

## OPro Sem Preparation

Unit-4 Topics

- Instruction Execution
- Building Data path
- Designing a control unit
- Hardwired control
- Microprogrammed control
- Pipelining
- Data Hazard
- Control Hazard

Note: To those who read from my notes, my handwriting can be messy, so if you don't understand anything, please, don't ask!

### Pipelining

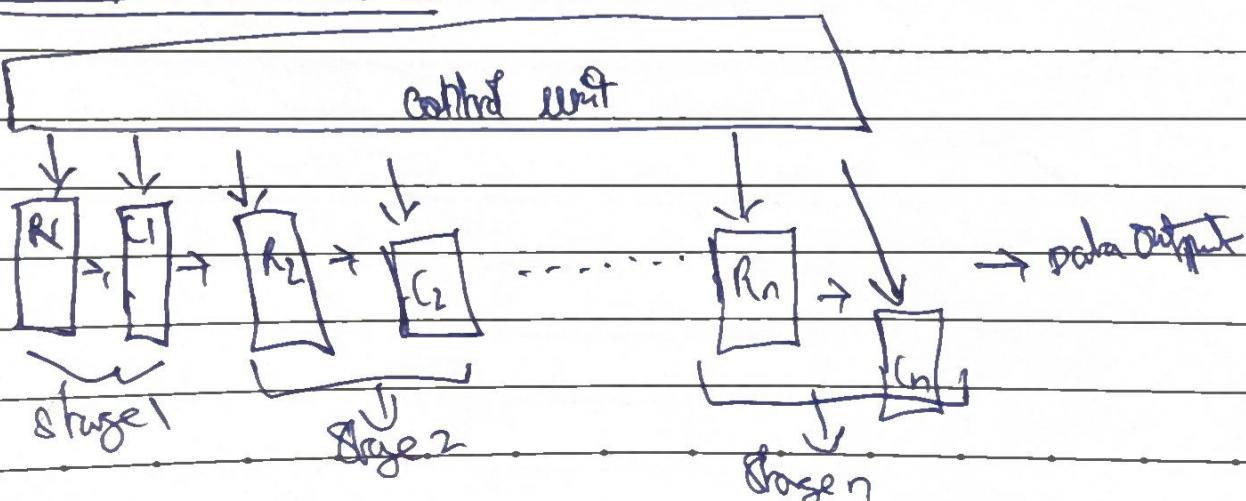
Pipelining is the process of storing and prioritizing computer instructions that the processor executes.

→ Stages: fetch instruction, decode instruction, calculate operands, fetch operands, execute instructions, write operands

Time between instructions (pipelined) =  $\frac{\text{Time between instruction (or pipeline)}}{\text{Number of pipeline stages}}$

5 stages: fetching, decoding, executing, accessing, writing  
 +  
 Instruction → Instruction → Operation → Operands → result into register

### Pipelined Datapath and Control



- \* A Pipeline processor consists of a sequence of  $n$  data processing circuits called elements, stages or segments.
- \* These stages collectively perform a single operation on a stream of data operands passing through them.
- \* Each stage consists of 2 major blocks
  - Multicard Input register
  - Data path circuit
- \* If time required to perform single sub operation in pipeline is  $T$  sec then from  $M$  stage pipeline the time required to complete a single operation is  $MT$  sec. This is called delay or latency of the pipeline.
- \* The max no of operations completed per second can be given as  $\frac{1}{T}$ . This is called throughput of the pipeline.

### Advantages:

- \* The cycle time of the processor is reduced.
- \* It increases the throughput of the system.
- \* It makes the system reliable.

### Disadvantages:

- \* The design of pipelined processor is complex and costly to manufacture.
- \* The instruction latency is more.

Ex : Laundry :

i) Roommate

ii) Dirty clothes  $\rightarrow$  washing machine  $\rightarrow$  dryer  $\rightarrow$  fold  $\rightarrow$  keep away

iii) Dirty clothes  $\rightarrow$  washing machine  $\rightarrow$  dryer  $\rightarrow$  fold  $\rightarrow$  keep away

iv) Report

Time between instructions :  $\frac{\text{Time between instruction (exp. period)}}{\text{No. of pipe stages}}$

### Pipeline performance:

The cycle time can be determined as

$$t = \max [t_i] + d$$

$$= t_m + d \quad i=1 \dots k$$

$t_i \Rightarrow$  time delay of the circuitry in  $i^{\text{th}}$  stages of pipeline

$t_m \Rightarrow$  Max stages delay

$k \Rightarrow$  number of stages in pipeline

$d \Rightarrow$  time delay of latch

### Define datapath ?

datapath is an unit used to operate an or hold data within a processor.

### Hazard in pipeline

Any reason that causes the pipelining to stall is called Hazard.

## Structural Hazard:

The Hazard that exists because of conflict due to insufficient resources when even with all possible combination, it may not be possible to overlap the operation is called structural hazard.

## Advantages of pipelining:

- Instruction throughput increases
- Increase in number of pipeline stages

## Various phases in executing an instruction:

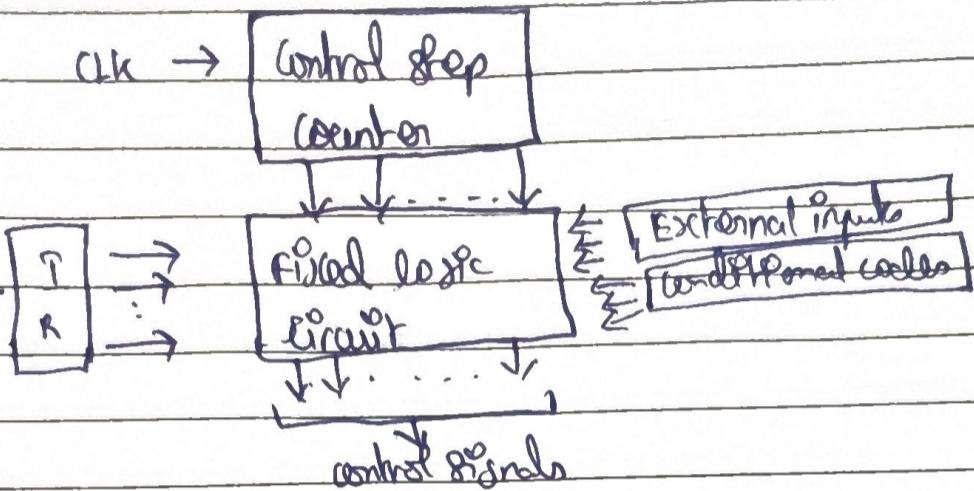
- i) fetching an instruction from memory
- ii) Decoding the fetched instructions
- iii) Execute the operation or calculation of an address
- iv) Access an operand in local memory
- v) Write the result into register

## Process to handle interrupt request

- ① The process that runs when an interrupt is generated is an interrupt handler.
- ② The CPU saves the state of ongoing process and shifts its attention to the interrupt generated by giving access to the interrupt handler. This entire process is called interrupt handling.

## Hardwired Control

Diagram:



- \* In the hardwired control, the control unit uses fixed logic circuits to interpret instructions and generate control signals from them.
- \* The fixed logic circuit uses contents of the control step counter, contents of instruction register, contents of the conditional code flag and the external input signals such as MFC and interrupt requests to generate control signals.
- \* Furthermore, the hardware control unit is relatively inflexible because it is difficult to change the design, if one wishes to correct design error or modify the instruction set.

### Advantages:

- More fast
- Greater chip area efficiency

### Disadvantages:

- More the control signals required by the CPU, more complex will be design of control unit.
- Modifications in control signals are very difficult.

## Microprogrammed control

- \* Microprogramming is a method of control unit design in which the control signals selection and sequencing information is stored in a ROM or RAM called a control memory CM.
- \* A sequence of one or more operations designed to control specific operation, such as addition, multiplication is called a microprogram.
- \* The microprograms for all instructions are stored in a memory.
- \* The address where these microprograms instructions are stored in CM is generated by Microprogram sequencer / microprogram controller.

### Horizontal MPs:

The control signals are represented in the decoded binary format that is 1 bit /cs. More than 1 control signals can be enabled at a time.

### Vertical MPs:

The control signals are represented in encoded binary format for n control signals log<sub>2(n)</sub> bits are required.

### Advantages

- ⇒ Cheap and less error prone to implement.
- ⇒ The design process is orderly and systematic.

## Roadblocks

- ⇒ flexibility is achieved at some extra hardware cost
- ⇒ slower than hardwired control unit

## Hazards

Present the next instructions in the instructing stream for executing during its designated clock cycle

### i) Structural Hazard :

- \* when too many complex instruction is given that in terms need to change the design of the control which is complex and hardware hard it leads to this kind of errors
- \* can be prevented by using proper instruction cycle

### ii) Data Hazard :

when either the source or destination operand of an instruction is available at the time of expected in the pipeline, and as a result pipeline is stalled, we say such situations are called data hazard

## Handling Hazards

### Data Hazard handling :

- i) Forwarding
- ii) Cache Renaming
- iii) Stall Instruction

### i) Forwarding

It adds special circuitry to pipelining. This method works because it takes less time for the required values to travel through a wire than it does for a pipeline register to compute its result.

### ii) Code Reordering

We need a special type of software to reorder code and call this type of software a hardware dependent compiler.

### iii) Stall insertion

It inserts one or more stall into the pipeline, which delays the execution of the current instruction until the required operand is written to register file. But this method decreased pipeline efficiency and throughput.

## Control Hazard

The Control Hazard occurs whenever the pipeline make incorrect branch prediction decisions resulting in instruction entering the pipeline that must be discarded.

A Control Hazard is often referred to as a branch hazard handling:

- i) Static Branch prediction
- ii) Branch delay slots
- iii) Dynamic Branch prediction
- iv) Loop unrolling
- v) Branch Prediction Buffer

i) Static Branch prediction: The hardware makes static guess about whether the branch will be taken or not. The simplest static branch prediction strategy is always guess guess the branch will not be taken.

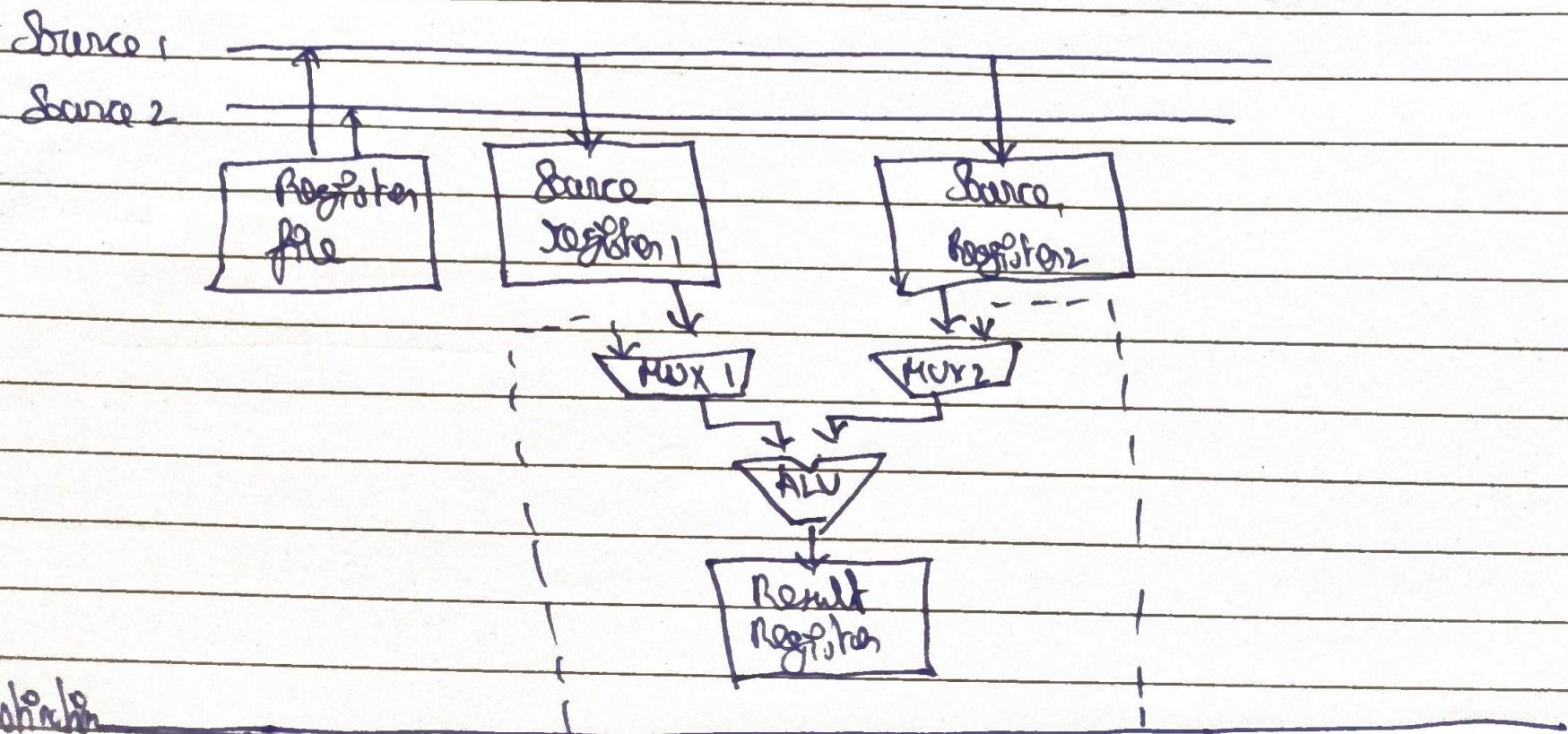
ii) Branch delay slots: Instructions that immediately follow a branch are executed in the pipeline stages behind the branch regardless of whether the branch conditions is satisfied or not.

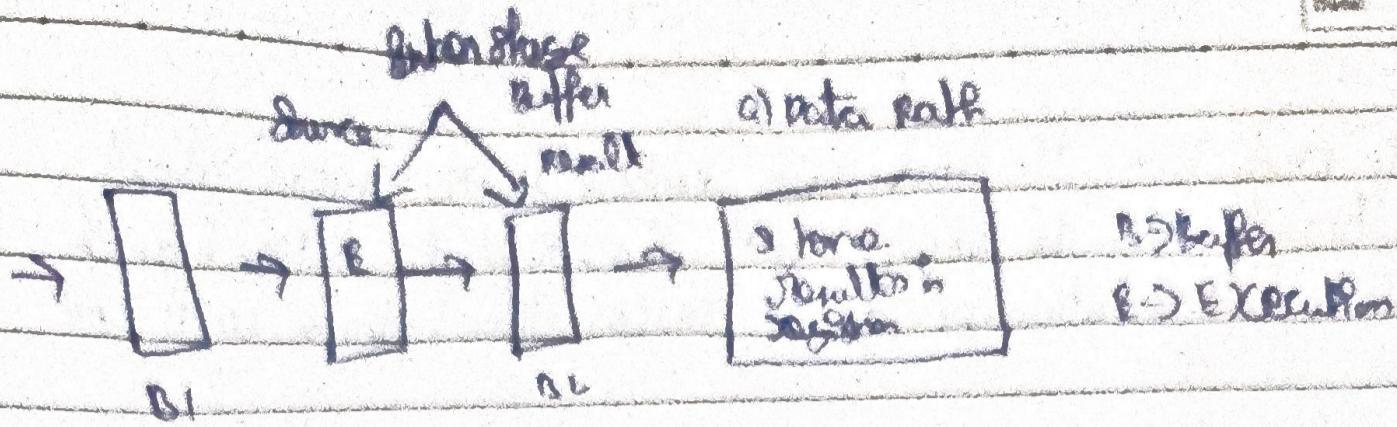
iii) Dynamic Branch prediction: It uses run-time info to make predictions past behaviour of the branch is used to predict future behaviour

iv) Loop unrolling: It involves replicating the body of the loop multiple times to decrease the overhead of loop control execution

v) Branch Prediction Buffer: It's a small memory indexed by the lower bits of the instruction address that holds a bit that says whether the branch was recently taken or not.

### Data Hazard operand forwarding





## Instruction Execution

The CPU executes instruction through a continuous cycle with these main phases

- 1) Fetch → gets instruction and input by using PC
- 2) Decode → decodes instructions in DB
- 3) Executes → ALU & used
- 4) Store / write back phase

Key Components:

Program Counter PC

Instruction Register

Memory Address Register

Memory Data Register

Control Unit

ALU → Arithmetic Logic Units

Building a data path:

1) Registers:

2) ALU

3) MUX

4) Buses

5) Memory Interface

6) Adders

## Dark path architecture

