UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

LABORATORIO 3 ORGANIZACIÓN DE COMPUTADORES: PIPELINE

HERNÁN OLMEDO

Profesores:
Felipe Garay,
Erika Rosas,
Nicolás Hidalgo
Ayudante:
Ian Mejias

Santiago - Chile
13 de octubre de 2015

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

CAPÍTULO 1. INTRODUCCIÓN

Hoy en día se vive en un mundo globalizado, donde la información se propaga rápidamente por el mundo en todo ámbito, ya sea industrial, redes sociales, economía, investigación, etc. La informática juega un papel crucial, siendo la ciencia encargada de proveer los medios a través de los cuales se propaga y procesa dicha información, todas las actividades ligadas al uso de medios informáticos exigen la mayor rapidez de respuesta posible. Cada día estas exigencias van creciendo, aumentando la cantidad de datos a procesar y en el menor tiempo posible. Una de las herramientas de las que disponen los informáticos para aumentar la rapidez de respuesta en sus programas es el profiling o perfilaje que nos entrega información detallada del tiempo que está tomando ejecutarse un programa. En este laboratorio se utilizará esta herramienta para reducir el tiempo de ejecución de una función que calcula la tangente hiperbólica mediante el análisis del pipeline.El presente laboratorio requiere la implementación de la función tanh utilizando la serie de Taylor,

se pide hacerlo en lenguaje C y con 3 decimales de precisión. Una vez implementada, se dibuja el pipeline identificando posibles hazards y posterior a esto se itera modificando el código hasta que ya no se encuentren hazards. En cada iteración se debe utilizar gprof, un profiler para sistemas operativos GNU y con este se debe calcular los tiempos que van tomando los distintos códigos para luego sacar conclusiones respecto a esto.El código será realizado en Ubuntu con el compilador GCC. El valor al cual se le va a calcular la

tangente hiperbólica se debe introducir como argumento del ejecutable con la bandera n.^{es}pacio y el valor. Para manipular este argumento dentro del código se hace uso de la función getopt que facilita el análisis de los argumentos e identifica los errores al momento de ingresarlos. Además el informe es desarrollado en Latex y se utiliza un repositorio en Github para registrar los avances realizados.Al realizar este laboratorio

se comprueba empíricamente, como afecta en el tiempo de ejecución el orden de las instrucciones en un código y las dependencias que se generan entre estas. Así se podrá tener un criterio para futuros proyectos a la hora programar, con el fin de reducir lo más posible el tiempo de ejecución.En el capítulo 2 se habla

sobre el pipeline, sus etapas en el procesador MIPS y sobre los hazard. En el capítulo 3 se explica como fue llevado a cabo el desarrollo del trabajo. En el capítulo 4 se discuten los resultados obtenidos y se explican de acuerdo a la materia vista en clases y a investigación personal.

CAPÍTULO 2. MARCO TEÓRICO

El pipeline es un técnica de implementación en la cual se traslapan las instrucciones durante la ejecución.

2.1 ETAPAS DEL PIPELINE

En MIPS se tienen 5 etapas:IF: En esta etapa se busca la instrucción en memoria.ID: Se leen los registros

mientras se decodifica la instrucción.EX: Se ejecuta la operación de la instrucción o bien se calcula una dirección de memoria.MEM: En esta etapa se accede a un operando que se encuentra en memoria.WB:

Finalmente se escriben los resultados en un registro.

2.2 HAZARDS

Los hazards son obstáculos que se producen durante la ejecución del pipeline cuando una instrucción no se puede ejecutar en el siguiente ciclo de reloj. Existen 3 tipos de hazards:Hazard Estructural: Estos se

producen cuando el hardware no soporta una combinación de instrucciones.Hazard de Datos: Se producen cuando los datos que requiere una instrucción aún no están disponibles.Hazard de Control: Se producen cuando la instrucción buscada no es la que se tiene que ejecutar.

CAPÍTULO 3. DESARROLLO

CAPÍTULO 4. DISCUSIONES DE LOS RESULTADOS

CAPÍTULO 5. CONCLUSIÓN

CAPÍTULO 6. BIBLIOGRAFÍA

Garay, F. (2015). Laboratorio 3 organización de computadores: pipeline.

Olmedo, H. (2015). Laboratorio-3. Recuperado desde https://github.com/hernanolmedo/Laboratorio-3

Wikipedia. (2013). Análisis de rendimiento de software — wikipedia, la enciclopedia libre. [Internet; descargado 11-octubre-2015]. Recuperado desde https://es.wikipedia.org/w/index.php?title=An%C3%83%C2%A1lisis_de_rendimiento_de_software&oldid=69967834