COMPUTER ARCHITECTURE

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

LECTURE 6

BY NDUMISO E. KHUMALO

CONTROL UNIT OPERATION AND MICROPRAGRAMMED CONTROL

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

Objective

- Micro operations
- Control of the Processor

Introduction

- we turn to a discussion of the control unit, which controls the operation of the processor.
- We need to:
  - Explain the concept of micro-operations and define the principal instruction cycle phases in terms of micro-operations.
  - Discuss how micro-operations are organized to control a processor.
  - Understand hardwired control unit organization

Micro operations

- We have seen that the operation of a computer, in executing a program, consists of a sequence of instruction cycles, with one machine instruction per cycle
  - We must remember that this sequence of instruction cycles is not necessarily the same as the written sequence of instructions that make up the program, because of the existence of branching instructions.
  - What we are referring to here is the execution time sequence of instructions.

Micro operations

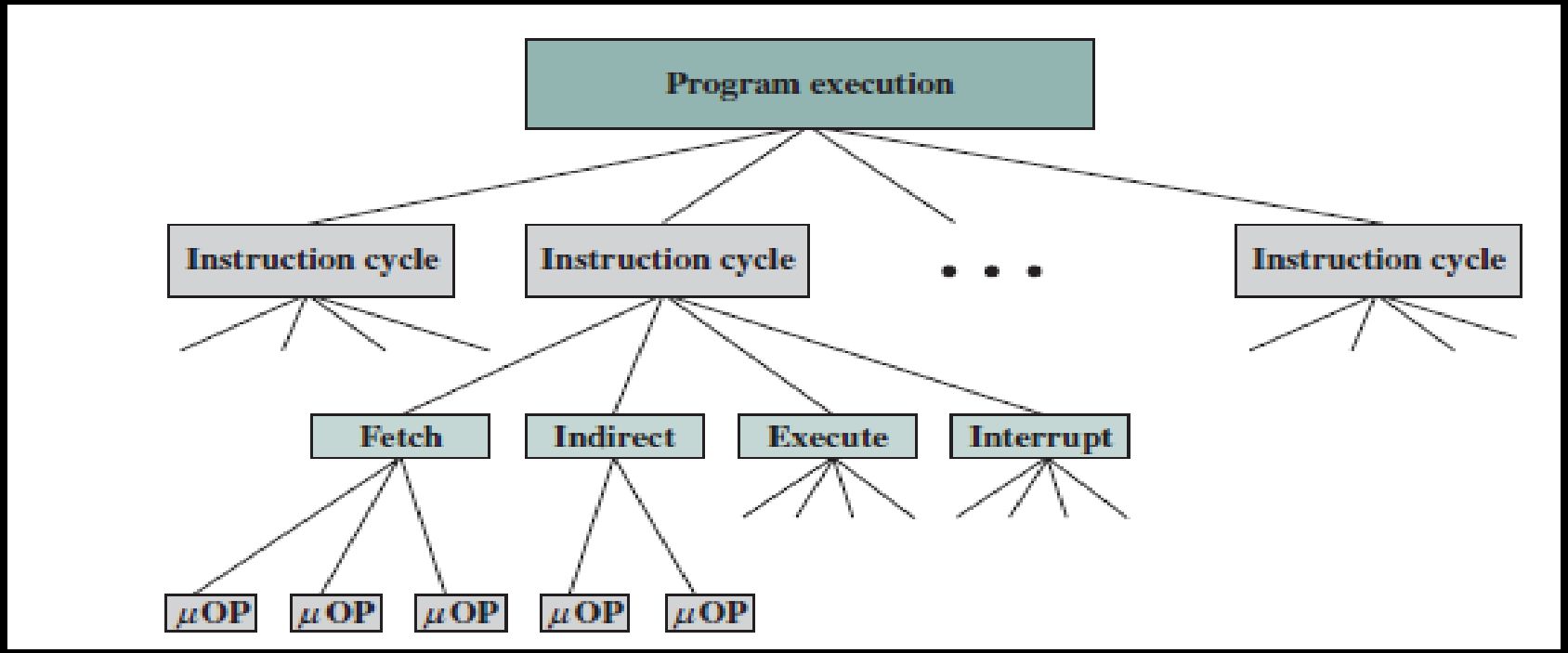
- We have further seen that each instruction cycle is made up of a number of smaller units.
  - One subdivision that we found convenient is fetch, indirect, execute, and interrupt, with only fetch and execute cycles always occurring.
- To design a control unit, however, we need to break down the description further.
  - each of the smaller cycles involves a series of steps called ***micro operations***, each of which involves the processor registers

Micro operations

- The prefix micro refers to the fact that each step is very simple and accomplishes very little
- the execution of a program consists of the sequential execution of instructions.
  - Each instruction is executed during an instruction cycle made up of shorter subcycles (e.g., fetch, indirect, execute, interrupt).
  - The execution of each subcycle involves one or more shorter operations, that is, micro- operations.

Micro operations

- Micro-operations are the functional, or atomic, operations of a processor



Micro operations

The Fetch Cycle

- occurs at the beginning of each instruction cycle and causes an instruction to be fetched from memory
- Four registers are involved:
  - 1. Memory address register (MAR): Is connected to the address lines of the system bus. It specifies the address in memory for a read or write operation.

Micro operations

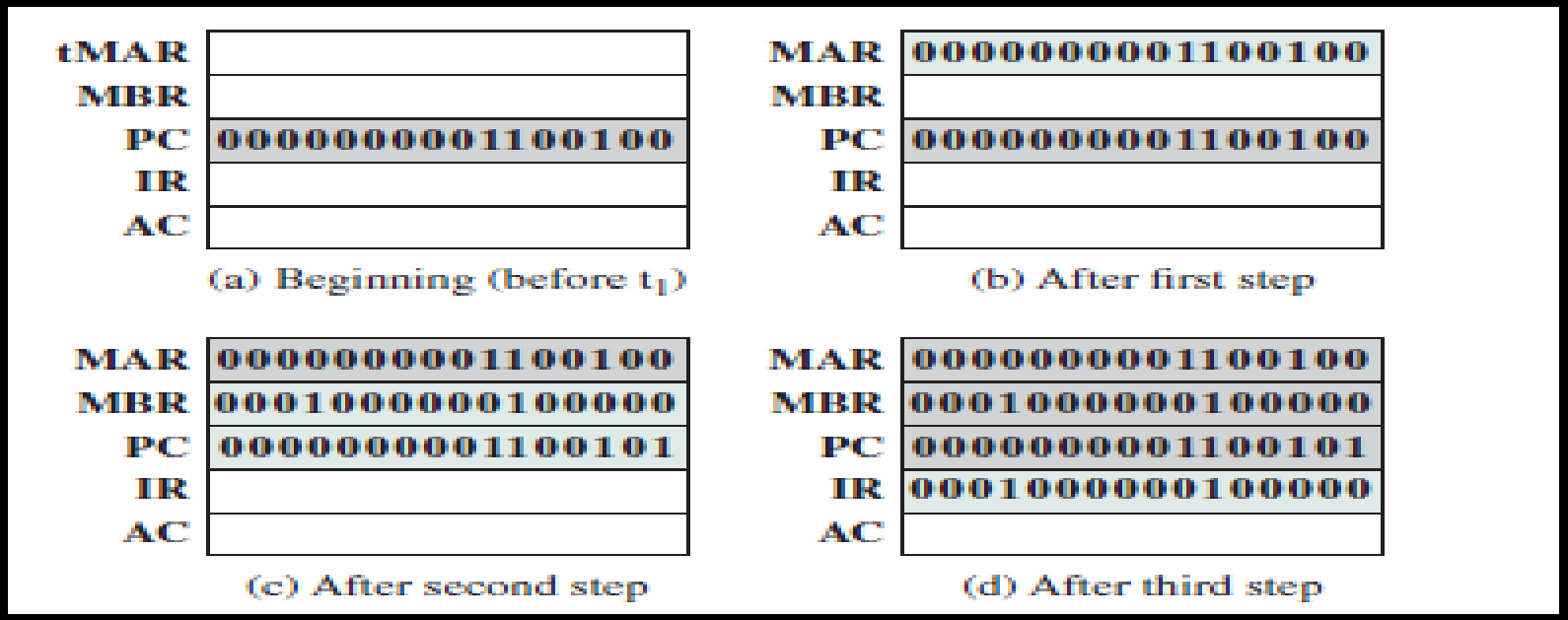
The Fetch Cycle

- Four registers are involved:
  - 2. Memory buffer register (MBR): Is connected to the data lines of the system bus. It contains the value to be stored in memory or the last value read from memory.
  - 3. Program counter (PC): Holds the address of the next instruction to be fetched.
  - 4. Instruction register (IR): Holds the last instruction fetched

Micro operations

The Fetch Cycle

- the fetch cycle from the point of view of its effect on the processor registers is as follows:



Micro operations: The fetch cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

The Fetch Cycle

- the fetch cycle from the point of view of its effect on the processor registers is as follows:
  - At the beginning of the fetch cycle, the address of the next instruction to be executed is in the program counter (PC); in this case, the address is 1100100.
  - The first step is to move that address to the memory address register (MAR) because this is the only register connected to the address lines of the system bus.

Micro operations: The fetch cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

– The second step is to bring in the instruction.
  - The desired address (in the MAR) is placed on the address bus, the control unit issues a READ command on the control bus, and the result appears on the data bus & is copied into the memory buffer register (MBR).
  - We also need to increment the PC by the instruction length to get ready for the next instruction
  - Because these two actions (read word from memory, increment PC) do not interfere with each other, we can do them simultaneously to save time

Micro operations: The fetch cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The third step is to move the contents of the MBR to the instruction register (IR).
  - This frees up the MBR for use during a possible indirect cycle.

- Thus, the simple fetch cycle actually consists of three steps and four micro- operations.
  - Each micro-operation involves the movement of data into or out of a register.

Micro operations: The fetch cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- So long as these movements do not interfere with one another, several of them can take place during one step, saving time.
  - Symbolically, we can write this sequence of events as follows:

$$
\begin{aligned}
t_1: &\quad MAR \leftarrow (PC) \\
t_2: &\quad MBR \leftarrow Memory \\
&\quad PC \leftarrow (PC) + I \\
t_3: &\quad IR \leftarrow (MBR)
\end{aligned}
$$

  - where I is the instruction length.

Micro operations: The Indirect Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Indirect Cycle
- Once an instruction is fetched, the next step is to fetch source operands.
- let us assume a one-address instruction format, with direct and indirect addressing allowed.

- If the instruction specifies an indirect address, then an indirect cycle must precede the execute cycle

Micro operations: The Indirect Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Indirect Cycle
- The data flow includes the following micro-operations:

t_1: MAR ← (IR(Address))
t_2: MBR ← Memory
t_3: IR(Address) ← (MBR(Address))

  - t1: The address field of the instruction is transferred to the MAR.
  - t2: This is then used to fetch the address of the operand.
  - f3: The address field of the IR is updated from the MBR, so that it now contains a direct rather than an indirect address.

Micro operations: The Interrupt Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Interrupt Cycle
- At the completion of the execute cycle, a test is made to determine whether any enabled interrupts have occurred.
- If so, the interrupt cycle occurs.
- The nature of this cycle varies greatly from one machine to another

Micro operations: The Interrupt Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Interrupt Cycle
- We have:

```
t1: MBR ← (PC)
t2: MAR ← Save_Address
    PC ← Routine_Address
t3: Memory ← (MBR)
```

  - At t1, the contents of the PC are transferred to the MBR, so that they can be saved for return from the interrupt.
  - Then at t2, the MAR is loaded with the address at which the contents of the PC are to be saved, and the PC is loaded with the address of the start of the interrupt-processing routine.

Micro operations: The Interrupt Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Interrupt Cycle
  - The step at t3 is to store the MBR, which contains the old value of the PC, into memory. The processor is now ready to begin the next instruction cycle.

Micro operations: The Execute Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Execute Cycle
- The fetch, indirect, and interrupt cycles are simple and predictable.

- Each involves a small, fixed sequence of micro-operations and, in each case, the same micro operations are repeated each time around.

Micro operations: The Execute Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Execute Cycle
- This is not true of the execute cycle
  - Because of the variety of opcodes, there are a number of different sequences of micro-operations that can occur.

- The control unit examines the opcode and generates a sequence of micro- operations based on the value of the opcode.
  - This is referred to as instruction decoding.

Micro operations: The Execute Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Execute Cycle
- Let us consider one example:
  - First, consider an add instruction:
    - ADD R1, X
  - which adds the contents of the location X to register R1.

  - The following sequence of micro-operations might occur:

Micro operations: The Execute Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Execute Cycle

```
t1: MAR ← (IR(address))
t2: MBR ← Memory
t3: R1  ← (R1) + (MBR)
```

- We begin with the IR containing the ADD instruction
  - In t1, the address portion of the IR is loaded into the MAR
  - In t2, the referenced memory location is read.
  - In t3 the contents of R1 and MBR are added by the ALU

Micro operations: The Execute Cycle

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The Execute Cycle
  - This is a simplified example.
  - Additional micro- operations may be required to extract the register reference from the IR and perhaps to stage the ALU inputs or outputs in some intermediate registers

Control of the processor

- we have decomposed the behavior or functioning of the processor into elementary operations, called micro- operations.

- By reducing the operation of the processor to its most fundamental level, we are able to define exactly what it is that the control unit must cause to happen

Control of the processor

- Thus, we can define the functional requirements for the control unit: those functions that the control unit must perform.

- A definition of these functional requirements is the basis for the design and implementation of the control unit.

Control of the processor

- the following three-step process leads to a characterization of the control unit:
  - 1. Define the basic elements of the processor.
  - 2. Describe the micro-operations that the processor performs.
  - 3. Determine the functions that the control unit must perform to cause the micro-operations to be performed

- We have already done 1 and 2

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

Control of the processor

- To remind you, the basic functional elements of the processor are the following:
  - ALU
  - Registers
  - Internal data paths
  - External data paths
  - Control unit

Control of the processor

- The execution of a program consists of operations involving these processor elements and these operations consist of a sequence of micro-operations
  - all micro- operations fall into one of the following categories:
    - Transfer data from one register to another.
    - Transfer data from a register to an external interface (e.g., system bus).
    - Transfer data from an external interface to a register.
    - Perform an arithmetic or logic operation, using registers for input and output.

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

Control of the processor

- The control unit performs two basic tasks:
  - Sequencing: The control unit causes the processor to step through a series of micro- operations in the proper sequence, based on the program being executed.
  - Execution: The control unit causes each micro-operation to be performed.

Control of the processor: Control Signals

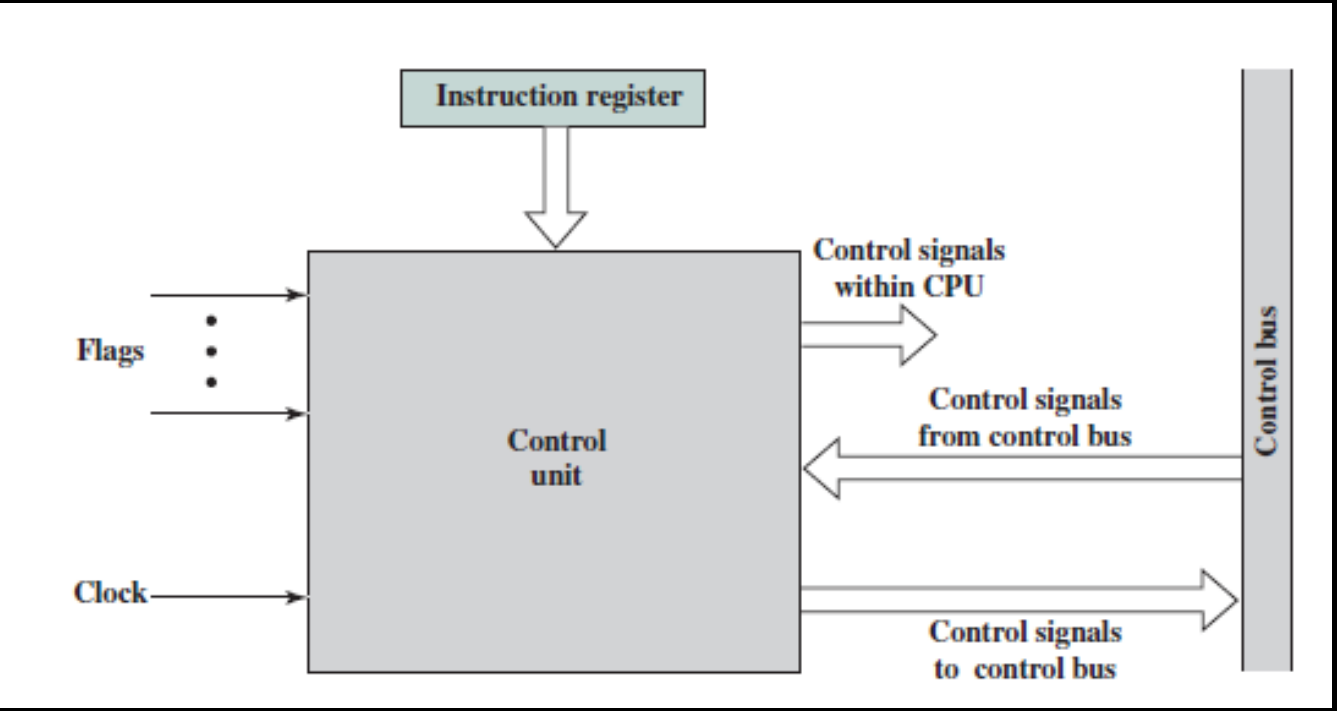
LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- For the control unit to perform its function, it must have :
  - inputs that allow it to determine the state of the system
  - outputs that allow it to control the behavior of the system
- The above are the external specifications of the control unit
- Internally, the control unit must have the logic required to perform its sequencing and execution functions

Control of the processor: Control Signals

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- Block Diagram of the Control Unit



Control of the processor: Control Signals

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The previous diagram shows a general model of the control unit, showing all of its inputs and outputs. The inputs are:
  - Clock:
    - This is how the control unit "keeps time." The control unit causes one micro- operation(or a set of simultaneous micro-operations) to be performed for each clock pulse.
    - This is sometimes referred to as the processor cycle time, or the clock cycle time

Control of the processor: Control Signals

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The inputs are:
  - Instruction register:
    - The opcode and addressing mode of the current instruction are used to determine which micro-operations to perform during the execute cycle.
  - Flags:
    - These are needed by the control unit to determine the status of the processor and the outcome of previous ALU operations.
    - For example, for the increment-and-skip-if-zero (ISZ) instruction, the control unit will increment the PC if the zero flag is set.

Control of the processor: Control Signals

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The inputs are:
  - Control signals from control bus:
    - The control bus portion of the system bus provides signals to the control unit.

Control of the processor: Control Signals

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- The outputs are as follows:
  - Control signals within the processor:
    - These are two types: those that cause data to be moved from one register to another, and those that activate specific ALU functions.
  - Control signals to control bus:
    - These are also of two types: control signals to memory, and control signals to the I/O modules.

Control of the processor: Hardwired Implementation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

- We have discussed the control unit in terms of its inputs, output, and functions.
- We now turn to control unit implementation.
- A wide variety of techniques have been used.

- Most of these fall into one of two categories:
  - Hardwired implementation
  - Microprogrammed implementation

LIMKOKWING UNIVERSITY COLLEGE OF CREATIVE TECHNOLOGY

THE END