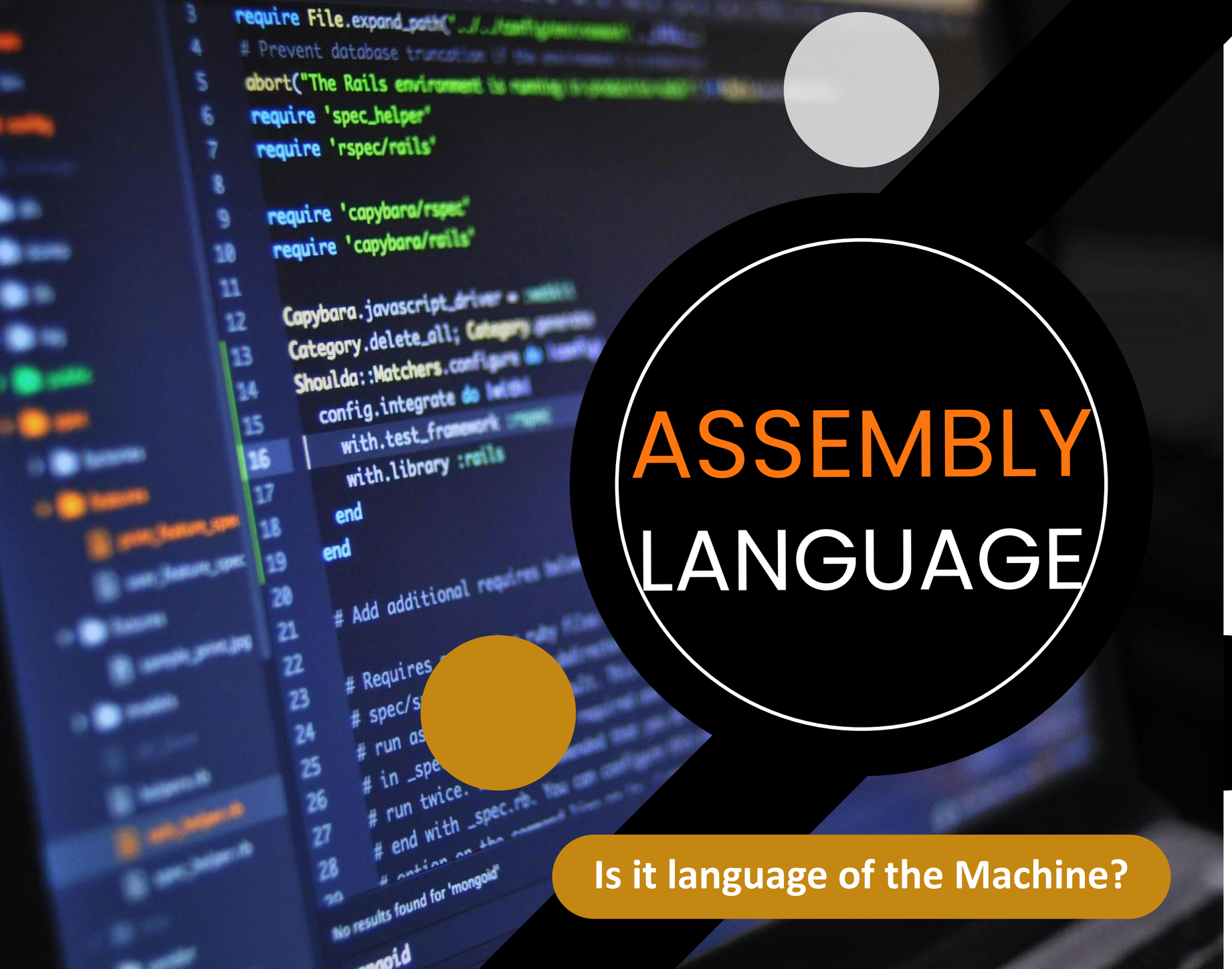
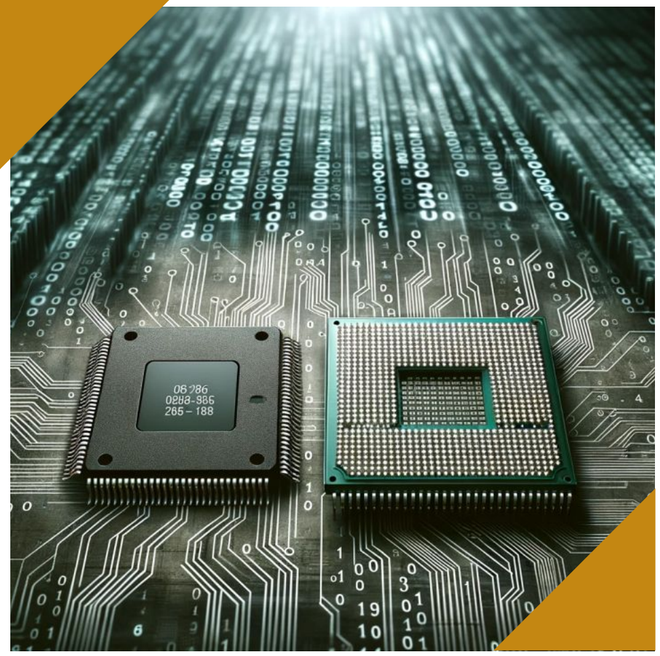


ASSEMBLY LANGUAGE

Is it language of the Machine?

Assembly Language



low-level programming language that is intended to communicate directly with a computer's hardware.

Unlike machine language, which consists of binary and hexadecimal characters, assembly languages are designed to be readable by humans.

Language 1 step removed from machine language

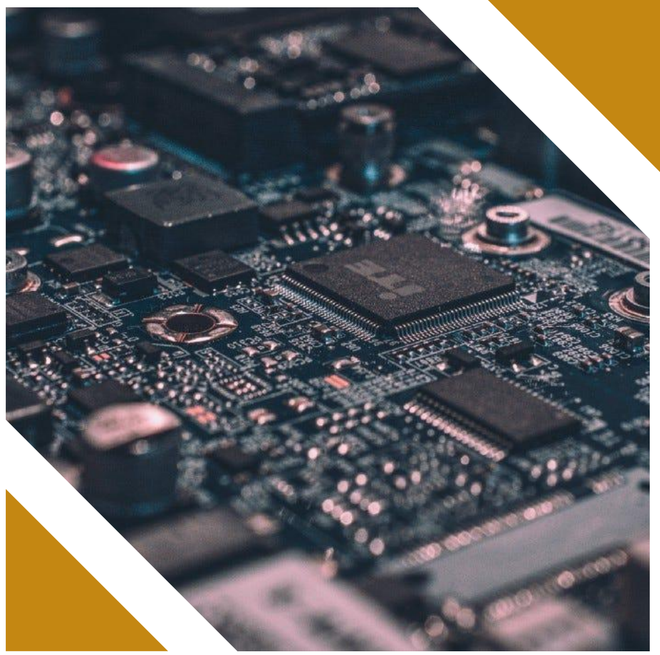
Translator or Bridge in between?

Low Level on a High Demand of attention

Low-level programming languages such as assembly language are a necessary bridge between the underlying hardware of a computer and the higher-level programming languages—such as Python or JavaScript—in which modern software programs are written.

An assembly language is a type of programming language that translates high-level languages into machine language. It is a necessary bridge between software programs and their underlying hardware platforms.

Both because its an assembler



How does it work?

the most basic instructions executed by a computer are binary codes

Those codes are directly translated into the “on” and “off” states of the electricity moving through the computer’s physical circuits.

these simple codes form the basis of “machine language,” the most fundamental variety of programming language.

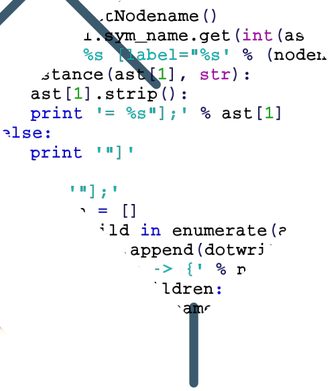
modern programmers issue commands in so-called “high-level languages,” which utilize intuitive syntax such as whole English words and sentences

Through Binary

Components

Of Assembly Language

programmers rely on assembly languages whose purpose is to automatically translate between these high-level and low-level languages.

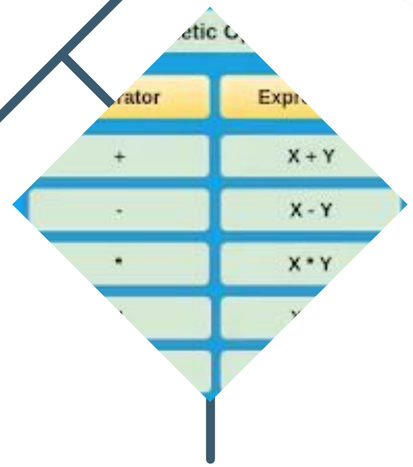


Syntax

When writing any code in any program language, there is an observable, specific order of rules that must be followed to allow a compiler to execute the code without error.

Label

A label is a symbol that represents the address where an instruction or data is stored. Destination reference to a statement.



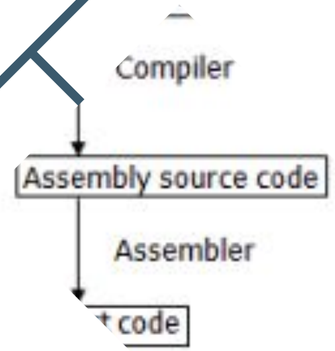
Operators

logical expressions that occur after the label field. In addition, it must be preceded by at least one white-space character.

Components

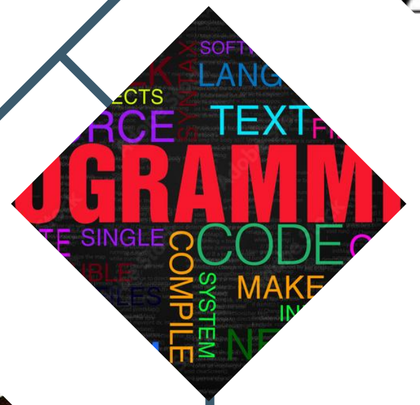
Of Assembly Language

Systemic approach on every programming languages.



Directive

Directives are instructions to the assembler that tell what actions must take place during the assembly process.

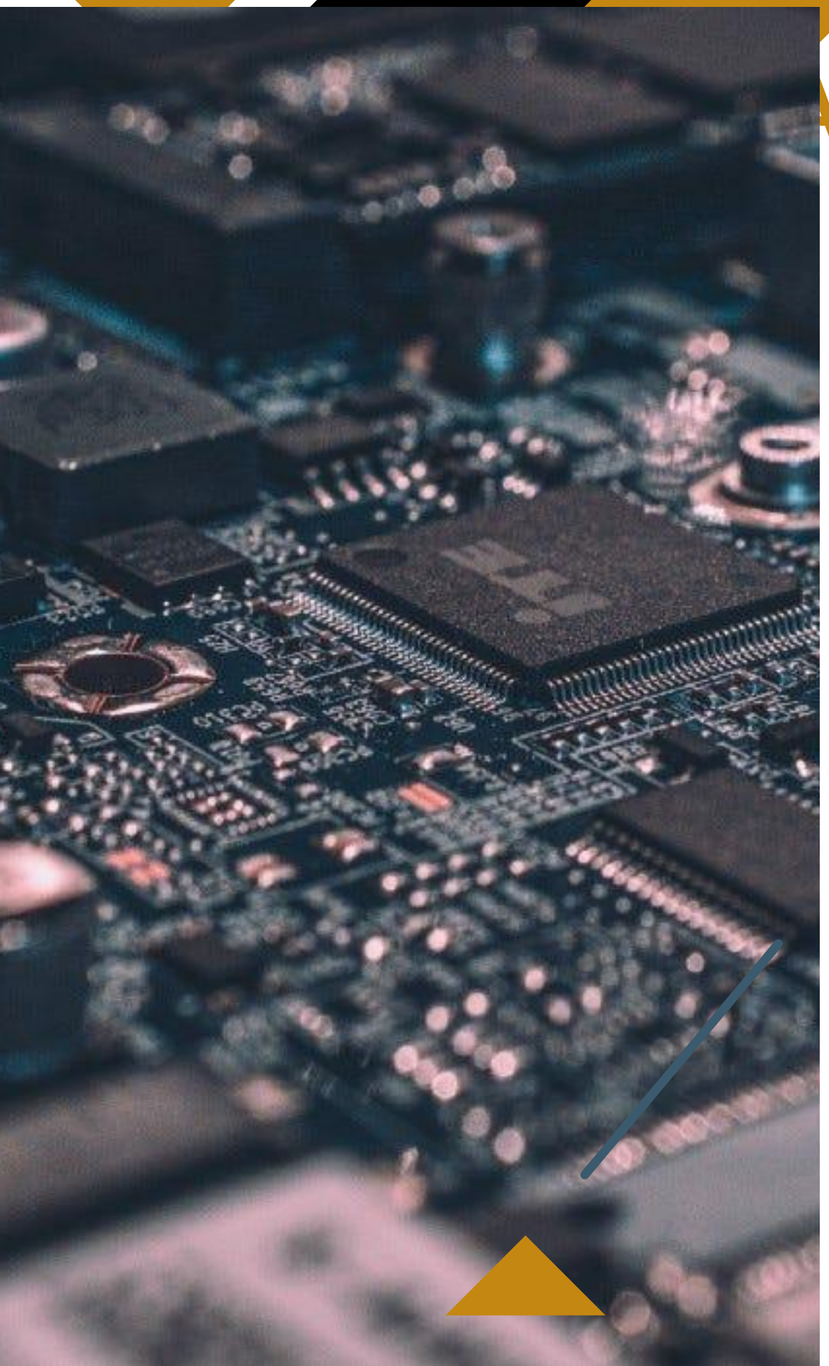


Macro

template shoe format presents a series or pattern of statements.
A word that stands for an entire group of instructions.

Mnemonic

is an abbreviation for an operation. entered into the operation code for each assemble program instruction to specify a shortened "opcode" that represents a larger, complete set of codes.



Assembly Programming

often used by programmers wanting greater control over their computers as assembly languages allow you to directly manipulate your hardware.

Pros

Execution may be more simple compared to other languages

Execution is usually faster compared to other languages

Allows for direct control over hardware

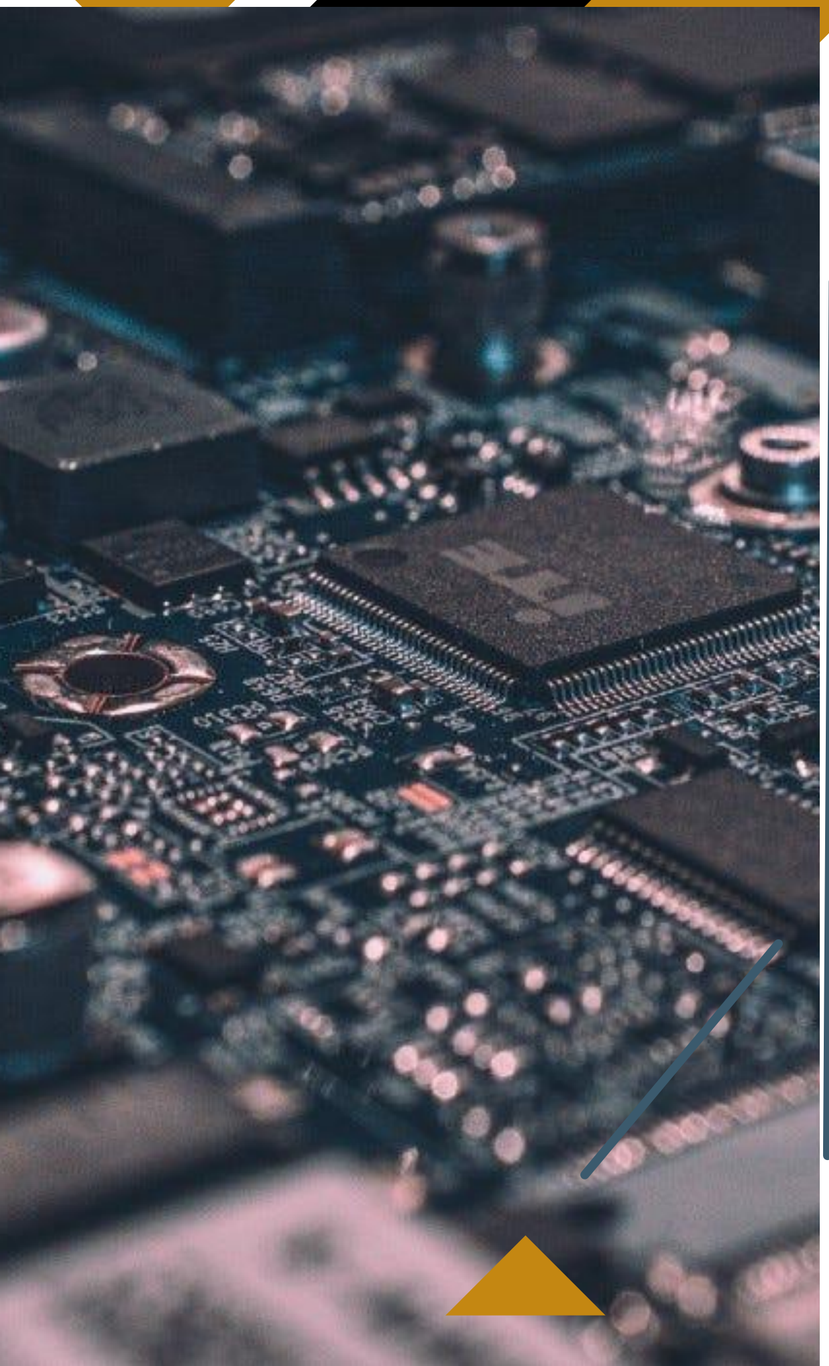
Code may remain smaller compared to other languages

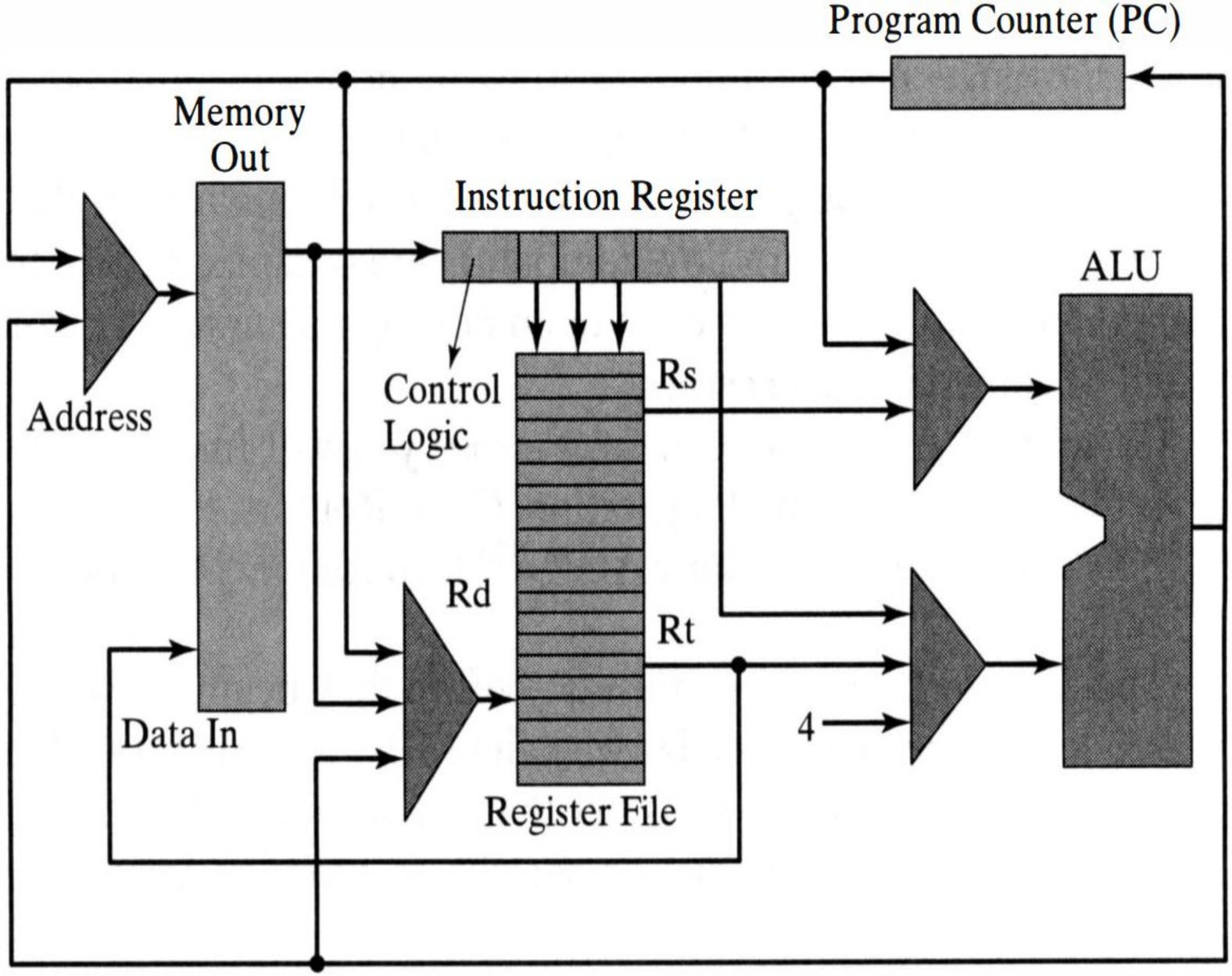
Cons

Programming may be more challenging to pick up compared to high-level languages

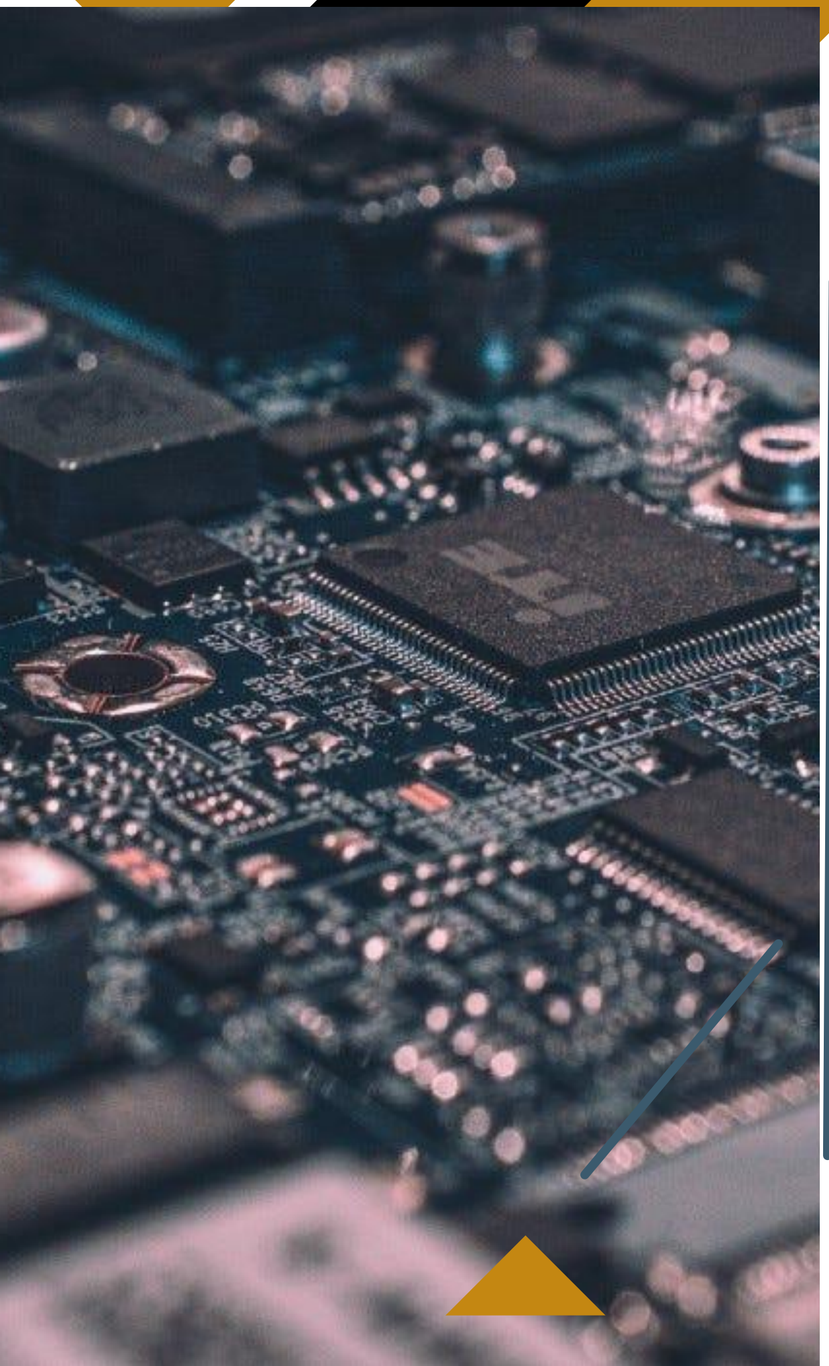
Syntax of assembly languages is difficult

Not portable between machines



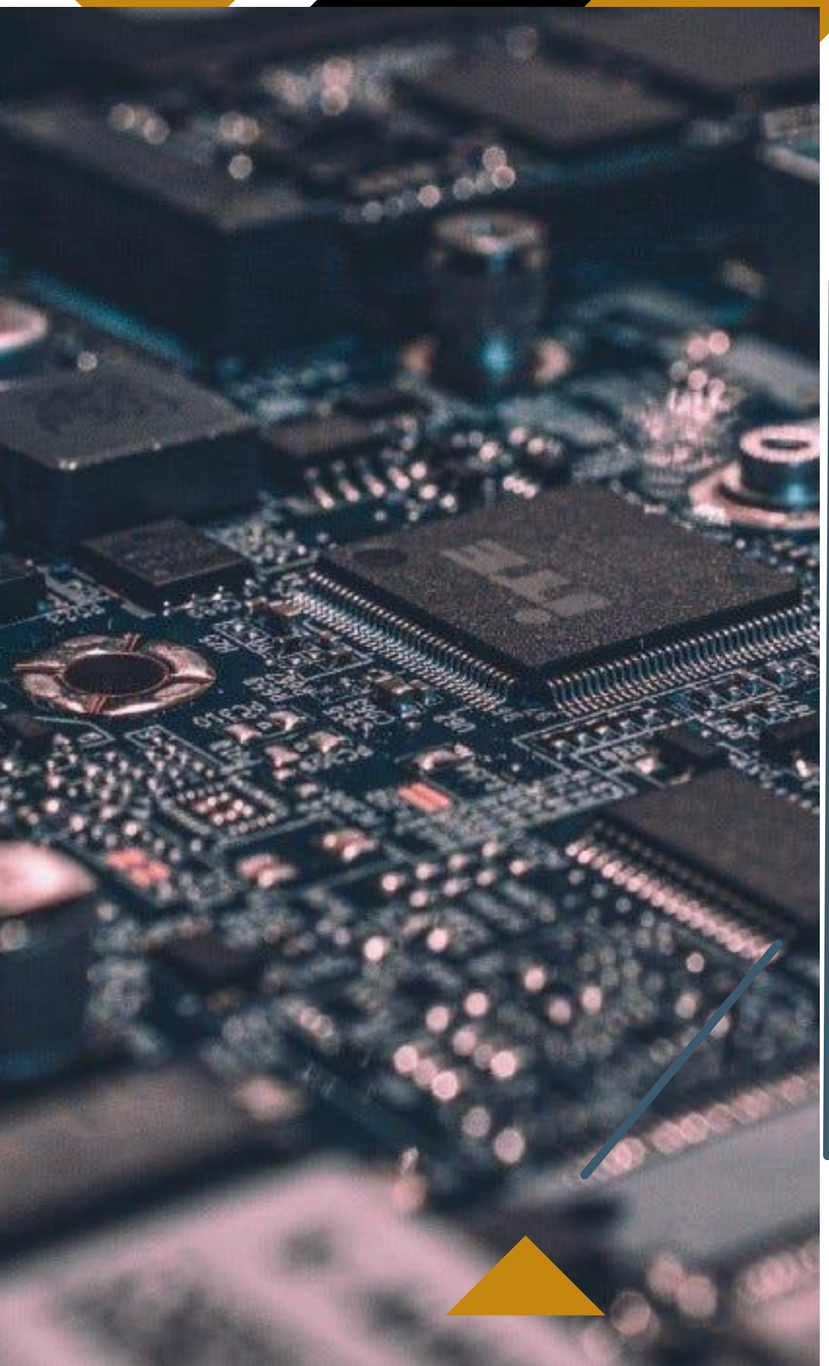


- control unit
- register file
- arithmetic and logic unit (ALU)
- program counter (PC)
- memory
- instruction register (IR)

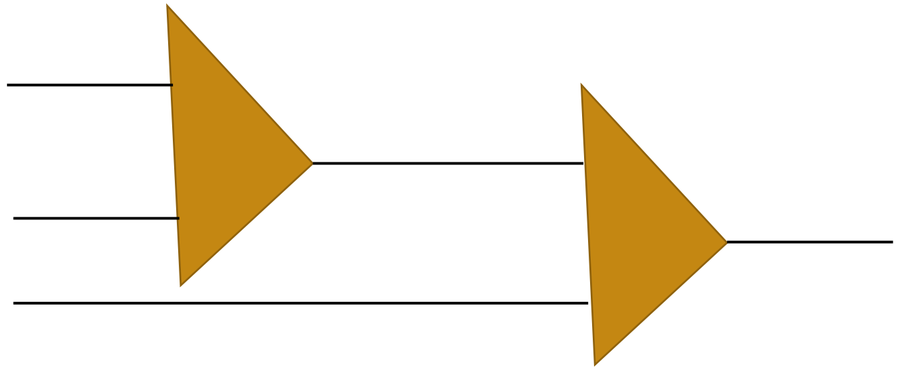


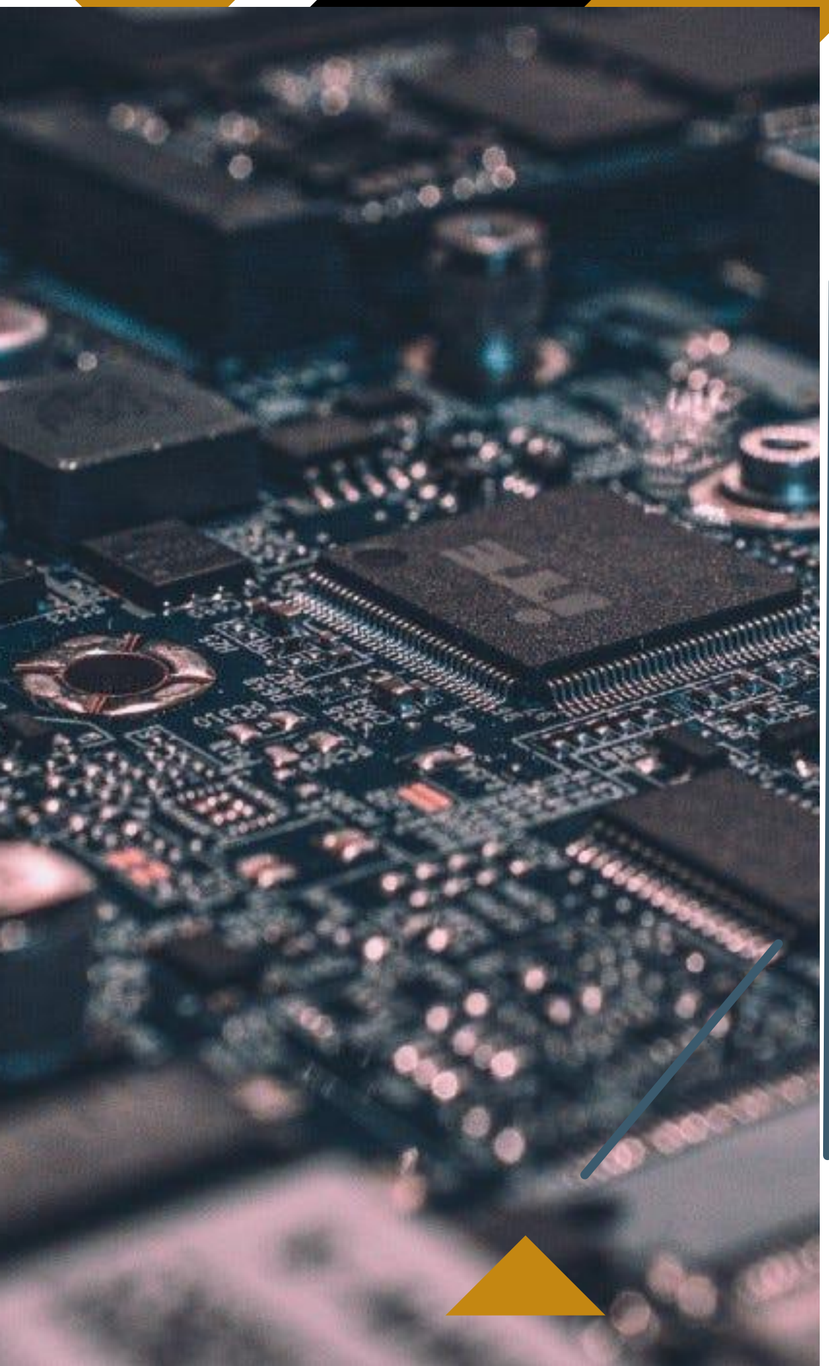
Interconnecting all of these functional components, except the control unit, are buses. A ***bus*** is nothing more than a set of electrical conducting paths over which different sets of binary values are transmitted.

MIPS architecture are 32 bits



we have the situation where we need to route information from more than one source to a destination, such as the ALU. One way to accomplish this at the hardware level is to use a multiplexer. ***Multiplexers*** are sometimes called ***data selectors.***





CONTROL UNIT

To fetch and execute instructions, control signals must be generated in a specific sequence to accomplish the task.

As you have already learned, data selectors must have control signals as inputs. Each register has an input control line, which when activated will cause a new value to be loaded into the register.

MIPS REGISTER FILE

Anyone who has ever used an electronic handheld calculator has experienced the fact that there is some electronic component inside the calculator that holds the result of the latest computation. This electronic storage component is called a register.

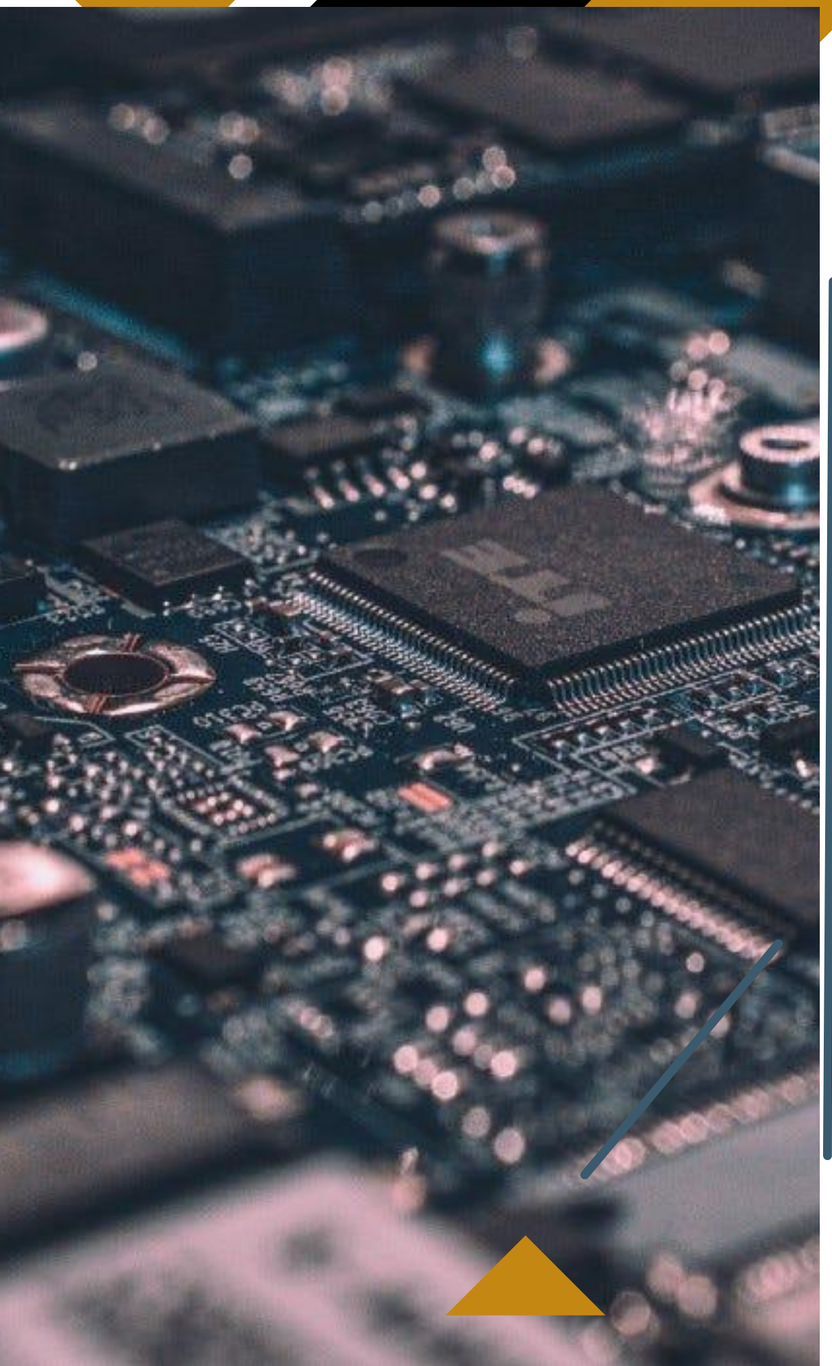
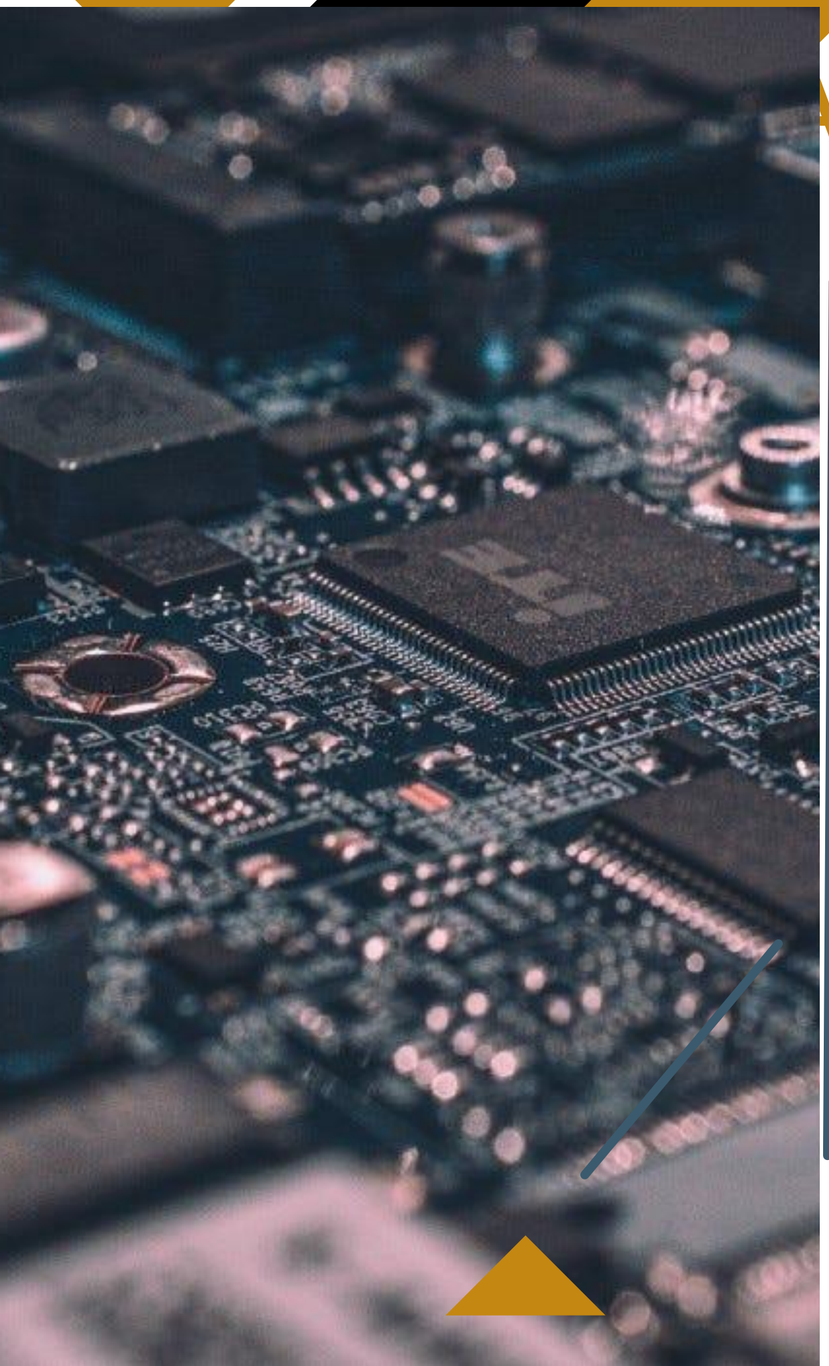


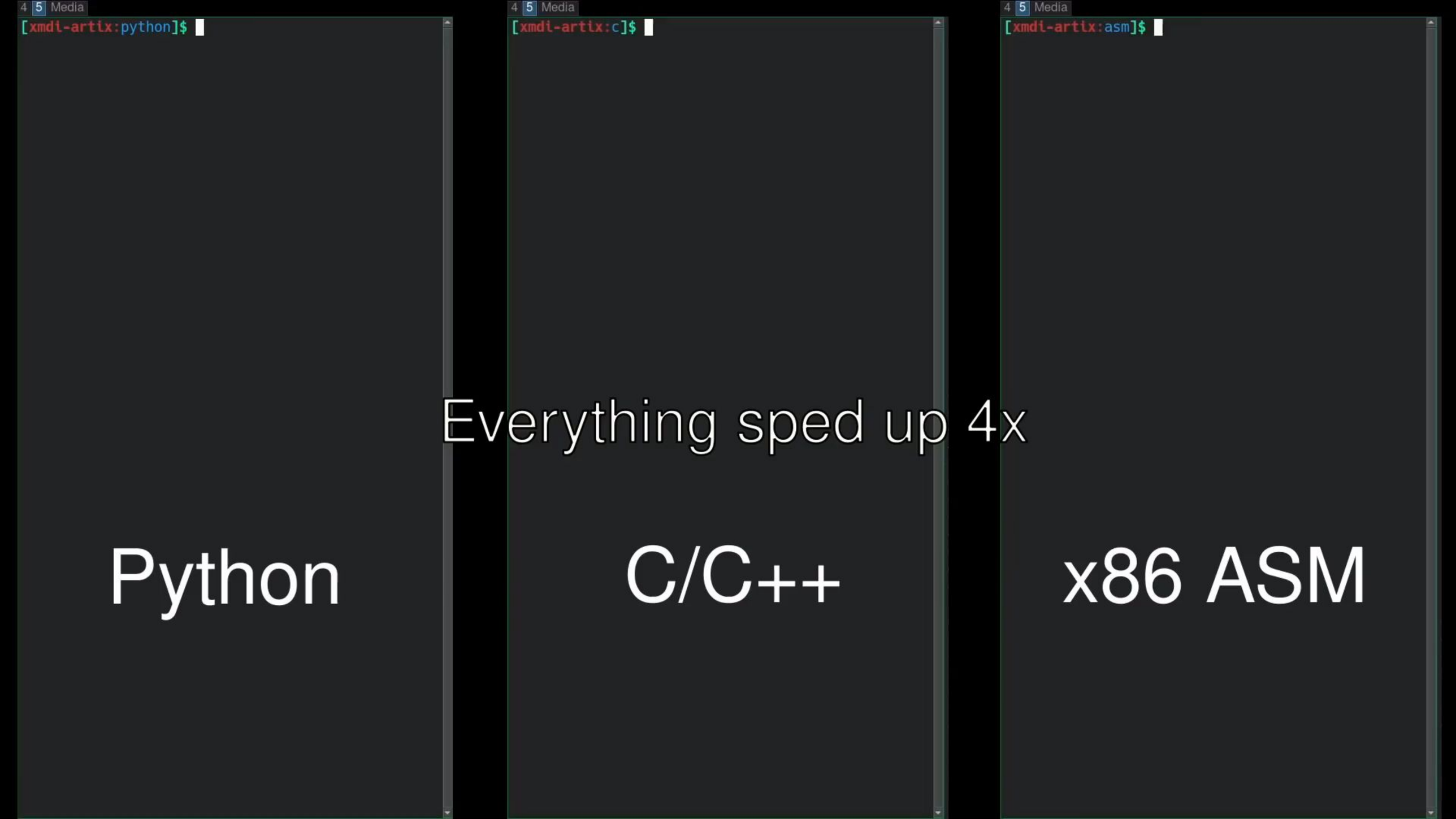
TABLE 1.1 The Register File

Register	Number	Usage
zero	0	Constant 0
at	1	Reserved for the assembler
v0	2	Used for return values from function calls
v1	3	
a0	4	Used to pass arguments to functions
a1	5	
a2	6	
a3	7	
t0	8	Temporary (Caller-saved, need not be saved by called functions)
t1	9	
t2	10	
t3	11	
t4	12	
t5	13	
t6	14	
t7	15	
s0	16	Saved temporary (Callee-saved, called function must save and restore)
s1	17	
s2	18	
s3	19	
s4	20	
s5	21	
s6	22	
s7	23	
t8	24	Temporary (Caller-saved, need not be saved by called function)
t9	25	
k0	26	Reserved for OS kernel
k1	27	
gp	28	Pointer to global area
sp	29	Stack pointer
fp	30	Frame pointer
ra	31	Return address for function calls



Assembly Programming

Service	Code in $v0	Argument(s)	Result(s)
Print Integer	1	$a0 = number to be printed	
Print Float	2	$f12 = number to be printed	
Print Double	3	$f12 = number to be printed	
Print String	4	$a0 = address of string in memory	
Read Integer	5		number returned in $v0
Read Float	6		number returned in $f0
Read Double	7		number returned in $f0
Read String	8	$a0 = address of input buffer in memory $a1 = length of buffer (n)	
Sbrk	9	$a0 = amount	address in $v0
Exit	10		


4 5 Media
[xmdi-artix:python]$
4 5 Media
[xmdi-artix:c]$
4 5 Media
[xmdi-artix:asm]$
Everything sped up 4x
Python
C/C++
x86 ASM

Assembly Programming

Assembly language is low-level code that relies on a strong relationship between the instructions input using the coding language and how a machine interprets the code instructions. Code is converted into executable actions using an assembler that converts input into recognizable instructions for the machine. Though prevalent in the early days of computing, many larger systems use higher-level languages.

Bottom Line