**Logo

Description automatically generated San Francisco Bay University**

**EE488 - Computer Architecture**

**Homework Assignment #2**

**Due day: 2/22/2023**

**Instruction:**

1. **Push the answer sheet to Github in word file**
2. **Overdue homework submission could not be accepted.**
3. **Takes academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**

**Github link:**

1. Discuss how stack architecture computer works by giving example. And compare the pros and cons between stack-based virtual machine and register-based virtual machine (1.5~2 pages)

**Answer:**

Stack computers perform their operations by using stack. A stack architecture computer is a kind a computer architecture. On stack-based computers, operands are pushed onto the stack and operations are performed on the top element of the stack. The outcome is removed from the stack after completing its operation cycle and then the result is removed from the stack.

On stack-based computers have short instructions and compiler is easy to write, operands are pushed onto the stack and operations are performed on the top element of the stack. The outcome is removed from the stack after completing its operation cycle and then the result is removed from the stack. The data structure is Firs-In Last-out data structures(FILO).Equation is A=B+(C\*D) by following operation: push B, push C, push D, mul, add and pop A. Diagram looks like below

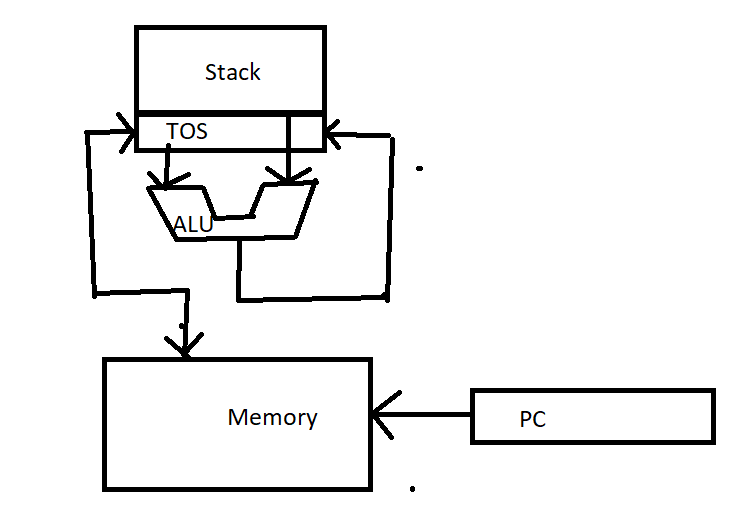


Figure: Stack Architecture

Numerous embedded systems and microcontrollers employ this kind of architecture because they can be implemented using a small number of transistors and require minimal memory. Nevertheless, they can also be employed in larger systems, such as virtual machines, where they can provide an efficient way to execute code. On top of that, the easy and foreseeable nature of RISC architecture makes it easier to adjust code for performance. This can be especially significant in applications that need a real-time performance, such as control systems or robotics.

Forth language is an example of a stack-based computer, which is used in many embedded systems. Forth is a stack-based language, where all operations are performed on a stack. For instance, the addition of two numbers would be done by pushing the two numbers onto the stack and then popping them off to add them. The result is then pushed back onto the stack. HP3000/70 and Burroughs B550/6500 also some examples.

In contrast, a register-based virtual machine stores data in registers rather than on a stack. In a register-based virtual machine, the operands are stored in registers and the operations are performed on those registers. The result is then stored back in a register. The IBM 7090 and DEC PDP-8 are the specimens of register based virtual machines.

The instructions: -ALU(Acc<-ACC+\*M –Load to accumulator(Acc<-\*M) –store from accumulator(\*M<-Acc. Equation is A=B+(C\*D) by following operation: load C, mul D, add B, store A. Diagram looks like below:

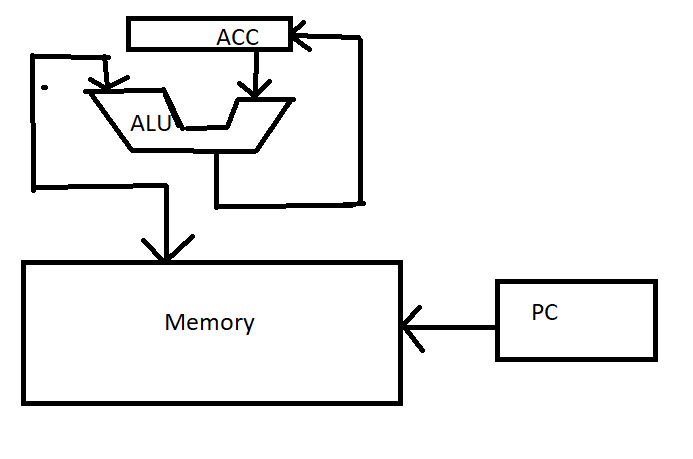


Figure: Accumulator Architecture

There is a choice of trade-offs between stack-based and register-based virtual machines. Stack-based virtual machines have a smaller instruction set, which makes them simpler to implement and faster to execute. They also have a smaller memory footprint, which is important in embedded systems. However, stack-based virtual machines are less efficient in terms of memory access, as they require more memory operations to access the operands.

On the other side, register-based virtual machines are well-organized in terms of memory access, as the operands are stored in registers. They are also more proficient in terms of instruction execution time, as the instructions can directly access the registers. However, they have a larger instruction set and a larger memory footmark, which makes them more complex to implement and slower to implement.

Overall, stack-based virtual machines and register-based virtual machines both have their advantages and disadvantages. The choice between them depends on the specific requirements of the system being implemented. For embedded systems with limited memory and simpler requirements, stack-based virtual machines may be the better choice. For more complex systems where efficiency is important, register-based virtual machines may be the better choice.

1. Processor is one of most important components in computing system. Its performance can make big impact to the whole system. Discuss about processor design metrics and benchmarking tools (1.5~2 pages)

**Answer:**

For all computing systems, the performance of a processor is a serious aspect and is influenced by several key design metrics. These metrics include clock speed, instruction set architecture (ISA), cache size, number of cores, and power consumption.

Processor design metrics are the key performance indicators that are used to evaluate the efficiency and effectiveness of a processor. Based on these metrics, benchmarks can be classified into two categories: microarchitectural metrics and macroarchitectural metrics.

Microarchitectural insights into the underlying performance factors and its not a good predictor of application performance .Microarchitectural metrics are used to evaluate the performance of individual processor components, such as the arithmetic and logic unit (ALU), cache, and memory. These metrics include:

Latency: This is the time it takes for a processor to complete a single operation, such as a load or a store.

Throughput: This is the number of operations a processor can complete per unit of time.

Instruction per cycle (IPC): This metric measures the average number of instructions executed per clock cycle.

Cache hit rate: This is the percentage of memory accesses that are satisfied by the cache.

Branch misprediction rate: This is the percentage of times the processor incorrectly predicts the outcome of a branch instruction.

Macroarchitectural measures overall performance, but on just one application and need application suite. Macroarchitectural metrics are used to evaluate the performance of the processor as a whole, taking into account all the components and their interactions. These metrics include:

Performance: This is the overall speed of the processor, measured in terms of instructions per second or cycles per second.

Energy efficiency: This is the amount of energy consumed by the processor per unit of work done.

Scalability: This is the ability of the processor to handle increasing workloads by adding more processors.

Reliability: This is the ability of the processor to perform correctly and consistently over time.

Microbenchmarks are small, targeted programs designed to measure the performance of a specific component or feature of a processor. These benchmarks are used to evaluate the latency, throughput, and IPC of individual processor components.

Macrobenchmarks, on the other hand, are larger, more complex programs that are designed to simulate real-world workloads. These benchmarks are used to evaluate the overall performance and energy efficiency of the processor, as well as its ability to handle complex workloads and real-world applications.

Overall, processor design metrics play a key role in measuring the performance, efficiency, and effectiveness of a processor. Microarchitectural metrics emphasis on individual processor components wise , while macroarchitectural metrics evaluate the entire processor’s performance. Microbenchmarks are used to measure the performance of individual components, while macrobenchmarks evaluate the performance of the processor as a whole, in the context of real-world workloads. storage and graphics processing. The key design metrics and benchmarking tools can play a huge role while selecting a processor.