# **Lecture Assignment – 3**

# **Name: Abdallah Mahmoud Amin ID:20211062**

**Q1: Explain what’s an instruction decoder with an example:-**

The Instruction Decoder is the piece in our microprocessor that is responsible for the decoding/translating our instructions, it does that by generating a number of signals that implements the execution of this instruction on its operands, the process of executing an instruction is done in several steps and in each step certain signal lines have to be activated, there are different ways to make these signals like: (Combinational logic, signal generator with a diode matrix).

We have two types of instruction decoders: RISC and CISC instruction decoders, they vary according to the architecture of our machine.

1. The RISC decoder has two variations:

* A variant in which the bits of the instruction are directly fed to the hardware which means there is no decoding at all, that variation needs to be very time-sensitive by careful programming of the machine code.
* The second variation includes a simple but fast decoder between the instruction code and the hardware components. This process is usually done by discrete logic gates, this allows a “cleaner” instruction set representation.

1. The CISC decoder:

It usually uses a state machine set up as random length sequencer, the decoder sequences through a number of states for the instruction code to determine which sequence shall be executed, the timing is controlled by the sequencer.

In general, the instruction decoder translates the IR (Instruction Register) value to a set of control signals, that signal is the one controlling which operation should be executed, the location of the source and destination operands in the RF to store our instruction’s result.

**A diagram of a microprocessor

Description automatically generated**

**Resources (** [**ScienceDirect**](https://www.sciencedirect.com/topics/engineering/instruction-decoder#:~:text=The%20instruction%20decoder%20decodes%20the,to%20store%20its%20operation%20result.) **,** [**Wikipedia**](https://en.wikibooks.org/wiki/Microprocessor_Design/Instruction_Decoder) **,** [**RelaisComputer**](https://www.relaiscomputer.nl/index.php/decoder) **)**

**Q2: compare RISC Instruction Decoder and CISC Instruction Decoder. How are they different?**

RISC & CISC are different approaches to processors’ architecture although they both aim to boost our CPU performance.

Now we’ll discuss how they are different using a simple example: in CISC we can add two numbers using a single simple command such as ADD, that will complete the task, meanwhile in RISC, we will have to load data into registers, then use a suitable operator and finally store the result in the desired destination.

**Major differences between RISC & CISC**

|  |  |
| --- | --- |
| **RISC** | **CISC** |
| Is a reduced instruction set computer. | Is a complex instruction set computer |
| The instruction must be stored in different register sets | We need a single register set to save the instruction |
| Instruction decoding is straightforward | Instruction decoding is complicated |
| Pipeline applications are simple, and the code is complex | The code is small, but the pipelining use is complex |
| It only uses hardwired control units | Uses hardwired and microprogrammed control units |
| Execution of an instruction is 1 clock cycle | Execution of an instruction is >1 clock cycle |
| Decoding instructions is simple | Decoding of instructions is complex |
| Doesn’t require an external memory for calculations | Requires an external memory for calculations |

The RISC decoder has two variations:

* A variant in which the bits of the instruction are directly fed to the hardware which means there is no decoding at all, that variation needs to be very time-sensitive by careful programming of the machine code.
* The second variation includes a simple but fast decoder between the instruction code and the hardware components. This process is usually done by discrete logic gates, this allows a “cleaner” instruction set representation.

The CISC decoder:

It usually uses a state machine set up as random length sequencer, the decoder sequences through a number of states for the instruction code to determine which sequence shall be executed, the timing is controlled by the sequencer.

A diagram of a computer program

Description automatically generated

**Resources (** [**Guru99**](https://www.guru99.com/risc-vs-cisc-differences.html#:~:text=In%20RISC%2C%20the%20decoding%20of,only%20a%20single%20register%20set.) **,** [**Shiksha**](https://www.shiksha.com/online-courses/articles/cisc-vs-risc/#:~:text=CISC%20vs%20RISC%20%2D%20Key%20Differences&text=Instruction%20decoding%20in%20RISC%20is,of%20a%20pipeline%20is%20complex.) **,** [**Chipedge**](https://chipedge.com/risc-vs-cisc-which-is-better/#:~:text=Major%20Distinctions%20Between%20CISC%20and%20RISC&text=Each%20instruction%20in%20a%20CISC,easy%20to%20decode%20and%20execute.) **)**