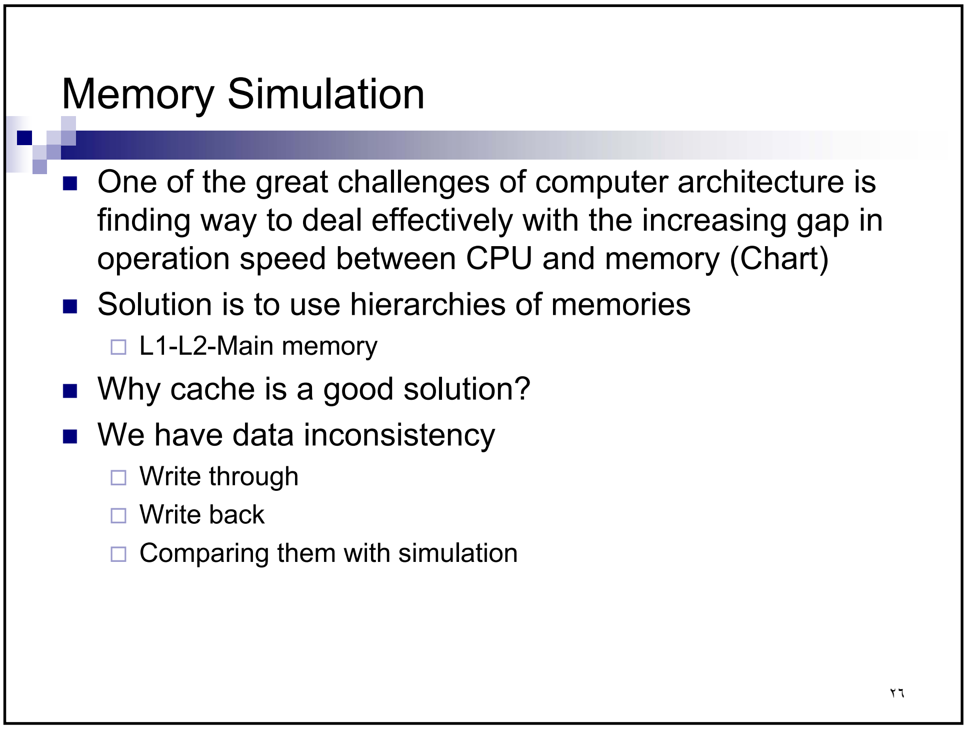
Simulation model of ILP CPU

- Fetch
  - □ Read from simulated memory and cache
- Decode
  - □ Register mapping
  - □ Branch prediction
- Issue
  - □ Input registers must be available
  - □ Functional units must be available
- Execute & complete
  - □ Find branch
  - □ Register writing
  - □ Release functional units and registers
- Graduation
  - □ Find exception



Process/event based or activity scanning

- Because of enormous number of instructions event based is better
- Also activity based for active instructions is a good idea
  - □ We must check stall conditions every cycle

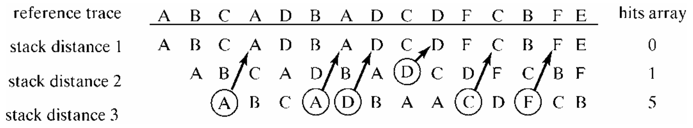


Memory Simulation

- One of the great challenges of computer architecture is finding way to deal effectively with the increasing gap in operation speed between CPU and memory (Chart)
- Solution is to use hierarchies of memories
  - □ L1-L2-Main memory
- Why cache is a good solution?
- We have data inconsistency
  - □ Write through
  - □ Write back
  - □ Comparing them with simulation

Memory Simulation

- Increasing hit ratio
- Replacement Policy (LRU)
- Set associative
  - □ Full associative



reference trace	A	B	C	A	D	B	A	D	C	D	F	C	B	F	E	hits array
stack distance 1	A	B	C	A	D	B	A	D	C	D	F	C	B	F	E	0
stack distance 2		A	B	C	A	D	B	A	D	C	D	F	C	B	F	1
stack distance 3			A	B	C	A	D	B	A	A	C	D	F	C	B	5