

## Topic 5: Performance

Reference : Very Background reading...  
Computer Architecture Design and Performance,  
Barry Wilkinson (Prentice Hall).

C.Sc. 110: Computer Systems

## Processor Developments

- Focus on processor oriented developments:
  - pipeline processors.
  - RISC and CISC.
  - multiprocessor computers.
- There have been corresponding advances in software and I/O devices which we shall ignore.

C.Sc. 110: Computer Systems

## Performance

- ◆ Many factors affect the overall performance of a computer:
  - the processor clock speed.
  - the capacity and speed of the memory.
  - the number of bits in each stored word.
  - the features provided by the instruction set.
- Performance is typically measured in either MIPS or MFLOPS but these are not very useful.

C.Sc. 110: Computer Systems

## The Role of The Clock

- The clock period is particularly important in determining the performance of a computer.
- Clock period = time allocated for each basic internal operation to the processor.
- E.g. clock frequency of 1000 MHz (1 GHz) = period of 1 ns. **If** one instruction per cycle then speed = 1000 MIPS (1 GIP).
- Speeding up the clock speeds up the processor.
- But, practical factors limit the speed at which processors can execute.

C.Sc. 110: Computer Systems

## The Role of The Clock

- The clock period is particularly important in determining the performance of a computer.
- Clock period = time allocated for each basic internal operation to the processor.
- E.g. clock frequency of 1000 MHz (1 GHz) = period of 1 ns. **If** one instruction per cycle then speed = 1000 MIPS (1 GIP).
- Speeding up the clock speeds up the processor.
- But, practical factors limit the speed at which processors can execute.

$$\text{time/program} = \left[ \frac{\text{instructions/program} \times \text{cycles/instruction}}{\text{time/cycle}} \right]$$

C.Sc. 110: Computer Systems

## Pipelined Systems

- Pipelining is a method which can be used to increase the speed of operation of the processor.
- The basic task is divided into a number of subtasks to be performed in sequence.
- Each subtask can be performed by a separate unit.

C.Sc. 110: Computer Systems

### Pipelined Systems ... contd.

IN

Stage 1 → Stage 2 → Stage 3

OUT

- Analogous to 'conveyor-belts' or 'Assembly Lines' in a factory.

C.Sc. 110: Computer Systems

### Pipelined Systems ... contd.

IN

Stage 1 → Stage 2 → Stage 3

OUT

- Analogous to 'conveyor-belts' or 'Assembly Lines' in a factory.

C.Sc. 110: Computer Systems

### Pipelined Systems ... contd.

IN

Stage 1 → Stage 2 → Stage 3

OUT

- Analogous to 'conveyor-belts' or 'Assembly Lines' in a factory.

C.Sc. 110: Computer Systems

### Pipelined Systems ... contd.

IN

Stage 1 → Stage 2 → Stage 3

OUT

- Analogous to 'conveyor-belts' or 'Assembly Lines' in a factory.

C.Sc. 110: Computer Systems

### Pipelined Systems ... contd.

IN

Stage 1 → Stage 2 → Stage 3

OUT

- Analogous to 'conveyor-belts' or 'Assembly Lines' in a factory.

C.Sc. 110: Computer Systems

### Pipelined Systems ... contd.

IN

Stage 1 → Stage 2 → Stage 3

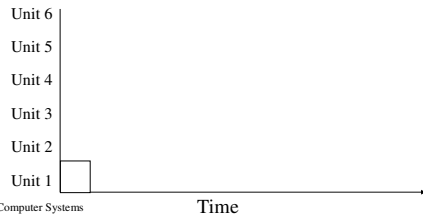
OUT

- Analogous to 'conveyor-belts' or 'Assembly Lines' in a factory.

C.Sc. 110: Computer Systems

## Performance of Pipelined Systems

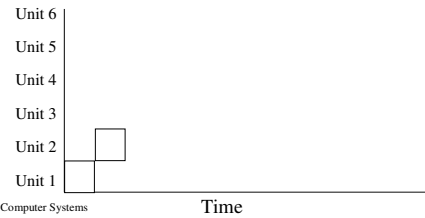
- A pipelined system with multiple units completes a single task no faster than a single unit.



C.Sc. 110: Computer Systems

## Performance of Pipelined Systems

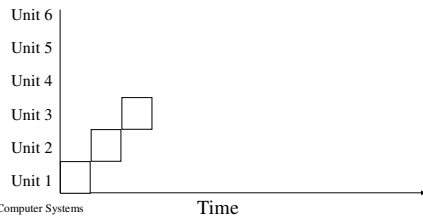
- A pipelined system with multiple units completes a single task no faster than a single unit.



C.Sc. 110: Computer Systems

## Performance of Pipelined Systems

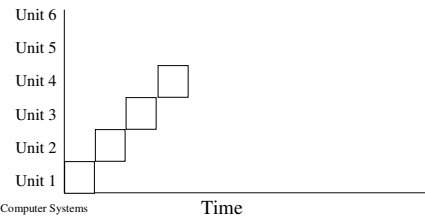
- A pipelined system with multiple units completes a single task no faster than a single unit.



C.Sc. 110: Computer Systems

## Performance of Pipelined Systems

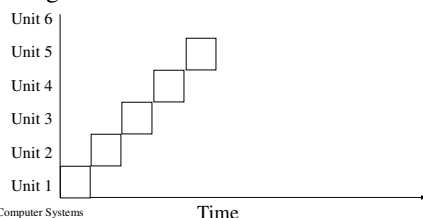
- A pipelined system with multiple units completes a single task no faster than a single unit.



C.Sc. 110: Computer Systems

## Performance of Pipelined Systems

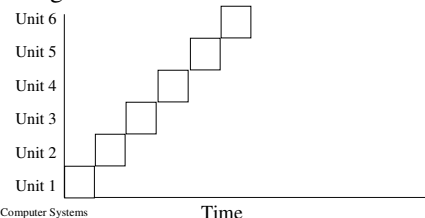
- A pipelined system with multiple units completes a single task no faster than a single unit.



C.Sc. 110: Computer Systems

## Performance of Pipelined Systems

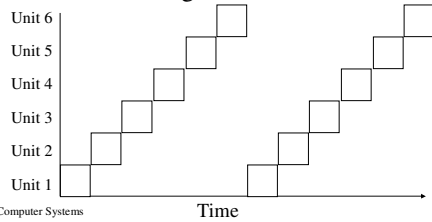
- A pipelined system with multiple units completes a single task no faster than a single unit.



C.Sc. 110: Computer Systems

### Performance of Pipelined Systems ... contd.

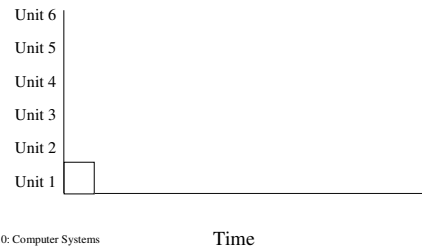
- If we wait for each task to complete then a pipelined system still completes tasks no faster than a single unit.



C.Sc. 110: Computer Systems

### Performance of Pipelined Systems ... contd.

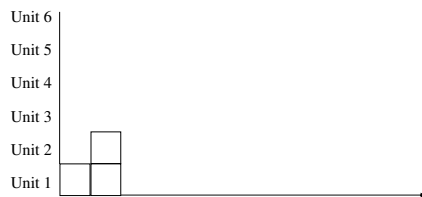
- But ...



C.Sc. 110: Computer Systems

### Performance of Pipelined Systems ... contd.

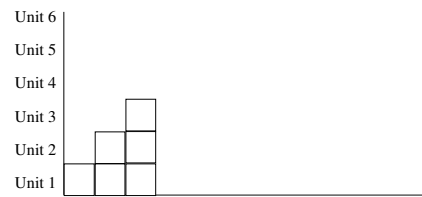
- But ...



C.Sc. 110: Computer Systems

### Performance of Pipelined Systems ... contd.

- But ...



C.Sc. 110: Computer Systems

### Performance of Pipelined Systems ... contd.

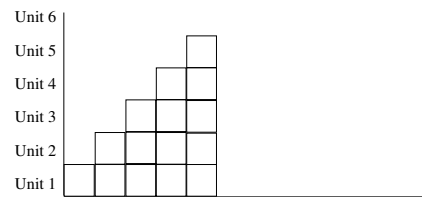
- But ...



C.Sc. 110: Computer Systems

### Performance of Pipelined Systems ... contd.

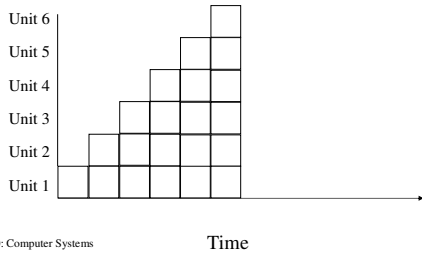
- But ...



C.Sc. 110: Computer Systems

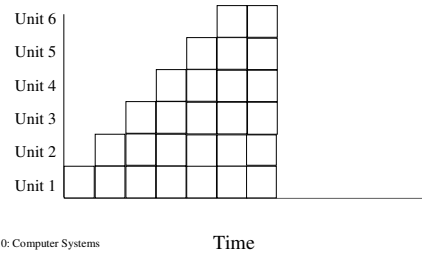
### Performance of Pipelined Systems ... contd.

- But ...



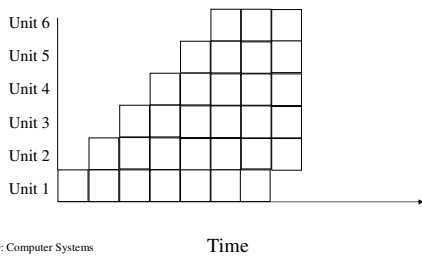
### Performance of Pipelined Systems ... contd.

- But ...



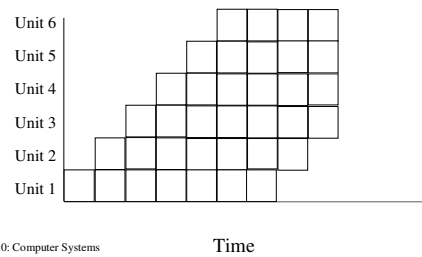
### Performance of Pipelined Systems ... contd.

- But ...



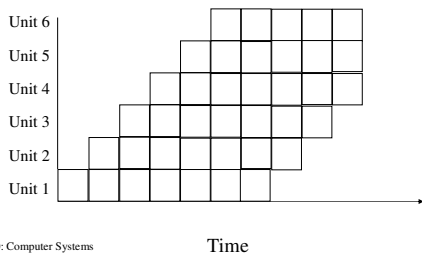
### Performance of Pipelined Systems ... contd.

- But ...



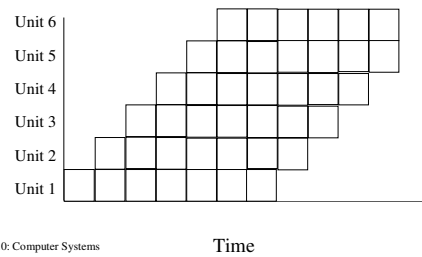
### Performance of Pipelined Systems ... contd.

- But ...



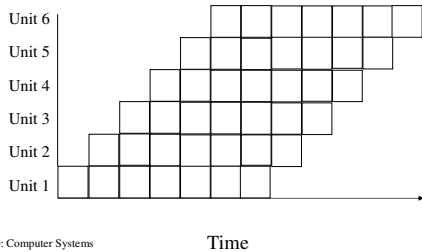
### Performance of Pipelined Systems ... contd.

- But ...



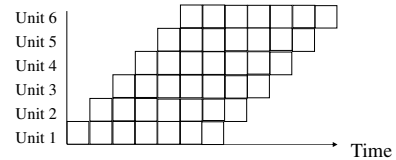
## Performance of Pipelined Systems ... contd.

- But ...



## Performance of Pipelined Systems ... contd.

- But ... if we keep the line busy we can complete seven tasks in the time it took us to do 2 without a pipeline.



## Fetch-Execute Cycle

```

procedure interpreter is
begin
  while (true) do
    fetch next machine language from address in B
    add 1 to contents of B
    decode the instruction
    execute the instruction
  end while
end
    
```

C.Sc. 110: Computer Systems

## Two, Three and Five Stage Pipelines

- Fetch Stage and Decode/Execute Stage.



- Fetch Stage, Decode Stage and Execute Stage.



- Fetch Stage, Decode Stage, Fetch Operands Stage, Execute Stage and Store Stage.
- C.Sc. 110: Computer Systems

## Problems With Pipelines

- Pipelines assume:-
    - instructions are executed in sequence.
    - no interaction between tasks.
  - Problems caused by:-
    - branch instructions.
    - data dependencies.
    - conflicts for hardware resources.
- C.Sc. 110: Computer Systems

## Branch Instructions

- Typically 10-20% of instructions in a program are branch instructions.
  - Conditional jumps are much worse than unconditional jumps as it takes longer to work out the correct next instruction.
- C.Sc. 110: Computer Systems

## Branch Instructions ... contd.

- Simplest approach is to flush the pipeline as soon as the branch instruction is detected - the longer the pipeline the more costly this is (i.e. greater sep-up time).
- Other possibilities include:
  - Buffers to fetch both possible instructions.
  - Prediction logic to fetch the most likely next instruction.
  - Delayed branch instructions.

C.Sc. 110: Computer Systems

## Data Dependencies

- Consider:

```
ADD #100      ; add 100 to the accumulator
ShiftL       ; shift the accumulator left
Store 200    ; store accumulator in addr. 200
```
- And use of a 5 stage pipeline, namely: Fetch Stage, Decode Stage, Fetch Operands Stage, Execute Stage and Store Stage.
- There are dependencies between the instructions.
  - e.g. ADD#100 in execute stage while ShiftL in Fetch Operands stage
- Dependencies need to be detected

C.Sc. 110: Computer Systems

## Pipeline Summary

- Substantial speed-ups are possible using pipelines.
- Almost all modern processors use pipelines.
- The gains are heavily dependent on the mix of instructions - this is one of the reasons why a straightforward measure of MIPS isn't very useful.

C.Sc. 110: Computer Systems

## Processor Design: RISC and CISC

- The design of a computer's instruction set can have a significant impact on its performance.
- Key question is how sophisticated to make the instruction set.
- Traditional approach is to provide complex instruction sets:
  - replace sequences of primitive operations.
  - many addressing modes.
  - support for procedure calls and parameter passing.

C.Sc. 110: Computer Systems

## Complex instruction Set Computers (CISCs)

- Early example was the VAX-11/780 with 303 instructions and 16 addressing modes
- Argument for complex instructions is that they can be implemented in hardware and are therefore faster than a corresponding sequence of primitive instructions.

C.Sc. 110: Computer Systems

## Complex instruction Set Computers (CISCs)

- Problem: adding complexity slows down all instructions.
- Complex instructions are not often used: e.g. in the VAX 20% of the instructions require 60% of the microcode but are used 0.2% of the time.
- 303 instructions requires 9 bits to specify

C.Sc. 110: Computer Systems

## The Reduced Instruction Set Computer (RISC) Philosophy

- Transfer Complexity
- Have very simple instructions and few addressing modes.
- Optimise these instructions - possibly implementing them directly in hardware.
- Less frequent operations are provided only if their inclusion doesn't affect performance.

C.Sc. 110: Computer Systems

## RISC Characteristics

- RISC machines typically have < 128 instructions.
- Often only support 4 addressing modes.
  - Many registers...
- Instructions execute in one cycle and microcode is avoided where possible.

C.Sc. 110: Computer Systems

## Compilers and RISC and CISC

- RISC machines require programs to be translated into a very basic set of instructions:
  - large machine code programs.
  - complex compilers.
- CISC machines provide more support for the compiler:
  - smaller machine code programs.
  - but ... difficult for compilers to make use of all the features.

C.Sc. 110: Computer Systems

## Multiprocessor Systems

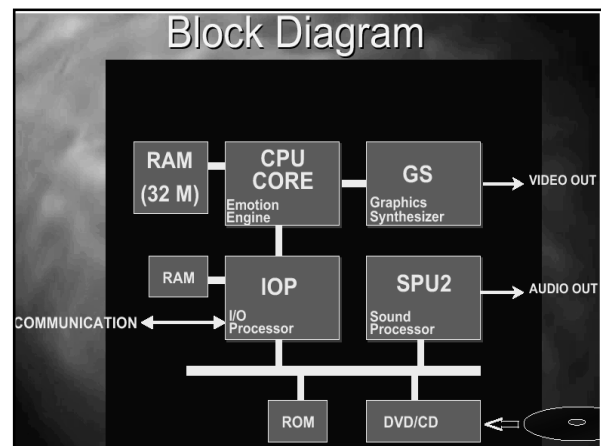
- Another approach to improving performance is to introduce more processors.
- Each processor works on some part of the overall task in parallel.
- Problems:
  - Dividing task into parallel units with fewest possible dependencies.
  - Minimising the communication overheads between the processors.
  - Marginal utility

C.Sc. 110: Computer Systems

## Case Study – Sony's PS2

- Targets performance...
  - Clock speed 'only' 300 MHz
  - Employs 6 stage pipeline
  - 128-bit core
  - Many RISC attributes
    - (at least one) instructions completed per cycle
- Effectively multiprocessor system
  - Sound processor, Graphics Synthesiser, I/O processor
  - Two vector units (co-processors)
    - Approx 1 GIP
    - 6.2 GFLOPS

C.Sc. 110: Computer Systems



## Summary

---

- Focused on performance.
- Problems with measuring performance.
- Improvements in processor performance:
  - pipeline processors.
  - RISC and CISC machines.
- Multiprocessor systems.