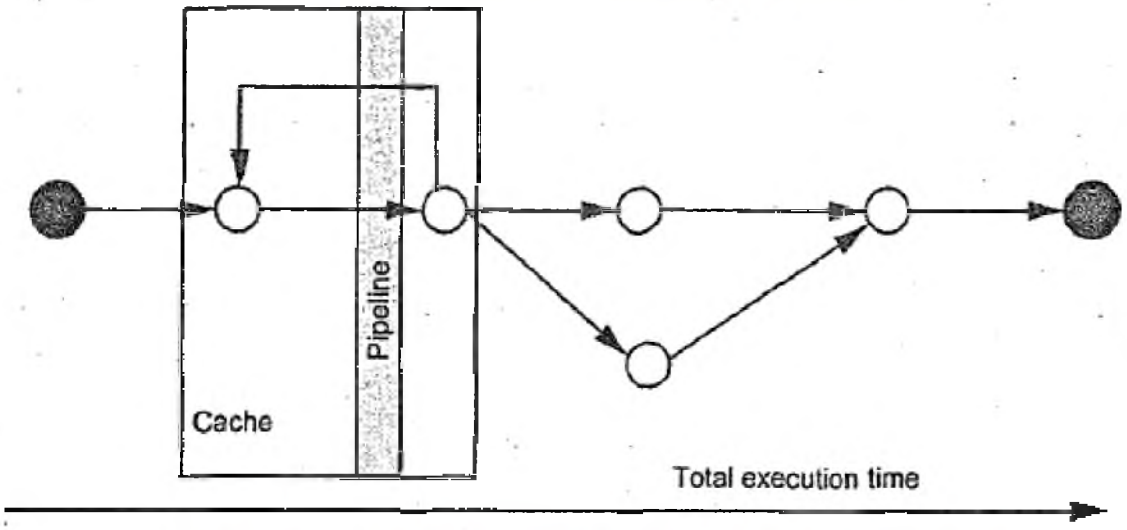
PROGRAM LEVEL PERFORMANCE ANALYSIS

Introduction

Embedded systems helps to perform functions in real time, we often need to know how fast a program runs. The techniques we use to analyze program execution time are also helpful in analyzing properties such as power consumption.



Consider the CPU pipeline and cache act as windows into our program. To understand the total execution time of our program, we must look at an execution paths, which in general are far longer than the pipeline and cache windows.

The pipeline and cache influence execution time, but execution time is a global property of the program. It is very difficult to calculate the execution time of **programs because of the following reasons:**

(i) The execution time of a program often varies with the input data values because those values select different execution paths in the program.'

(ii) The cache has a major effect on program performance but the cache's behavior depends in part on the data values input to the program.

(iii) Execution times may vary even at the instruction level. In general, the' execution time of an instruction in a pipeline depends not only on that instruction but also on the instructions around the pipeline.

Measuring Execution Speed

We can measure program performance in several ways:

(i) Some microprocessor manufacturers supply simulators for their CPUs which can runs on a workstation or PC that takes as input datas, and simulates the execution of that program. Some of these functional simulations also measure the execution time of the program.

(ii) A timer connected to the microprocessor bus can be used to measure the performance of executing sections of a code.

(iii) A logic analyzer that can be connected to the microprocessor bus is used to measure the start and stop times of a code segment.

Three different types of performance measures on programs are,

(i) Average- Case Execution Time (ACET):

This execution time would expect for typical data. First, we clearly define the typical inputs.

(ii) Worst-Case Execution Time (WCET):

This is the longest time that the program can spend on any input sequence We can check whether the system meets the deadlines.

(iii) Best-Case Execution Time (BCET):

It is the shortest time that the program can spend on any input sequence. T h:s measure can be important in multirate real-time systems.

Elements of Program Performance

Program execution time can be simply expressed as,

Execution Time = Program Path + Instruction Timing

The path is the sequence of instructions executed by the program. The instruction timing is determined based on the sequence of instructions traced by

the program path, which takes into account the data dependencies, pipeline behavior, and caching.

(1) Instruction Timing :

Once we know the execution path of the program, we have to measure the execution time of the instructions that are executed along that path.

The simplest estimate is to assume that every instruction takes the same number of clock cycles, which means we need only to count the instructions and multiply by the execution time of one instruction to obtain the program's total execution time.

Measurement-Driven Performance Analysis

(1) Program Traces:

Most of the methods that measures the program performance by combining the determination of the execution path and the timing of that path.

When the program executes, it chooses a path and we can observe the execution time along that path.

(2) Measurement Issues :

One of the challenging problems in measuring program performance is to find out the useful set of inputs to give the program.

First, we have to determine the actual input values. By using a simple program, we may be able to analyze the algorithm to determine the inputs that cause the worst-case execution time.

(3) Profiling :

Profiling is a simple method for analyzing the software the performance. It is achieved by instrumenting either the program source code or its binary executable form using a tool called a profiler (or) code profiler.

Profilers may use a number of different techniques, such as event-based, statistical, instrumented, and simulation methods.

A profiler does not measure execution time instead, it counts the number of times those procedures or basic blocks in the program are executed.

(4) Hardware Traces:

Some CPUs have hardware facilities for automatically generating the trace information. If we record only traces, we can reconstruct the instructions executed within the basic blocks while greatly reducing the amount of memory required to hold the trace.

(5) Simulation Based Performance Measurement

A CPU simulator is a program that takes an input as memory image for a CPU and performs the operations on that memory image that the actual CPU would perform, leaving the results in the modified memory image.