

MCRE. Ejercicio 5: Elementos de memoria



E.T.S.I. Telecomunicación

Fecha tentativa de entrega: 29 de Mayo de 2009

El objetivo de este ejercicio es implementar los subsistemas de memoria del microcontrolador del proyecto final. Por lo tanto se debe realizar la memoria de programa (ROM) y de datos (banco de registros, GPR).

1. Memoria de programa (ROM)

La memoria de programa disponible en el chip real es 1 Kword ($1K \times 14bits$), aunque como se ha simplificado el diseño para que el PC sea de sólo un byte, en nuestro diseño basta con implementar una ROM de 256 posiciones (256×14 bits).

En un diseño real esta memoria de programa sería una EEPROM, pero en nuestro caso vamos a realizarla con una memoria ROM básica. Las memorias ROM se pueden implementar de forma sencilla con sólo un transistor por célula de memoria, de tal forma que si el transistor existe se lee un "1" y si no existe se lee un "0". El programa *MicroWind* proporciona un modelo de cómo se realizaría la matriz de células de una ROM de 8x8 bits (rom.MSK). A esta matriz será necesario añadirle los circuitos de selección y lectura necesarios para su correcto funcionamiento (los mismos que se utilizan para realizar una RAM).

El correcto funcionamiento del microcontrolador final se debería comprobar mediante la programación de la ROM con un pequeño programa de prueba y la simulación del sistema completo. Debido a las limitaciones de la herramienta *MicroWind*, y a que queda fuera de los objetivos del curso, no es necesario realizar esta simulación completa. Sin embargo, resulta interesante que a nivel teórico se plantee cómo funcionaría el mini-pic en el caso de un programa real. Este programa debería incluir unas pocas instrucciones que sirvan para verificar el correcto funcionamiento de los elementos básicos del microcontrolador: ALU, banco de registros y pila. A modo de ejemplo se adjunta el siguiente programa, que se puede utilizar como base para la justificación del correcto funcionamiento del diseño realizado.

```
; FILE: adding.asm
; DESC: Adds two numbers contained in registers

;-----
; cpu equates (memory map)
portB equ 0x06
trisB equ 0x86
reg1 equ 0x0c
reg2 equ 0x0d
;-----
```

00h	INDF	INDF	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2	89h
0Ah	PCLATCH	PCLATCH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	36 Registros de Propósito General (GPR)		8Ch
2Fh			

Figura 1: Mapa de memoria de datos

```

org      0x000

start    movlw  0x00          ; load W with 0x00 make port B output
         movwf  trisB        ; copy W tristate, port B config reg
         clrf  portB        ; clear all lines low
         call  adding
         movf  reg2          ; copy reg2 to W
         movwf portB        ; move W to port B
         goto  start

adding   movf  reg1, 0       ; set w = reg1 decimal
         addwf reg2, f       ; reg2 = reg2 + reg1
         return              ; else exit subroutine

end

```

En este programa se han utilizado etiquetas para facilitar su lectura. Se supone que el compilador las sustituye por sus valores reales. Con un programa de complejidad similar es suficiente para probar el funcionamiento del microcontrolador, pues contiene instrucciones de la ALU, de control y de movimiento entre registros.

2. Banco de registros

Como se dijo en el enunciado del proyecto final, la memoria de datos se descompone a su vez en RAM de propósito general, *Registros de propósito general (General Purpose Register, GPR)* y en *Registros con función especial (Special Function register, SFR)*, todos de 8 bits. Esta memoria (GPR y SFR) está distribuida en dos bancos, por lo que es necesario controlar a qué banco se accede mediante bits de control. Estos bits están en el registro de estado (status<7.5>). En la figura 1 se puede ver qué posiciones ocupan los registros especiales en el fichero de registros (Register File, RF en lo sucesivo), así como la estructura que sigue esta memoria.

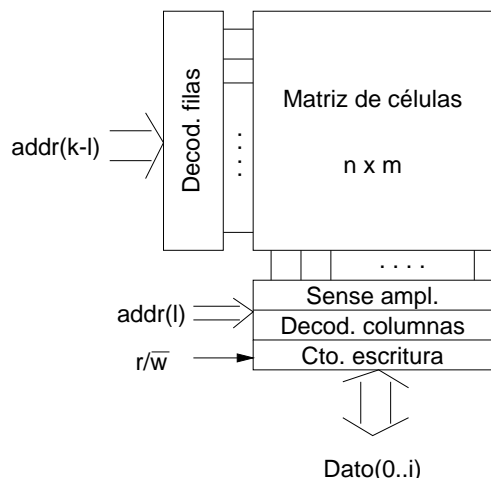


Figura 2: Estructura de una memoria

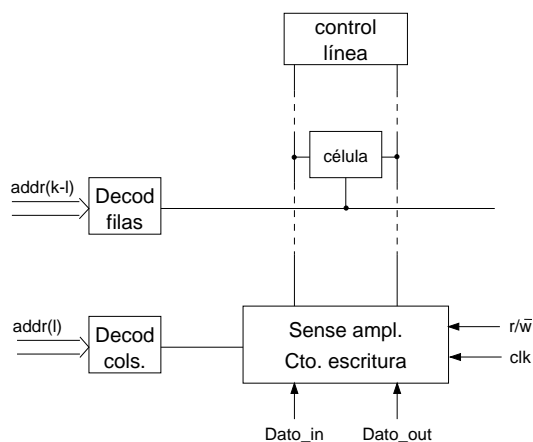


Figura 3: R/W de una célula de memoria

2.1. Implementación del banco de registros

Un banco de registros (*Register File*) no es más que una memoria RAM rápida con múltiples puertos de lectura y escritura. Una célula de memoria CMOS convencional se puede hacer multi-puerto añadiéndole transistores de paso. El funcionamiento global del microcontrolador será quién imponga la necesidad de uno o más puertos en el banco de registros.

Una memoria CMOS¹ integra los siguientes componentes básicos:

La célula de memoria. Puede ser estática o dinámica dependiendo de la forma de almacenamiento del dato. Las células estáticas son más grandes que las dinámicas, pero tienen la ventaja de no necesitar refresco. La forma más eficiente de distribuir las células es colocarlas en una matriz cuadrada, como se muestra en la figura 2.

El decodificador de filas, que permite seleccionar la fila de la matriz de células de memoria a la que se quiere acceder (tanto para lectura como para escritura). Para ello se utilizarán parte de las líneas de dirección.

El decodificador de columnas, que permite seleccionar las células de la palabra que se va a leer o escribir entre los datos de la fila proporcionada por el decodificador de filas.

Circuito de escritura de las células adecuadas (las de la palabra deseada).

Circuito de lectura de la fila adecuada (*sense amplifier*).

Circuito de control de línea de bit, que acondiciona esta línea para que sean factibles las operaciones de lectura y escritura.

La forma de distribuir las células en la matriz se describe a continuación. Hay que intentar hacer una matriz lo más cuadrada posible². Siguiendo la terminología de la figura 2, supongamos que tenemos una memoria con 2^k palabras (por lo tanto con k líneas de dirección) y cada palabra

¹El capítulo 8 de [WE94] o el capítulo 11 de [WH05] contienen una excelente descripción de la implementación de memorias CMOS. Igualmente el capítulo 7 del manual de usuario de MicroWind [Sic04] incluye la información necesaria para saber cómo realizar y simular memorias.

²Salvo que el plano de base nos imponga otra distribución.

tiene b bits. En total tendremos $2^k \times b$ células. Todas estas células se ordenan en una matriz de dimensiones $n \times m$, donde $m = 2^{k-l}$ y $n = 2^l \times b$, siendo l una parte de las líneas de dirección necesaria para hacer la selección de las columnas. Hay que intentar que n y m sean lo más parecido posible para tener forma cuadrada.

En la figura 3 aparece representada la relación existente entre todos los elementos constituyentes de la memoria y su disposición en las operaciones de lectura y escritura.

3. Preguntas

1.
 - a) Implemente y simule los elementos básicos de la memoria de programa del Mini-PIC. Recuerde que es suficiente con realizar una ROM, por lo que sus elementos principales son, además de la célula de memoria, los selectores de fila y columna. Utilice para ello si le sirven los elementos pre-definidos dentro de la herramienta MicroWind.
 - b) Caracterice los elementos de la memoria ROM que ha implementado, proporcionando el área, el retardo para el caso peor y el consumo de potencia. (Nota: para la medida del retardo en el caso peor considere la célula peor situada en el trazado.)
2.
 - a) Implemente y simule los elementos básicos de la memoria de datos del Mini-PIC (banco de registros). De nuevo, utilice para ello si le sirven los elementos pre-definidos dentro de la herramienta MicroWind. En el manual de usuario de la herramienta [Sic04] se explica cómo realizar una RAM. Como célula básica se recomienda utilizar la célula `RamStatic6T` o `RamStatic6TCompact`. En este caso, al ser una memoria re-programable serán necesarios los correspondientes circuitos de lectura y escritura.
 - b) Caracterice los elementos de la memoria RAM que ha implementado, proporcionando el área, el retardo para el caso peor y el consumo de potencia. (Nota: al igual que sucedía con la ROM, para la medida del retardo en el caso peor considere la célula peor situada en el trazado.)
3.
 - a) Realice los trazados completos de la ROM y la RAM. Para ello:
 - Trate de conseguir un plano equilibrado, ni muy alargado, ni muy estirado. Aunque puede que para el trazado final del Mini-PIC pueda interesarle una versión con proporciones diferentes.
 - Trate de conseguir un trazado lo más compacto posible, minimizando al máximo los espacios vacíos.
 - Planifique previamente cómo se van a colocar los distintos módulos dentro del plano global, teniendo en cuenta sus tamaños y formas.
 - b) Explique las decisiones de diseño que ha tomado: tipo de ROM, tipo de RAM, distribución en el espacio, número de puertos, etc.
 - c) Caracterice las memorias. Establezca el área total, el área ocupada (porcentual aproximada), el área vacía (porcentual aproximada), el camino crítico, el consumo de potencia y $A \times t_p^2$, explicando cómo ha realizado los cálculos. Considere que las memorias están cargadas en sus salidas con 100fF.

Referencias

- [Sic04] Etienne Sicard. *Microwind&DSCH 3.0 User's Manual*. INSA Toulouse, 2004.
- [WE94] Neil H. E. Weste and Kamran Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley, 2nd edition, 1994.
- [WH05] Neil H. E. Weste and David Harris. *CMOS VLSI Design*. Pearson Addison-Wesley, 2005.