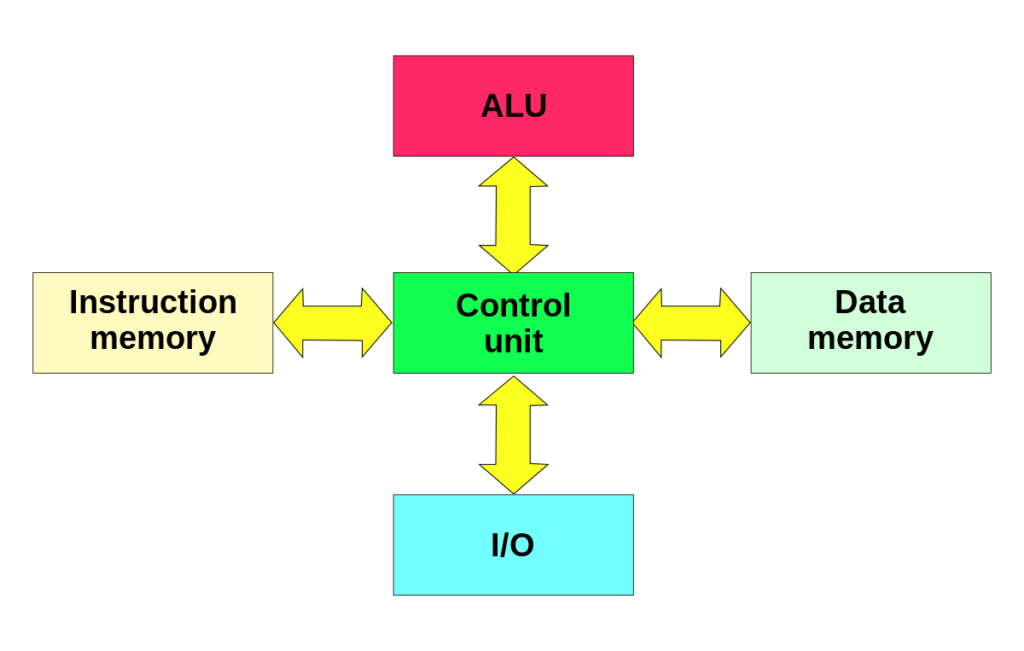
Lab Assignment: Computer Architecture

1. What are the major difference between the Von Neumann Computer Architecture and the Harvard Architecture?

**Von-Neumann architecture**:  
In a Von-Neumann architecture, the same memory and bus are used to store both data and instructions that run the program. Since you cannot access program memory and data memory simultaneously, the Von Neumann architecture is susceptible to bottlenecks and system performance is affected.

[](https://rh6stzxdcl1wf9gj1fkj14uc-wpengine.netdna-ssl.com/wp-content/uploads/2018/03/Harvard_architecture.svg_.png)

**Harvard Architecture**:  
The Harvard architecture stores machine instructions and data in separate memory units that are connected by different busses. In this case, there are at least two memory address spaces to work with, so there is a memory register for machine instructions and another memory register for data. Computers designed with the Harvard architecture are able to run a program and access data independently, and therefore simultaneously. Harvard architecture has a strict separation between data and code. Thus, Harvard architecture is more complicated but separate pipelines remove the bottleneck that Von Neumann creates.

1. What are the major building blocks in a general computer system?

**Ans:**

The Building Blocks of a Computer. Think of the motherboard as the computer's nervous system: It's a big slab of fiberglass etched with circuitry that connects each component of a computer together. Every piece of computer hardware will connect to the motherboard.

1. What does ISA stand for in Computer Architecture?

* Industry Standard Architecture (ISA) is the [16-bit](https://en.wikipedia.org/wiki/16-bit) internal [bus](https://en.wikipedia.org/wiki/Bus_(computing)) of [IBM PC/AT](https://en.wikipedia.org/wiki/IBM_Personal_Computer/AT) and similar computers based on the [Intel 80286](https://en.wikipedia.org/wiki/Intel_80286) and its immediate successors during the 1980s. The bus was (largely) [backward compatible](https://en.wikipedia.org/wiki/Backward_compatible) with the 8-bit bus of the [8088](https://en.wikipedia.org/wiki/Intel_8088)-based [IBM PC](https://en.wikipedia.org/wiki/IBM_Personal_Computer), including the [IBM PC/XT](https://en.wikipedia.org/wiki/IBM_PC/XT) as well as [IBM PC compatibles](https://en.wikipedia.org/wiki/IBM_PC_compatible).
* Originally referred to as the PC/AT-bus, it was also termed *I/O Channel* by IBM. The ISA term was coined as a [retronym](https://en.wikipedia.org/wiki/Retronym) by competing PC-clone manufacturers in the late 1980s or early 1990s as a reaction to IBM attempts to replace the AT-bus with its new and incompatible [Micro Channel architecture](https://en.wikipedia.org/wiki/Micro_Channel_architecture).

1. In roughly one English sentence, give a reason that it is better to have fewer registers in an instruction-set architecture.

**Ans:**

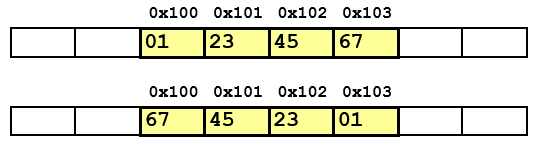
If an [operating system](https://en.wikipedia.org/wiki/Operating_system) maintains a standard and compatible [application binary interface](https://en.wikipedia.org/wiki/Application_binary_interface) (ABI) for a particular ISA, machine code for that ISA and operating system will run on future implementations of that ISA and newer versions of that operating system

1. In roughly one English sentence, give a reason that it is better to have many registers in an instruction-set architecture.

**Ans:**

ISA can be extended by adding instructions or other capabilities , or adding support for larger addresses and data values; an implementation of the extended ISA will still be able to execute machine code for versions of the ISA without those extensions. Machine code using those extensions will only run on implementations that support those extensions.

1. Variable x has 4-byte representation 0x01234567. The address is of x in memory starts at location 0x100. The value of x can be stored in two ways as shown below. Please identify the terminology associated with the corresponding representation.



**Ans:**

Big Endian: Most significant byte has lowest (first) address.

Little Endian: Least significant byte has lowest address.

1. Describe the steps that transform a program written in a high-level

language such as C into a representation that is directly executed by a computer processor.

* High level language consisted of words and algebraic notations. Each word in high level language for instance C++ is referred in computer language in 0s and 1s, Binary code.
* All instructions in a high level language is divided into operations, mnemonic, followed by a set of operators by a compiler.
* After the compiler converts high level language in to assembly language, assembler converts this language into binary code.

1. What are some of the great ideas in computer architecture? Please list them below (e.g. “Design for Moore’s Law”, “Use Abstraction to Simplify Design”.

|  |  |
| --- | --- |
| • | Moore's law |
| • | Abstraction |
| • | Common Case |
| • | Parallelism |
| • | Pipelining |
| • | Prediction |
| • | MemoryHierarchy |
| • | Redundancy |

1. Consider three different processors P1, P2, and P3 executing the same instruction set. P1 has a 3 GHz clock rate and a CPI of 1.5. P2 has a 2.5 GHz clock rate and a CPI of 1.0. P3 has a 4.0 GHz clock rate and has a CPI of 2.2.
2. Which processor has the highest performance expressed in instructions per second?
3. If the processors each execute a program in 10 seconds, find the number of cycles and the number of instructions.
4. We are trying to reduce the execution time by 30% but this leads to an increase of 20% in the CPI. What clock rate should we have to get this time reduction?

**Ans:**

1. P1: 3GHz / 1.5 = 2 \* 10^9 instructions per second P2: 2.5GHz / 1.0 = 2.5 \* 10^9 instructions per second P3: 4GHz / 2.2 = 1.82 \* 10^9 instructions per second So P2 has the highest performance among the three.

2. Cycles: P1: 3GHz \* 10 = 3 \* 10^10 cycles P2: 2.5GHz \* 10 = 2.5 \* 10^10 cycles P3: 4GHz \* 10 = 4 \* 10^10 cycles Num of instructions: P1: 3GHz \* 10 / 1.5 = 2 \* 10^10 instructions P2: 2.5GHz \* 10 / 1.0 = 2.5 \* 10^10 instructions P3: 4GHz \* 10 / 2.2 = 1.82 \* 10^10 instructions

3. Execution time = (Num of instructions \* CPI) / (Clock rate) So if we want to reduce the execution time by 30%, and CPI increases by 20%, we have: Execution time \* 0.7 = (Num of instructions \* CPI \* 1.2) / (New Clock rate) New Clock rate = Clock rate \* 1.2 / 0.7 = 1.71 \* Clock rate New Clock rate for each processor: P1: 3GHz \* 1.71 = 5.13 GHz P2: 2.5GHz \* 1.71 = 4.27 GHz P3: 4GHz \* 1.71 = 6.84 GHz