

Capítulo 2

Titulo por definir

*«La verdadera ciencia enseña, sobre
todo, a dudar y a ser ignorante.»*

Ernest Rutherford

RESUMEN: En este capítulo se define con detalle lo que es un procesador y su importancia en el mundo hoy en día. También se habla sobre una arquitectura más concreta, la arquitectura ARM.

A continuación se define qué es un fallo y qué tipos de fallos pueden ocurrir en los sistemas. Además se explican algunas técnicas de tolerancia a fallos.

Para terminar se justifica la importancia de la tolerancia en los sistemas y concretamente porque es necesaria la tolerancia en los microprocesadores.

Introducción

...

2.1. Procesador

El Diccionario de la Real Academia Española (DRAE) define al procesador como la «unidad central de proceso (CPU), formada por uno o dos chips» 2.1.

La CPU es el circuito integrado encargado de acceder a las instrucciones de los programas informáticos y ejecutarlas. Para poder ejecutar un programa, el procesador debe realizar las siguientes tareas:

1. Acceder a las instrucciones almacenadas en memoria.

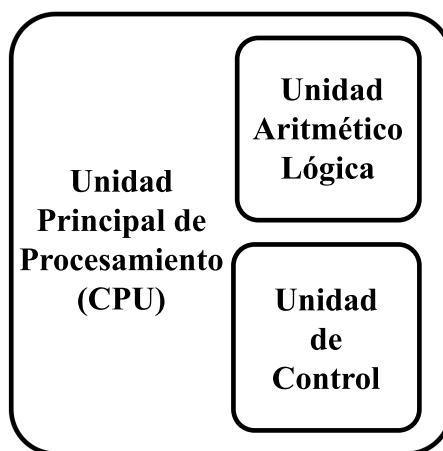


Figura 2.1: Procesador DRAE

2. Analizar las instrucciones y establecer las señales de control internas.
3. Ejecutar operaciones sobre datos.
4. Almacenar los resultados en memoria.

A continuación se definen los elementos fundamentales para definir un procesador.

2.1.1. Arquitectura

Un procesador está formado por una serie de módulos conectados entre sí. La arquitectura del procesador define el diseño de estos módulos y de qué manera se conectan entre ellos.

La arquitectura del procesador diseñada por Von Neumann separó los componentes del procesador en módulos básicos. La CPU es el núcleo de los computadores, se divide en computación y control, y contiene todo excepto la memoria y los elementos de entrada y salida [Hennessy y Patterson (1993)].

2.1.2. Repertorio de instrucciones

El repertorio de instrucciones define todas las operaciones que el procesador es capaz de entender y ejecutar. Este juego de instrucciones incluye las operaciones aritmético-lógicas que puede aplicar a los datos, las operaciones de control sobre el flujo del programa, las instrucciones de lectura y escritura en memoria así como todas las instrucciones propias que se hayan diseñado para el procesador.

2.1.3. Memoria

Los procesadores tienen una serie de registros que almacenan temporalmente los valores con los que está trabajando. El conjunto de estos registros se conoce como banco de registros. Estos registros de propósito general son muy limitados. Por ello el procesador necesita de apoyo externo para alojar la información, para esto tiene acceso a una memoria externa.

El acceso a la memoria externa divide las arquitecturas en dos tipos. La arquitectura Von Neumann utiliza una única memoria para almacenar tanto los datos como las instrucciones. Las arquitecturas Harvard, sin embargo, separan la memoria de datos de la memoria de instrucciones 2.2.

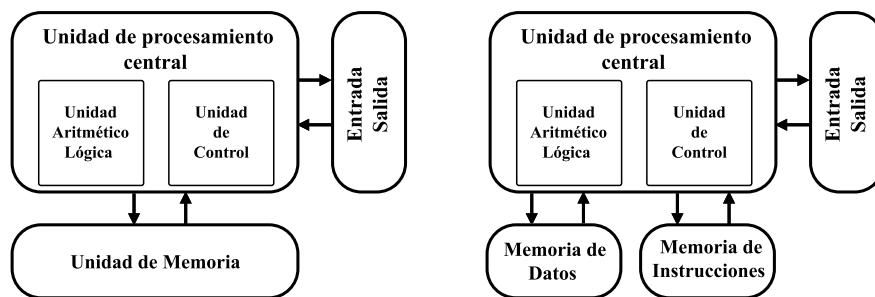


Figura 2.2: Arquitectura Von Neumann y Arquitectura Harvard

2.1.4. Segmentación

La segmentación es una técnica de implementación, que no es imprescindible pero aumenta el rendimiento del procesador. Esta permite que haya varias instrucciones en ejecución al mismo tiempo en el mismo procesador. El procesador es dividido en etapas y en cada etapa se realiza una parte del trabajo completo de la instrucción de forma secuencial.

La segmentación permite que en cada ciclo de reloj se busque una instrucción y se comience su ejecución. Esto permite reducir el número de ciclos total que necesita el programa.

En la tabla 2.1 podemos ver cómo se lanza una serie de instrucciones. Se observa cómo las instrucciones ocupan únicamente una etapa del procesador, y cómo avanzan por el procesador dejando libre la etapa anterior para la siguiente instrucción.

Para conocer los ciclos por instrucción que necesita un programa se utiliza la fórmula 2.1. Si aplicamos esta fórmula a un programa en dos procesadores iguales, con la única diferencia de que uno de ellos está segmentado ayuda la segmentación al rendimiento del procesador.

| | Ciclo de reloj | | | | | | | | |
|-----------------------|----------------|----|----|-----|-----|-----|-----|-----|----|
| Número de instrucción | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| i | IF | ID | EX | MEM | WB | | | | |
| i + 1 | | IF | ID | EX | MEM | WB | | | |
| i + 2 | | | IF | ID | EX | MEM | WB | | |
| i + 3 | | | | IF | ID | EX | MEM | WB | |
| i + 4 | | | | | IF | ID | EX | MEM | WB |

Tabla 2.1: Segmentacion simple de 5 etapas

$$\text{Ciclos por instrucción (CPI)} = \frac{\text{Número de ciclos total}}{\text{Número de instrucciones}} \quad (2.1)$$

Vamos a suponer un procesador no segmentado que tarda 5 ciclos de reloj en ejecutar cualquier instrucción. Suponemos un segundo procesador que también tarda 5 ciclos de reloj, pero este está segmentado en 5 etapas de 1 ciclo cada una. Ahora vamos a ejecutar un programa de 10 instrucciones en ambos procesadores.

En el procesador no segmentado obtenemos que el programa tarda 50 ciclos en ejecutarse y el CPI es de 5. Al ejecutar el mismo programa en el procesador segmentado, este tarda 5 ciclos en llenar las 5 etapas del procesador. A partir de ahora cada ciclo de reloj se termina una instrucción, terminando la ejecución en 15 ciclos de reloj. El nuevo CPI es de 1,5. La segmentación ha reducido el tiempo de ejecución de este programa a menos de una tercera parte.

2.1.5. DLX

...

2.1.6. ARM

...

2.2. Fallos

Existen una gran variedad de fallos que pueden ocurrir en un sistema electrónico. Los fallos se pueden clasificar en fallos software y fallos hardware. Y los podemos encontrar desde fallos en la definición de requisitos que se propagan hasta la fase de producción, hasta fallos producidos en el sistema por agentes externos como la radiación.

En esta sección se hablará de esta última clase de fallos, los fallos producidos por agentes externos que no se pueden evitar en las fases de diseño. Y que afectan al hardware, dañando sus componentes o alterando los valores de las señales con las que se trabaja.

Los fallos analizados en esta sección se van a dividir en dos categorías; fallos permanentes y fallos transitorios.

2.2.1. Fallos Permanentes

Los fallos permanentes son aquellos que afectan al sistema de forma irreversible. Producen cambios en el diseño que estropean el correcto funcionamiento del módulo o circuito que lo sufre. Estos fallos no se solucionan reiniciando el sistema. [Jedec (2006)]

critical charge:

hard error:

single-event functional interrupt (SEFI):

single-event latch-up (SEL):

single event transient (SET):

single-event upset (SEU):

single-event upset (SEU) rate:

soft error, device:

soft error rate (SER):

2.2.2. Fallos Transitorios

...

2.3. Tolerancia a Fallos

La tolerancia a fallos se define como la capacidad de un sistema de funcionar correctamente incluso si se produce un fallo o anomalía en el sistema

Existen dos tipos de tolerancia; tolerancia estática y tolerancia dinámica.

2.3.1. Tolerancia estática

...

2.3.2. Tolerancia en dinámica

...

2.3.3. Tolerancia en microprocesadores

...

Bibliografía

*Y así, del mucho leer y del poco dormir,
se le secó el cerebro de manera que vino
a perder el juicio.*

Miguel de Cervantes Saavedra

HABINC, S. Functional Triple Modular Redundancy (FTMR). *Design and Assessment Report, Gaisler Research*, páginas 1–56, 2002.

HENNESSY, J. L. y PATTERSON, D. A. *Arquitectura de Computadores: Un enfoque cuantitativo*. Mcgraw Hill Editorial, 1993. ISBN 1558600698.

JEDEC. Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Error in Semiconductor Devices: JESD89A. *JEDEC Solid State Technology Association*, páginas 1–85, 2006.

SADASIVAN, S. An introduction to the arm cortex-m3 processor. 2006.