



Proyecto #3

Microarquitectura e ISA

Generalidades

- Fecha de entrega: Lunes, 22 de Noviembre, 2010 @ 12:00pm.
- Modalidad: Grupal (2 personas)

Descripción

En esta última etapa del diseño de una arquitectura de computadores de 6 bits es necesario completar los niveles superiores del nivel de lógica digital: el nivel de microprogramación y el nivel ISA.

Para completar la arquitectura se utilizará una unidad de control microprogramada, lo que reducirá el tiempo de diseño dado que no será necesario realizar la implementación en lógica digital de dicha unidad, solamente será necesario proveer el conjunto de microinstrucciones que deberán ejecutarse. Además, será necesario presentar el conjunto de instrucciones de la arquitectura (ISA) y finalmente el diseño de la microarquitectura que soporta este conjunto de instrucciones.

En resumen, el proyecto se divide en tres secciones:

1. Microinstrucciones
2. ISA
 - a. Tipos de datos nativos
 - b. Instrucciones
 - c. Registros
3. Microarquitectura

Requerimientos

Microinstrucciones

Durante todo el desarrollo de la arquitectura (ALU, subsistema de memoria y registros de trabajo) se ha trabajado en el nivel más bajo en una máquina multinivel, el nivel de lógica digital (ver Ilustración 1). Sin embargo, es necesario proveer una serie de microinstrucciones a la unidad de control microprogramada para así simplificar las tareas de programación. Por tanto se hace necesario crear el *nivel de microarquitectura*.



En esta etapa del proyecto se les solicita diseñar el conjunto de microinstrucciones que debe proveer la arquitectura a los niveles superiores. A manera de recordatorio, una microinstrucción es una palabra de N bits que representa las señales de control de un microprocesador.

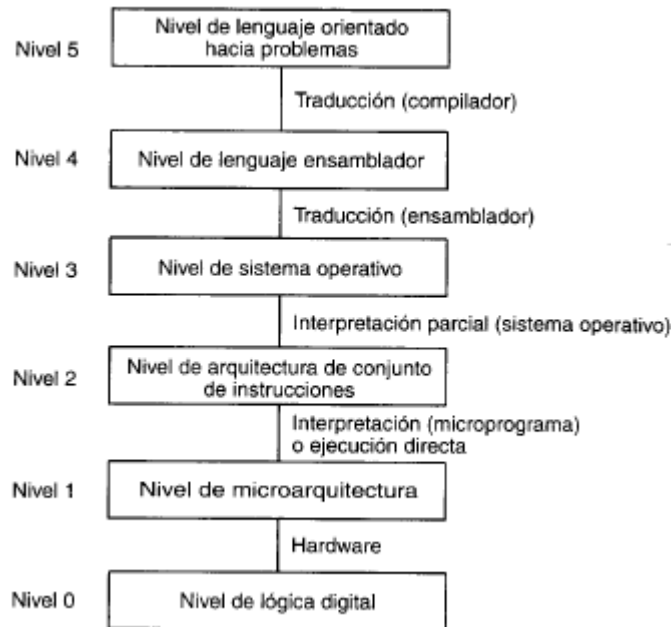


Ilustración 1. Computadora de seis niveles. Cada nivel se apoya sobre el que está inmediatamente abajo de él.

A manera de ejemplo se tiene la operación de alto nivel de sumar dos valores numéricos. Para poder realizar esta operación en la arquitectura diseñada es necesario realizar una serie de pasos. Cada uno de estos pasos consiste en habilitar los componentes requeridos y transferir los datos necesarios de un componente a otro:

1. Activar la señal de habilitación y de lectura del registro que contiene el valor del operando #1.
2. Activar la señal de habilitación y de lectura del registro que contiene el valor del operando #2.
3. Activar la señal de habilitación del sumador en el ALU. Además se debe activar la señal de control para realizar una suma (y no una resta).
4. Esperar unos ciclos de reloj para que se complete la operación (de acuerdo al tiempo de propagación de los circuitos implementados).
5. Almacenar el resultado en el registro de salida.

Para esta etapa es necesario identificar todas las señales de control de la arquitectura y definir el formato y tamaño de la palabra de microinstrucción. Cada bit deberá tener su significado. El resultado de esta etapa (lo que se debe presentar) es:



- Formato de microinstrucción. Descripción del formato, así como el detalle de cada bit que forma parte de la microinstrucción y cuál labor desempeña.
- Referencia de las microinstrucciones, desglosado por componente. Es decir, por cada circuito presente en la arquitectura diseñada es necesario presentar qué microinstrucciones realiza. Cada microinstrucción debe tener el nombre de la operación, su descripción y el desglose de la palabra de microinstrucción, resaltando únicamente los bits que son relevantes.

Instruction Set Architecture

Una vez diseñado el nivel de microarquitectura es necesario abstraer las microinstrucciones en lo que se conoce como ISA (Instruction Set Architecture). Este nivel consiste en una especificación que detalla las instrucciones que el CPU puede entender y ejecutar. Esta especificación debe incluir:

1. Tipos de datos nativos: Los tipos de datos que la arquitectura entiende nativamente. Es decir, sus componentes funcionales a nivel de lógica digital y/o mediante el uso de microinstrucciones. Por ejemplo: una microarquitectura puede tener a nivel de lógica digital solamente un sumador de números enteros positivos, pero mediante el uso de un circuito de complemento a dos se puede exponer una función de resta mediante la integración de más de una microinstrucción (complementar y luego sumar).
2. Conjunto de instrucciones: El lenguaje ensamblador que será utilizado por los programadores para implementar programas en la arquitectura diseñada. Se deberá seleccionar entre un conjunto RISC o CISC. Y deberán dividirse las instrucciones en categorías (transferencia de datos, operaciones aritméticas, etc). Se debe proveer una referencia donde se detalle cada instrucción, qué operandos recibe, qué formato deben llevar, y cómo funciona y se almacena el resultado de la operación (y dónde). Finalmente, para cada instrucción ensamblador se deben especificar cuáles son las microinstrucciones que debe ejecutar la unidad de control para ejecutarla.
3. Registros: Se deben especificar cuántos y qué tipo de registros contiene la arquitectura y especialmente qué función desempeñan. El número de registros deberá ir con la mano del *instruction set* seleccionado (RISC o CISC). Se debe proveer el nombre de cada registro y las funciones que desempeñan deberán ser parte del conjunto de instrucciones.

Los puntos anteriores se resumen en lo que se conoce como el *Manual de Referencia del Lenguaje de Máquina*. Y en su totalidad deberán cumplir con los objetivos de un ISA: ser completo, ser eficiente, las instrucciones deberán ser autocontenidas y además independientes.

Microarquitectura

En esta parte lo que se deberá hacer es diseñar el diagrama de bloques de la microarquitectura. Este diagrama presentará todos los diagramas funcionales y cómo están interconectados entre



ellos. Se deberá especificar también el número de *buses*, su ancho de banda, y qué función desempeñan.

Además, se deberán utilizar al menos dos de las siguientes técnicas de diseño de microarquitecturas. Las técnicas deberán verse reflejada tanto en el diagrama de bloques como en el ISA. Y además se deberá proveer el diseño específico de la técnica.

- Instruction Pipeline: Tipo de pipeline, cantidad de etapas y manejo de condiciones de riesgo.
- Cache: Niveles de cache, tamaño y tipos.
- Branch Predictor: Estrategia de predicción.
- Superscalar Architecture: Unidades funcionales duplicadas y cómo se manejará el paralelismo.
- Out-of-order Execution: Unidades funcionales adicionales y estrategia de ordenamiento y ejecución.

Entregable

- Documentación en formato digital dividido y estructurado en tres secciones:
 - Microinstrucciones
 - Manual de referencia del lenguaje de máquina
 - Microarquitectura

Criterios de Evaluación

- Documentación – 100 pts.