Computer Organization and Architecture: A Pedagogical Aspect

Module

Control Unit

Units in the Module

- **Instruction Cycle and Micro-operations**
- **Control Signals and Timing sequence**
- **Control Signals for Complete Instruction execution**
- **Handling Different Addressing Modes**
- **Handling Control Transfer Instructions**
- **Design of Hard-wired Controlled Control Unit**
- **Different Internal CPU bus Organization**
- **Micro-instruction and Micro-program**
- **Organization of Micro-programmed Controlled Control Unit**

Module Summary

- There are several components inside a CPU, namely, ALU, control unit, general purpose registers, special purpose registers like instruction register etc.

- There are several ways to place these components and interconnect them called processor organization.

- Single bus, two buses and three buses are some of the widely accepted processor organizations.

Module Summary

- The control unit is responsible for generating signals for data flow control within the components of the CPU, memory and I/O devices.
- The functionalities of the different Arithmetic and Logic units are determined by the control signals.
- The control unit takes input from the flags (representing the result of the previous instruction(s)), instruction resister (Opcode), control signals from the control bus and internal signals.
- The output of the control unit comprises signals that determine which functionality (out of all possible ones) is to be performed by a sub-module and what would be the data flow

Module Summary

- Each instruction in control unit executes in terms of a sequence of micro-instructions.

- The CPU is responsible for fetching, decoding and executing the instructions. The control unit in the CPU is responsible for decoding the instructions and generating the control signals corresponding to the sequence of micro-instructions involved in the instruction.

- There are two approaches to implement a control **unit—Hardwired** and **Micro-programmed**.

Module Summary

- The Hardwired control unit is a sequential state machine to generate the specific fixed sequences of control signals. In this case the state machine is implemented as a fixed sequential circuit; hence the name (microinstructions are) *hard-wired* into the computer.

- In micro-programmed control unit, the logic of the control unit is specified by a micro-program.

- A micro-program consists of a sequence of instructions in a microprogramming language. These are instructions that specify micro-operations.

- The concept of micro-program is similar to computer program. The sequence of instruction fetch is controlled by counter called micro-program counter (MPC). Micro-instructions in micro-program are stored in micro-program control memory and the execution is controlled by MPC.

Module objectives

- **Comprehension: Describe:--**Describe about the control steps and control signals needed to execute an instruction.

- **Synthesis: Design**:--Design issues of the control steps of the basic instructions (like memory read, memory write, data transfer, arithmetic and logic etc.) for execution with reference to a given organization

- **Synthesis :Design:** --Design of instructions for control transfer operations like conditional branch, function CALL/RETURN, etc.

- **Application: Demonstrate:--**Demonstrate the implementation of a hardwired-controlled control unit.

- **Comprehension: Describ**e :--Describe about the concept of micro-instruction, micro-program and sequencing issues of micro-instructions

- **Synthesis: Design**:--Design issues for implementation of Microprogram-controlled control unit

Module Learning Strategy

This module deals with the need and design issues of Control Unit of the processor.

- Unit-I explains the Instruction Cycle and Micro-Operation of an instruction and Unit-II explains the control signals needed for a micro-operation and timing sequence.
- Unit-III describes the control signals and timing sequence required for the execution of a complete instruction.
- Unit-IV and Unit-V explain the handling of different addressing modes and control transfer instruction.
- Unit-VI describes the design issues of Hardwired Controlled Control Unit.
- Unit-VII gives brief idea about different internal bus organization of the processor.
- Unit-VIII and Unit-IX deals with micor-program and Micro-Programmed Controlled Control Unit design.

Module Learning Strategy

- UNIT-I

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 15 (15.1).

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.1)

- UNIT-II

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 15 (15.2).

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.1)

- UNIT-III

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 15 (15.2).

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.2)

- UNIT-IV

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 15 (15.2).

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.2)

Module Learning Strategy

- UNIT-V

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 15 (15.2).
Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.2)

- UNIT-VI

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 15 (15.3).
Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.4)

- UNIT-VII

Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 15 (15.3).
Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.3)

Module Learning Strategy

- UNIT-VIII
- Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 16 (16.1 and 16.2).
  Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.5)

- UNIT-IX
- Willium Stallings, Computer Organization and Architecture - Designing for Performance, 8th Eds., Pearson. Chapter 16 (16.3).
  Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, 5th Eds, Mc-Graw Hill, Chapter 7 (7.5)

- Learner can also look for the course on Computer Organization and Architecture in the NPTEL course repository which is available in the following link
  http://nptel.ac.in/courses/106103068/

Please go through the module CPU Design.

Module
Control Unit

Unit-1
Instruction Cycle and Micro-operations

Units in the Module

- **Instruction Cycle and Micro-operations**
- **Control Signals and Timing sequence**
- **Control Signals for Complete Instruction execution**
- **Handling Different Addressing Modes**
- **Handling Control Transfer Instructions**
- **Design of Hard-wired Controlled Control Unit**
- **Different Internal CPU bus Organization**
- **Micro-instruction and Micro-program**
- **Organization of Micro-programmed Controlled Control Unit**

Unit Summary

- Machine instructions are generally complex and require multiple clock cycles to complete. So, machine instructions are also termed as macro-instructions in this context. Each machine instruction is implemented in terms of micro-operations i.e., set of actions (data flows and controls) that can be completed in a single clock cycle.

- The operations involved in the four cycles can be carried out by performing one or more of the following micro-operations in some pre-specified sequence:
  - Fetch the contents of a given memory location and load them into a CPU register.
  - Store a word of data from a CPU register into a given memory location.
  - Transfer a word of data from one CPU register to another or to the ALU.
  - Perform an arithmetic or logic operation, and store the result in a CPU register.

Unit Summary

- In the simplest controller design, all the micro-operations can be executed in sequence devoting one clock pulse to each operation.
- However, there may be many micro-operations involving movement of data into or out of registers and they do not interfere with one another.
- So devoting separate time units to them may lead to under utilization of resources.
- So, we can parallelize several micro-operations in a single clock thus saving the time and utilize resources optimally.
- This concept of parallelizing micro-operations in a single clock is called clock grouping. However, care must be taken that we do not parallelize operations that interfere with one another.

Unit Objectives

- **Comprehension: Discuss:--** Discuss the concept of Instruction cycle and micro-operations of an instruction.

- **Analysis: Specify:--** Specify the different phases involve in an instruction and the micro-operations needed to carry out those phases.

- **Synthesis: Design:--**Design the sequence of micro-operations to complete the execution of a given instruction.

What is a micro-operation?

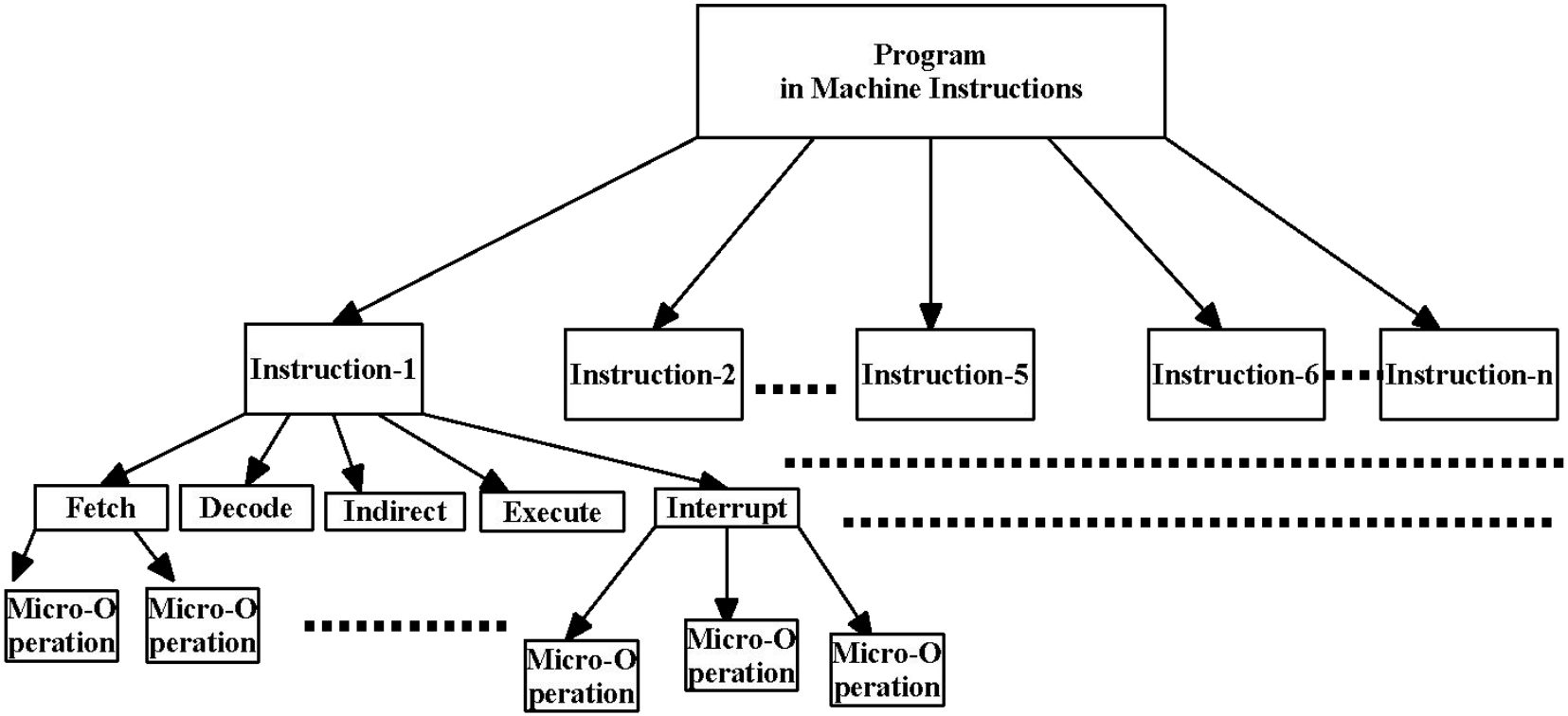
- Machine instructions are generally complex and require multiple clock cycles to complete. So, machine instructions are also termed as macro-instructions in this context.

- Each machine instruction is implemented in terms of micro-operations i.e., set of actions (data flows and controls) that can be completed in a single clock cycle. In other words, micro-operations are detailed low-level atomic instructions which can be executed in a single clock cycle and are generally used to implement complex machine instructions

- The micro-operations can be classified into the following classes:

  - Register transfer micro-operations
  - Arithmetic micro-operations
  - Logic micro-operations.
  - Shift micro operations

What is a micro-operation?



Micro- operations

The execution of a program consists of sequential execution of machine instructions. Execution of an instruction (called instruction cycle) includes the sub-cycles Fetch, Decode, Indirect, Execute, Interrupt etc.

Further, each sub cycle has a number of smaller or atomic steps called Micro-operations. Whenever, an instruction is executed it passes through the sub-cycles each of which involves the corresponding sequence of Micro-operations.

For example, all instructions involve the “Fetch” sub-cycle.

$$t_1 : PC \rightarrow MAR$$
$$t_2 : MEMORY \rightarrow MBR$$
$$t_3 : PC + 1 \rightarrow PC$$
$$t_4 : \ MBR \rightarrow IR$$

Clock cycle grouping

- Each macro instruction involves execution of a sequence of micro-operations.
- In the simplest controller design, all the micro-operations can be executed in sequence devoting one clock pulse to each operation.
- However, there may be many micro-operations involving movement of data into or out of registers and they do not interfere with one another.
- So devoting separate time units to them may lead to under utilization of resources. So, we can parallelize several micro-operations in a single clock thus saving the time and utilize resources optimally.

Clock cycle grouping

$$t_1 : PC \rightarrow MAR$$
$$t_2 : MEMORY \rightarrow MBR$$
$$t_3 : PC + 1 \rightarrow PC$$
$$t_4 : MBR \rightarrow IR$$

1st time step the address of instructions from PC is placed into the MAR.

In the second step the contents of the data bus is read into MBR.

These two micro-operations cannot be performed in a single step because address needs to be give in the address bus (via MAR) before the corresponding content can be read into the MBR (via the data bus). In other words, MAR<=PC must precede MBR<=(MEMORY).

In the 3rd step PC is incremented. This operation can be merged with the 2nd step because the content of PC is already saved in MAR in the 1st step.

However, we cannot merge the 4th operation into the 2nd step because reading contents of data bus into MBR and reading contents of MBR into IR must be done in different clock cycles.

Clock cycle grouping

The rules for clock cycle grouping are as follows:

- Proper sequence must be followed.
- Ideally there should not be any conflicts.
  - Must not read and write in the same register in the same clock cycle.
- Incrementing PC involves addition.
  - PC = PC+1 involve addition

Micro-operation during the fetch phase of instruction

$$PC \rightarrow MAR$$
$$MEMORY \rightarrow MBR$$
$$PC + 1 \rightarrow PC$$
$$MBR \rightarrow IR$$

- At the beginning of the fetch cycle, the address of the next instruction to be executed is in the Program Counter (PC).This address is copied to the MAR. This happens in one clock cycle. So the micro-operation involved is

  t1: MAR<-- (PC)

- After the completion of this operation, the address in the MAR is placed in the address bus of the memory and the control unit issues a READ command on the control bus. The memory location specified in the address bus is read and result appears on the data bus and is copied into the MBR. Also, the PC is incremented for the execution of next instruction.

- These two actions i.e., reading from the memory and incrementing the PC, do not interfere each other, hence these are performed simultaneously in a single clock cycle to save time. The micro-operations involved in time t_2 are

  t2: MBR<--Memory

  PC<--(PC) + 1

Micro-operation during the fetch phase of instruction

- After the completion of this operation, the contents of the MBR are moved to the instruction register (IR). This frees the MBR for use during an indirect cycle, where some operands are required to be fetched from the memory. The micro-operation involved here is given below, if there is no requirement of indirect fetch.

  t3:- IR<--(MBR)

- This micro-operation just involves transferring the instruction (OPcode and immediate data) from MBR register to IR.

- However, in case of direct (or non-immediate) cycle the micro-operations given below are required. In this case the address of the operand present in the IR is copied to MAR. In the next step, the operand is copied from the memory to the MBR which in the subsequent time step is copied to IR-address field.

  *t3: MAR <-- (IR(**address**))* - ***indirect address field***

  t4: MBR <-- (memory)

  *t5: IR(**address**) <-- (MBR(**address**))* - ***direct address field***

Micro-operation during the fetch phase of instruction

The advantages of the clock-cycle grouping in the fetch cycle are as flows:

- This grouping avoids conflicts between operations. Due to this clock-cycle grouping, there cannot be both read and write operation from the same register in one time unit, as they occur during different time units in this grouping. For example, the micro-operations (MBR<--Memory) and (IR<--MBR) cannot occur at the same time.

- This also maintains the proper sequencing of the instructions

- Saving of one time unit. Using clock grouping, in case of non-indirect and indirect fetch cycle the time cycles required are 3 and 5, respectively. If clock grouping is not used then we require 4 and 6 time cycles, respectively for the same situation.

Micro-operations in interrupt cycle and execute cycle

The micro-operations in the interrupt cycle are as follows:

t1: *MBR <-- (PC)*
t2: *MAR <--* ***Address to save the PC contents***
PC <-- ***address of start of interrupt service routine***
t3: *memory<-- (MBR)* ***- actual saving of the PC contents***

The micro-operations in the execute cycle for addition is as follows:

We assume that addition instruction is ADD R, X (add the contents of memory location X to Register R, place the result in R).

t1: *MAR <-- (IR(address(X)))*
t2: *MBR <-- (Memory)*
t3: *R <-- (R) + (MBR)*

Questions and Objectives

- Q1: What is a micro-operation? Explain the principle of program execution in terms of instruction cycle and micro-operation.
- Q2: What is the concept of clock cycle grouping? Mention the rule for clock cycle grouping and explain these with examples.
- Q3: Give the micro-operation involved during the fetch phase of an instruction. Show the symbolic representation of each micro-operation along with time step. Show the clock cycle grouping and its advantages.
- Q4: Give the micro-operations involved in interrupt cycle and execute cycle (some mathematical operation). Show the symbolic representation of each micro-operation along with time step.

- **Comprehension: Discuss:--** Discuss the concept of Instruction cycle and micro-operations of an instruction.
- **Analysis: Specify:--** Specify the different phases involve in an instruction and the micro-operations needed to carry out those phases.
- **Synthesis: Design:--**Design the sequence of micro-operations to complete the execution of a given instruction.