**Northwestern Polytechnic University**

**EE488 - Computer Architecture**

**Homework Assignment #2**

**Due day: 10/6/2021**

**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)**

***1. Discuss how stack architecture computers work by giving examples. And compare the pros and cons between stack-based virtual machine and register-based virtual machine (1.5~2 pages)***

**Stack Architecture:**

The computers which use Stack-based CPU architecture are based on a data structure called stack. It uses the most popular method Last In First Out (LIFO) to access which is used in most of the CPU. A register is used to store the address of the topmost element of the stack which is known as Stack pointer (SP). In this architecture, ALU operations are performed on stack data. It means both the operands are always required on the stack. After calculation, the result is placed in the stack.

The main two operations that are performed on the operators of the stack are **Push** and **Pop**. These two operations are performed from one end only, that is the top of stack.

**Push:**   
This operation inserts one operand at the top of the stack and decreases the stack pointer register. The format of the PUSH instruction is:

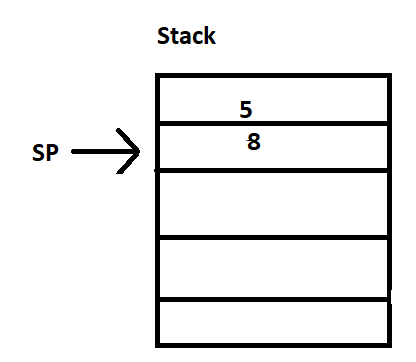
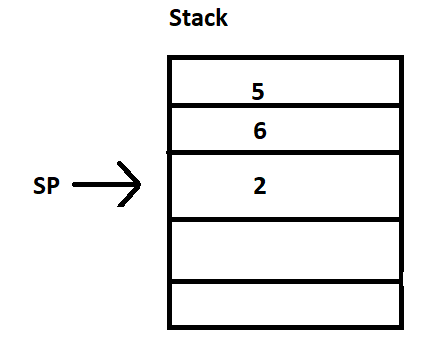
**Pop:**  
This operation results in deleting one operand from the top of the stack and it increases the stack pointer register.

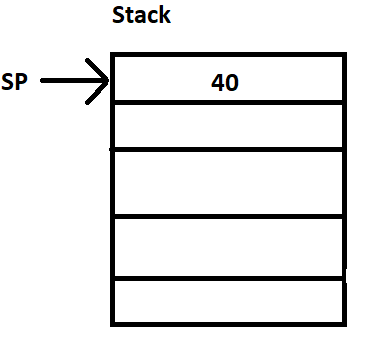
**Example:**

5 \* ( 6 + 2 ) which is equivalent to 562+\*

While solving an expression using stack, we convert the normal expression to a postfix expression for easy access.

All the operands are pushed onto the stack and when an operator is encountered, the last input is popped and added to the operator and then another input is popped out, evaluates and pushes the result back to the stack.





When the operator ‘+’ is encountered, 2 is popped and it forms 2+ and then 6 is popped resulting in 2+6 and pushes the result 8 to the stack.

**Pros of Stack based CPU architecture:**

* Computation of complex arithmetic expressions is efficient.
* As operand data are stored in consecutive memory locations, execution of instructions is faster.
* They do not have an address field which results in a short length of instruction .

**Cons of Stack based CPU architecture:**

* The size of the program increases.

# **Register based CPU Architecture:**

In this type of architecture, the computer uses two or three address fields in their instruction format. Each address field may specify a general register or a memory word. If many CPU registers are available for heavily used variables and intermediate results, we can avoid memory references much of the time, thus vastly increasing program execution speed, and reducing program size.

**Example:**

MULT R1, R2, R3

Consider an arithmetic multiplication written in assembly language. It uses three address fields R1, R2, and R3. The meaning of this instruction is: R1 = R2 \* R3

This instruction also can be written using only two address fields as:

MULT R1, R2

In this instruction, the destination register is the same as one of the source registers which is : R1 = R1 \* R2

The use of a large number of registers results in a short program with limited instructions.

**Pros of Register-based CPU architecture:**

* As a large number of registers are used in this organization, efficiency of CPU increases.
* Less memory space is used to store the program since the instructions are written in a compact way.

**Cons of Register based CPU architecture:**

* Make sure to avoid unnecessary usage of registers which requires compilers to be more intelligent in this aspect.
* Since a large number of registers are used, thus extra cost is required

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

Performance is the amount of work accomplished by a computer system. It basically depends on response time, throughput and execution time of a computer system.

**Response time:** It is the time from start to completion of a task which includes:

* Operating system overhead.
* Waiting for I/O and other processes
* Accessing disk and memory
* Time spent executing on the CPU or execution time.

**Throughput** is the total amount of work done in a given time.

**Execution time** is the total time a CPU spends on a given task. This is also referred to as simply CPU time.

Performance is determined by execution time as performance is inversely proportional to execution time.

Performance = (1 / Execution time)

(Performance of A / Performance of B)

= (Execution Time of B / Execution Time of A)

If given that Processor B is faster than processor A, that means execution time of B is less than that of execution time of A. Therefore, the performance of B is greater than that of A.

Execution time = CPU clock cycles x clock cycle time

Since clock cycle time and clock rate are reciprocals, so,

Execution time = CPU clock cycles / clock rate

The number of CPU clock cycles can be determined by,

CPU clock cycles

= (No. of instructions / Program ) x (Clock cycles / Instruction)

= Instruction Count x CPI

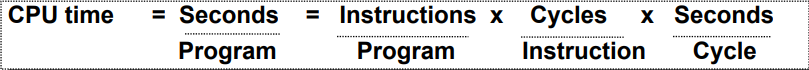
Which gives,

Execution time

= Instruction Count x CPI x clock cycle time

= Instruction Count x CPI / clock rate

The units for CPU Execution time are:



**Methods to improve Performance:**

To improve performance you can either:

* Decrease the CPI by using new Hardware.
* Decrease the clock time or Increase clock rate by reducing propagation delays or by using pipelining.
* Decrease the number of required cycles or improve ISA or Compiler.

**Benchmarking tools:**

To evaluate the performance of a computer system, we need a standard set of programs. Individuals cannot use their own programs that will favour their design enhancements and report improvements. So benchmarks are a set of programs that form a “workload” specifically chosen to measure performance.

**Microbenchmarks:**

* Measure one performance dimension - Cache bandwidth - Memory bandwidth - Procedure call overhead - FP(Floating Point) performance.
* Insight into the underlying performance factors.
* Not a good predictor of application performance

**Macro Benchmarks:**

* Application execution time - Measures overall performance, but on just one application - Need application suite.

**Tools:**

* One of the most successful attempts to create standardized benchmark application suites has been the SPEC (Standard Performance Evaluation Corporation) to deliver better benchmarks for workstations. SPEC is a non-profit corporation formed to establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers. SPEC develops benchmark suites and also reviews and publishes submitted results from the member organizations and other benchmark licensees.Just as the computer industry has evolved over time, so has the need for different benchmark suites, and there are now SPEC benchmarks to cover different application classes.