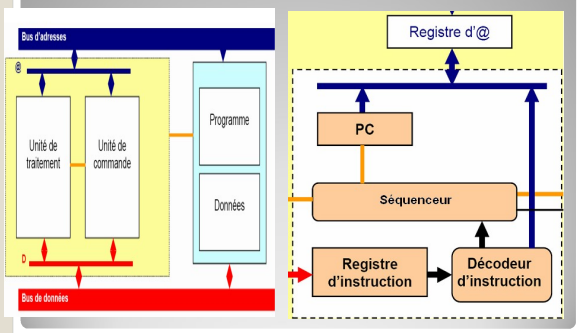
Chapter 5
Sequencers

Par
Pr. Ch. Bencheriet

Introduction

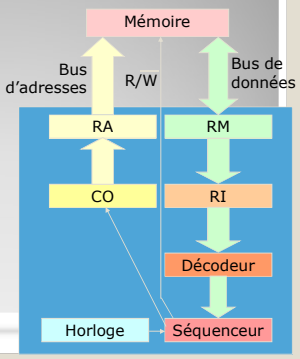
- The **Central Processing Unit (CPU)** or **Processor** is the core component of a computer that interprets and executes program instructions stored in main memory.
- The combination of the CPU and main memory forms the **Central Unit**.
- The CPU consists of the **Arithmetic Logic Unit (ALU)** and the **Control Unit**.
- The ALU performs arithmetic and logical operations.
- The Control Unit manages the operation of all other units: ALU, memory, input/output, etc., by providing them with timing and control signals.

Introduction



Introduction

This unit includes:
- The Program Counter (PC)
- The Instruction Register (IR)
- The Operation Code Decoder
- The Sequencer
- The Clock



Control Unit

- The pulses generated by the clock at regular intervals determine the machine's cycle time.
- The execution of an instruction typically takes more than one cycle because an instruction generally includes:
  - The instruction fetch time,
  - The instruction decode time,
  - The time to load operands and compute their effective address,
  - The actual execution time,
  - The result write-back time.

Control Unit

Étapes d'un cycle de recherche d'instruction (fetch) :
- Transfert de l'adresse de la nouvelle instruction de CO à RA.
  - ✓ La génération d'une impulsion de lecture par l'unité de commande provoque le transfert de l'instruction cherchée vers RM qui fonctionne comme registre tampon pour tous les échanges avec la mémoire.
- Transfert de l'instruction dans RI.
  - ✓ Instruction = code opération + adresse opérande
  - ✓ L'adressage de l'opérande peut demander le calcul de l'adresse effective, ce qui consomme des cycles machine.
  - ✓ Pendant que l'adresse de l'opérande est envoyée à RA, le code opération est transmis au décodeur qui détermine le type d'opération demandée et le transmet au séquenceur.
- Le CO est incrémenté en vue du cycle de recherche suivant.

Sequencer

Control Logic Block (or Sequencer):
It organizes the execution of instructions in sync with a clock. It generates all internal or external synchronization signals (control bus) of the microprocessor based on the instruction to be executed. It functions as a finite-state machine.

Sequencer

- The **sequencer** is a finite-state machine responsible for generating the control signals required to activate and manage the units involved in executing a given instruction.
- This function can be implemented in two ways: **hardwired sequencer** or **microprogrammed sequencer**.
- A hardwired sequencer is a complex sequential circuit that associates each instruction with a sub-circuit capable of controlling its execution.
- The same result can be achieved using a set of micro-instructions stored in a microprogram memory.
- This microprogram is capable of generating a series of control signals equivalent to those produced by a hardwired sequencer.

Sequencer

Hardwired Sequencer (RISC architecture):
This is a complex circuit that associates each executable instruction with a sub-circuit capable of controlling its execution. The sub-circuit is activated by a signal from the decoder.

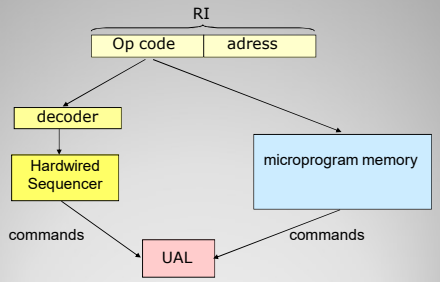
Microprogrammed Sequencer (CISC architecture):
Each instruction corresponds to a sequence of microinstructions stored in a very fast microprogram memory that is read-only. This memory can be of the ROM or non-volatile EEPROM type and is well-protected. The advantage of such a sequencer lies in its flexibility and simplicity, but its speed is slightly lower than that of a hardwired sequencer.

RISC/CISC Architecture

- Current general-purpose processors are divided into two main categories: **CISC** (Complex Instruction Set Computer) and **RISC** (Reduced Instruction Set Computer).
- Processors in these two categories differ in the design of their instruction sets.
- CISC processors have an extended set of complex instructions.
- Each of these instructions can perform multiple elementary operations, such as loading a value into memory, performing an arithmetic operation, and storing the result in memory.
- In contrast, RISC processors have a reduced instruction set where each instruction performs a single elementary operation.

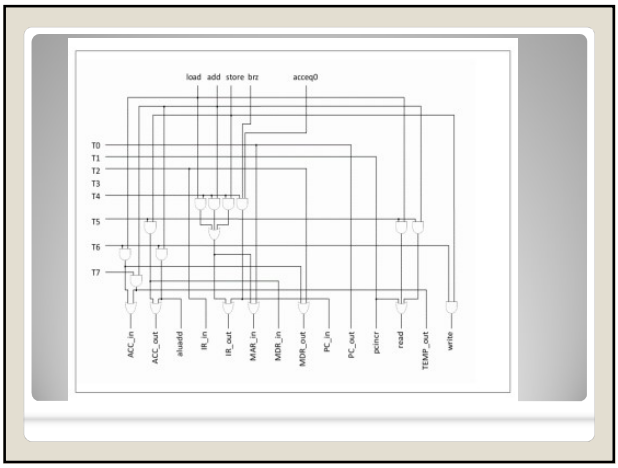
CISC	RISC
S/360 (IBM) VAX (DEC) 68xx, 680x0 (Motorola) x86, Pentium (Intel)	Alpha (DEC) PowerPC (Motorola) MIPS PA-RISC (Hewlett-Packard) SPARC

Hardwired Sequencer and Microprogrammed Sequencer



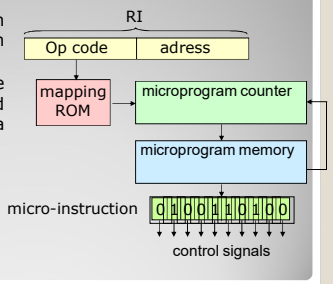
Hardwired Sequencer

1. Definition: Hardwired sequencers are hardware devices designed to perform specific tasks sequentially. They are configured using physical connections between the various components.
2. Operation: Hardwired sequencers are built using logic gates and flip-flops to create a sequence of instructions. Each step of the sequence is typically defined by physical electrical connections.
3. Limitations: These sequencers are inflexible and difficult to modify once they are wired. They are generally used for specific, fixed tasks.
4. Example: Early computers often used hardwired sequencers to perform predefined control tasks.



Microprogrammed Sequencer

- The microprogram can be stored in a ROM or an EPROM.
- This solution is more flexible than hardwired logic. The trade-off is a lower speed.



Microprogrammed Sequencer

1. **Definition:** Microprogrammed sequencers are based on a more flexible approach. They use a set of instructions stored in a special memory called microprogram memory.
2. **Operation:** Each instruction in the microprogram controls the sequencer to perform a specific operation. These instructions can be modified more easily than the physical connections in hardwired sequencers.
3. **Flexibility:** Microprogrammed sequencers offer greater flexibility because the microprogram can be modified without altering the physical hardware. This makes it easier to update instructions and adapt the sequencer to different tasks.
4. **Example:** Modern processors often use microprogrammed sequencers to execute instructions from complex instruction sets.

Microprogrammed Sequencer