

DISEÑO E IMPLEMENTACIÓN DEL SUBSISTEMA DE MEMORIA DE UN COMPUTADOR PARA LA REALIZACIÓN DE PRÁCTICAS DE LABORATORIO.

DANIEL CASCADO CABALLERO, L. MIRÓ AMARANTE, M.A. RODRÍGUEZ JODAR, G. JIMÉNEZ MORENO, C. AMAYA RODRÍGUEZ

*Area de Arquitectura y Tecnología de Computadores. Facultad de Informática.
Universidad de Sevilla. Avda. Reina Mercedes, s/n. 41012-Sevilla. España.*

Este trabajo describe una práctica de laboratorio con la que se pretende ilustrar el diseño del subsistema de memoria de un computador. Se utiliza memoria dinámica lo que implica una mayor complejidad. El sistema con el que se realiza esta práctica se basa en dar parcialmente resuelto el circuito, dejando en manos de los alumnos la parte de la unidad de control DRAM que permite una implementación más variada.

1. Objetivos, método y originalidad:

El estudio de un computador se puede llevar a cabo de diversas formas. Un procedimiento clásico es dividirlo en niveles [1], de forma que el nivel superior estaría ocupado por el software de usuario y el nivel inferior por los sistemas electrónicos que lo componen. El sistema operativo, el código máquina, etc, ocuparían niveles intermedios. Si se pretende estudiar el computador en el nivel electrónico, se puede dividir, desde el punto de vista académico, en tres subsistemas estrechamente relacionados: procesador, memoria y entrada/salida, todos conectados entre sí mediante la correspondiente "lógica de pegado" ("glue logic") [2]. Entender completamente cómo funciona un computador es conocer todos los niveles y subsistemas que lo constituyen, y cómo se relacionan entre sí. Además, no sólo es interesante que el alumno conozca su funcionamiento, es importante que pueda diseñar sistemas reales con los componentes y herramientas que habitualmente ofrece el mercado, en cualquiera de los niveles (o subsistemas) que anteriormente se han mencionado. En este contexto las prácticas de laboratorio orientadas al diseño son fundamentales.

Por otra parte las prácticas de laboratorio podríamos dividir las en dos tipos:

- Las que ilustran unos pocos conceptos fundamentales, realizan una abstracción de la realidad para simplificar su elaboración, suelen ofrecer escasa dificultad y su duración es corta.
- Y las que podríamos denominar de síntesis de conocimientos, precisan de una gran información previamente adquirida, se manejan múltiples conceptos, su duración es más larga y suelen ser más próximas a los sistemas reales.

El primer tipo de prácticas suele ser las habituales en los primeros cursos o en materias en las que se imparten fundamentos, el segundo tipo de prácticas son normales en los cursos superiores o en asignaturas de especialización. Ambos tipos son necesarios para una formación integral del alumno.

En este trabajo se describe una práctica de laboratorio de síntesis de conocimientos, en la que el alumno diseña y monta parte del subsistema de memoria. La práctica consiste en ampliar la memoria (semiconductora) de un computador diseñando para ello todo el sistema de control.

computador al que se le pretende aumentar la memoria. En nuestro caso se trata de un entrenador basado en i8088 cuyo bus de expansión es similar al del PC XT (Slot ISA de 8 bits) [4]. En esta placa de ampliación de memoria se encuentra cableado el bus de datos, la alimentación y parcialmente el bus de direcciones, los zócalos de los chips y de conexión al computador son del tipo wire wrapping.

Un esquema del sistema que se entrega a los alumnos se muestra en la figura 1, en él se puede observar, que de las PLDs sólo se conecta la señal de reloj, además hay una serie de pines en los conectores que están dispuestos para ser cableados (señalados en el esquema de forma especial (/HOLD, RESDRV ...)). En el esquema se puede observar que casi todas las líneas de direcciones están ya conectadas al módulo SIMM a través de los multiplexores, excepto las líneas A9 y A17 que deben multiplexar y conectar los alumnos mediante alguna de las PLDs. El contador 74LS169 se utiliza como "prescalador" para contar el tiempo y determinar cuando se realiza el refresco, este circuito se debe completar con una pequeña máquina de estado (a implementar de nuevo en alguna de las PLDs).

Evidentemente el alumno debe haber estudiado previamente cómo se desarrollan sistemas con PLDs, la decodificación de memoria, cómo son los accesos de las DRAM, los ciclos en el bus de expansión del computador, los tipos de refresco y, por último, el patillaje y funcionamiento de todos los chips de que disponen en esta práctica. En la documentación de la práctica se proporcionan las hojas de características de los diversos componentes y el mapa de memoria y los cronogramas del computador al que se le va a realizar la ampliación. La misión del alumno es generar los ciclos de lectura, escritura y refresco de la DRAM mediante los elementos que se proporcionan, comprobando si es necesario generar estados de espera. Para ello es necesario que en las clases teóricas se le haya explicado al alumno previamente como se diseñan las máquinas de estados que controlan los accesos a las memorias semiconductoras. El alumno debe utilizar las PLDs para diseñar dichas máquinas de estados y realizar la decodificación necesaria [5]. El contador junto con parte de la PLD, se utiliza para determinar cuándo toca refrescar la memoria y avisar a la máquina de estado que debe realizar un ciclo de refresco. En nuestro caso cuando se realiza el refresco se detiene y aísla el procesador (mediante señales del tipo HOLD y HLDA). El sistema está pensado para refrescos del tipo CAS Previo RAS. Sin embargo, se propone al alumno que elija y compare qué diferencia habría entre realizar el refresco distribuido o concentrado (ráfaga). También se pide que se estudie que tanto por ciento de tiempo se pierde debido a la parada por refresco.

2. Resultados:

Puede observarse que el cableado que debe realizar el alumno mediante las correspondientes herramientas de wire wrapping es relativamente escaso, sólo las señales de control y dos líneas de direcciones para la decodificación (se propone colocar los 256K de DRAM a partir de la posición 0x40000). El proceso a seguir por el alumno es el siguiente:

- Antes de entrar al laboratorio, el alumno debe estudiar toda la documentación que se le entrega, analizar los cronogramas, elegir el periodo y tipo de refresco, diseñar las máquinas de estado necesarias, y plantear las ecuaciones y definición de pines de las PLDs. Simplificando bastante el diseño basta con una sola 22V10 para llevarlo a cabo.
- Ya en el laboratorio, el alumno debe utilizar un entorno de desarrollo para PLDs (PALASM o CUPL) para generar los ficheros de programación (.JED), según el estudio previo.
- Después utilizando el programador universal se graban las PLDs (borrables eléctricamente).
- El cableado de las líneas de control se realiza mediante wire wrapping.
- Por último debe realizar un pequeño programa en el entrenador que ejecute un test de la ampliación de memoria para comprobar su correcto funcionamiento.

- Posteriormente debe entregar la correspondiente memoria con el diseño y evaluación. Se han construido siete placas de ampliación de memoria para siete grupos compuestos cada uno por dos alumnos (en la fotografía de la figura 2 se muestra el resultado). La construcción de este sistema se ha realizado sobre placas de prototipo de simple cara completando el cableado manualmente, pero está previsto para el curso que viene hacerla sobre una placa PCB específica.

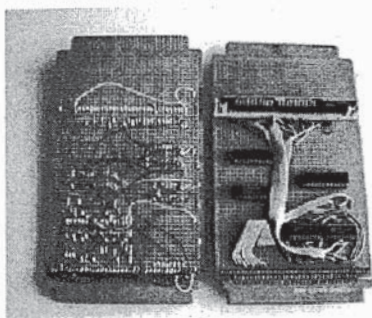


Figura 2: Fotografía de dos placas de prototipo montadas (parte anterior y posterior).

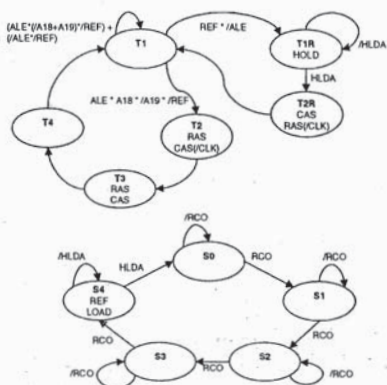


Figura 3: Posible solución para la unidad de control de la DRAM..

La primera máquina de estados de la figura 3 muestra una posible solución para la unidad de control de la DRAM, la segunda máquina de estado cuenta desbordamientos de los contadores (RCO de los 74LS169) para activar el refresco, este se produce cuando REF (señal de salida de la segunda máquina de estado) se activa y la señal ALE se encuentra desactivada. Esta solución es la adoptada por la mayoría de los alumnos (tiene un pequeño problema que estos descubren cuando ejecutan los programas de test).

Es evidente que esta práctica es relativamente compleja, sólo realizable en asignaturas de especialización, en los últimos años de carrera y con un número relativamente bajo de alumnos (en nuestro centro se imparte en una asignatura optativa). Por otra parte la cualidad principal que se consigue con la misma es la motivación del alumno; éste comprueba cómo funciona un sistema real que aparece en todos los computadores, dejando de ser para él o ella una caja negra, al mismo tiempo que le sirve para asentar muchos de los conocimientos adquiridos en diferentes asignaturas.

Referencias:

- [1] A.S. Tanenbaum, "Structured computer organization" (4ª ed.). Editorial: Prentice-Hall.
- [2] R. H. Katz. "Contemporary Logic Design". Editorial: Addison-Wesley.
- [3] J.D. Nicoud, "Microprocessor interface Design". Editorial: Chapman & Hall.
- [4] Liu, C.A. Gibson, "Arquitectura, programación y diseño de sistemas basados en microprocesadores". Editorial: Anaya
- [5] M. Bolton, "Digital Systems Design with Programmable Logic". Editorial: Addison Wesley.