Trevor Pope & Micheal Callahan

CST-405

Dr. Isac Artzi

12/15/2024

**Benchmark – Project 6: Complete Compiler**

**Compilation Time**

For our project, compilation time is calculated using the “time make” command in our Linux shell. The “time make” command measures the duration of the build process when using a Makefile. This command records three types of time: real, user, and sys. The real time reflects the total wall-clock time it takes to complete the compilation, including any time the system spends waiting for resources or processes. The user time represents the amount of CPU time spent in user mode, performing tasks like compiling source code. The sys time accounts for the CPU time spent in kernel mode, handling system-level operations such as file I/O. By analyzing these metrics, developers can assess the efficiency of their compilation process and identify potential bottlenecks.



Our compilation time for our testProg.cmm file was 0.980s which indicates that our compiler is relatively optimized for processing various constructs, such as functions, loops, nested conditionals, and array handling.

**Execution Time**

Our execution time is calculated using the syscall associated with MARS. We can run “i $v0, 30” and then “syscall” to get the current system time. We can do that at the beginning of our MIPS code and the end of our MIPS code to get the start and end times. We then can subtract those times and print the result to tell us our execution time in seconds. We divided the time by 1000 to tell us the execution time in milliseconds. Our testProg.cmm file runs in 0.0 milliseconds, executing with immense speed. We can also use the MARS simulator to determine the number of clock cycles ran during our program execution. For our testProg.cmm we see the following:

A screenshot of a computer

Description automatically generated

We are going to assume a simple multi-cycle MIPS architecture:

**Instruction Type --- CPI (Cycles Per Instruction)**

R-Type (Arithmetic/Logical): 1 cycle

I-Type (Load/Store): 2 cycles

J-Type (Branch/Jump): 2 cycles

Clock Frequency = 1 GHz (1,000,000,000 Hz)

Total Cycles = (# of R-Type \* R-Type Cycles) + (# of I-Type \* I-Type Cycles) + (# of J-Type \* J-Type Cycles)

Total Cycles = (231 \* 1) + (907 \* 2) + (44 \* 2) = 2133 Cycles

Execution Time = Cycles / Clock Frequency

Execution Time = 2133 Cycles / 1,000,000,000 Hz = 0.000002133 Seconds