

Arquitectura de computadores

Práctica 5: Entrada/salida

5.1. OBJETIVO

En la práctica 4 se ha utilizado el dispositivo PI/T MC68230 como generador de interrupciones de temporización. Este circuito también permite realizar comunicaciones en paralelo con sistemas externos: con el usuario, con otros ordenadores, con máquinas o instrumentos, etc. En esta práctica estudiaremos la programación de entradas y salidas en paralelo sirviéndonos de dicho dispositivo.

5.2. PRÁCTICA

Los puertos digitales contienen líneas de comunicación con el exterior que pueden estar configuradas, todas ellas o individualmente, como entradas y/o salidas. De las líneas de entrada se puede *conocer* el estado (alto/bajo, 5V/0V), y en las líneas de salida se puede *establecer* un determinado estado (alto/bajo, 5V/0V).

5.2.1. Descripción básica del circuito integrado PI/T MC68230.

El circuito integrado PI/T (*Parallel Interface/Timer*) MC68230 es un periférico de la familia del 68000, por lo que el acoplamiento hardware y software con dicho microprocesador resulta directo. Sin embargo, debido a su gran potencia y versatilidad, especialmente por sus numerosas posibilidades de programación, una descripción completa y exacta del mismo sería demasiado larga y tediosa; baste decir que la información proporcionada por Motorola para este chip tiene 60 páginas de gran densidad y con información en absoluto redundante. Como no es objeto de estas prácticas un estudio pormenorizado del 68230, se va a dar una descripción limitada, y en ocasiones simplificada, de dicho circuito.

El 68230 incorpora un temporizador de 24 y tres puertos de 8 bits para entrada/salida digital en paralelo. El temporizador se ha estudiado ya en la práctica anterior, en la que era utilizado como dispositivo generador de interrupciones. En esta práctica nos centraremos en el estudio de los puertos de entrada/salida. Estos puertos serán designados en adelante como A, B y C.

El funcionamiento de los puertos A y B puede ser programado eligiendo una de siete modalidades diferentes, representadas en la siguiente figura:

	8 bits	16 bits
Uni-direccional	<div style="text-align: right; margin-bottom: 5px;">Modo 00</div> <div style="text-align: left; margin-bottom: 5px;">submodos</div> 00 - A y B entrada 01 - A y B salida 1X - A y B bit a bit	<div style="text-align: right; margin-bottom: 5px;">Modo 01</div> <div style="text-align: left; margin-bottom: 5px;">submodos</div> X0 - AB entrada X1 - AB salida
Bi-direccional	<div style="text-align: right; margin-bottom: 5px;">Modo 10</div> <div style="text-align: left; margin-bottom: 5px;">submodos</div> XX - A bit a bit B bidireccional	<div style="text-align: right; margin-bottom: 5px;">Modo 11</div> <div style="text-align: left; margin-bottom: 5px;">submodos</div> XX - AB bidireccional

Figura 5.1.- Modos de programación de los puertos A y B.

En los casos bidireccionales, el sentido de la transmisión (entrada o salida) es controlado en cada momento por unas líneas externas (H1, H2, H3, H4) que se denominan líneas de *handshaking* o protocolo). A estas líneas se accede en la tarjeta del entrenador 68fil a través de un conector de 4 pines situado entre los conectores de los puertos A y B. El puerto C consta de 2 líneas de, exclusivamente, entrada o salida, y 6 líneas con doble función. Las líneas de doble función pueden trabajar individualmente como entradas/salidas, o bien como líneas relacionadas con los puertos A o B, o con el temporizador. En la presente práctica no se utiliza el puerto C.

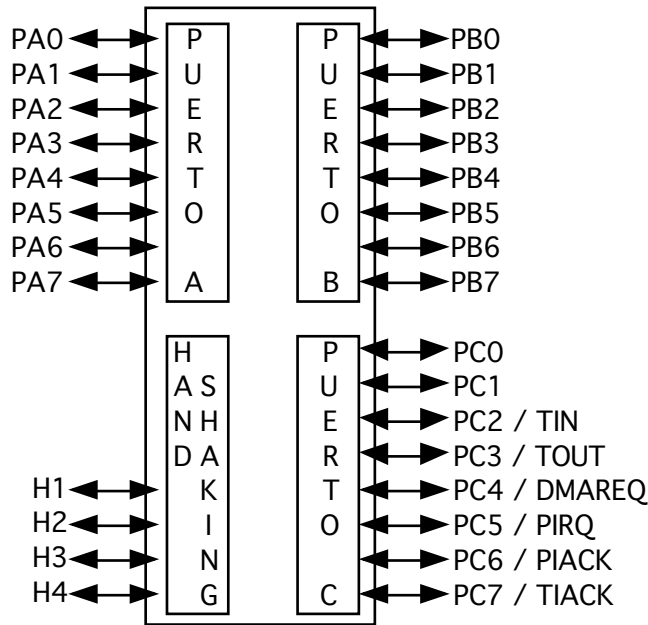


Figura 5.2.- Líneas de comunicación del 68230.

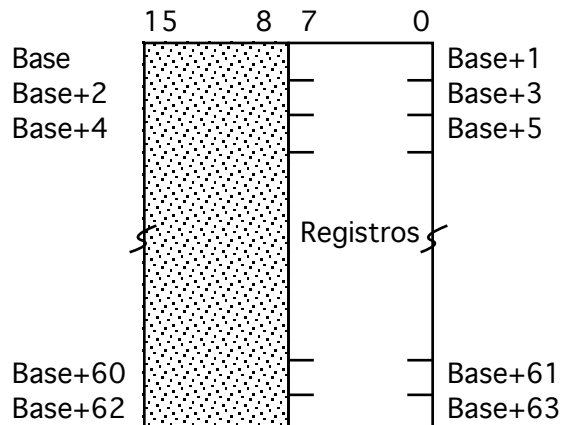


Figura 5.3.- Disposición de los registros del 68230 en el espacio de direcciones.

El modo de funcionamiento del 68230 se establece mediante sus *registros de control*. Desde el punto de vista del microprocesador, estos registros son similares a posiciones de memoria, en el sentido de que se accede a ellos mediante ciclos convencionales de lectura y escritura. No obstante, dichos registros *no son memoria*. Los registros del 68230 forman un vector compuesto por 32 palabras, a las que se accede mediante 5 líneas de selección de registro. Cada registro es de 8 bits, puesto que el 68230 dispone de 8 líneas de datos. Precisamente por este hecho, el 68230 sólo puede conectarse al 68000 bien usando la parte baja (líneas D0-D7), bien la parte alta (líneas D8-D15) del bus de datos, aunque lo normal es conectarlo a D0-D7, ya que durante el reconocimiento de una interrupción el 68000 lee siempre el número de vector a través de la parte baja del bus de datos. Esto implica que los registros del 68230 se encontrarán en direcciones impares del espacio de direcciones (figura 5.3). En el caso concreto del entrenador 68fil, la lógica de decodificación de direcciones está diseñada de modo que los registros del

68230 se hallan mapeados a partir de la dirección 800000_{16} (en realidad, a partir de la dirección 800001_{16} e impares sucesivas).

El mapa de los registros que hacen referencia a los puertos de entrada y salida en el 68230 es el siguiente:

00000	Registro general de control de los puertos	PGCR
00001	Registro de petición de servicio de los puertos	PSRR
00010	Registro de dirección de datos del puerto A	PADDR
00011	Registro de dirección de datos del puerto B	PBDDR
00100	Registro de dirección de datos del puerto C	PCDDR
00101	Número de vector de interrupción de puertos	PIVR
00110	Registro de control del puerto A	PACR
00111	Registro de control del puerto B	PBCR
01000	Registro de datos del puerto A	PADR
01001	Registro de datos del puerto B	PBDR
01010	Registro alternativo del puerto A	PAAR
01011	Registro alternativo del puerto B	PBAR
01100	Registro de datos del puerto C	PCDR
01101	Registro de estado de los puertos	PSR

Figura 5.4.- Registros del 68230 relacionados con los puertos de entrada y salida.

Para programar los puertos A y B en alguno de los 7 modos posibles, deben asignarse valores adecuados a los registros PGCR, PACR y PBCR. En la figura siguiente se muestra la estructura de estos registros:

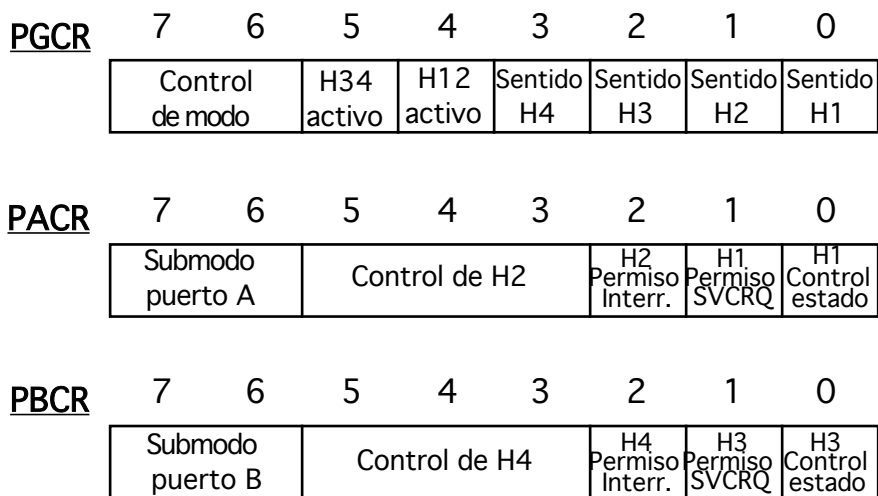


Figura 5.5. Registros de control de modo para los puertos A y B.

Cada modo de funcionamiento queda determinado por dos bits de modo: los bits 7 y 6 del PGCR, y por dos bits de submodo: los bits 7 y 6 del PACR o PBCR, según corresponda.

Por ejemplo:

Para utilizar los puertos A y B de forma que A sea un puerto de salida de 8 bits y B contenga 8 puertos independientes unidireccionales de un bit, se colocará el modo 00 en el PGCR ($PGCR = 00XXXXX_2$), el submodo 01 en el PACR ($PACR = 01XXXXXX_2$) y el submodo 1X en PBCR ($PBCR = 1XXXXXXX_2$).

Cuando un puerto funciona con bits independientes (modo 00, submodo 1X), debe indicarse el sentido (entrada/salida) de cada uno de los bits. Para ello el 68230 dispone de los “registros de dirección de datos” (PADDR, PBDDR, PCDDR). Cada uno de los bits de estos registros indica el sentido de los bits del puerto correspondiente (0 → entrada, 1 → salida).

Por ejemplo:

Supongamos que queremos programar los puertos A y B como en el ejemplo anterior, y concretamente que los bits 0 a 3 del puerto B deben ser salidas y los bits 4 a 7 entradas. Por tanto, además de lo considerado antes, se deberá escribir en el PBDDR el patrón binario 00001111₂.

Teniendo en cuenta que no vamos a usar las líneas H1-H4 -por lo que dejaremos a 0 todos los bits que hacen referencia a ellas-, las consideraciones anteriores pueden traducirse en las siguientes instrucciones:

```

PI_T    EQU    $800000
PGCR    EQU    PI_T+1
PBDDR   EQU    PI_T+7
PBACR   EQU    PI_T+13
PBBCR   EQU    PI_T+15

        MOVE .B    #%00000000,PGCR
        MOVE .B    #%01000000,PACR
        MOVE .B    #%10000000,PBCR
        MOVE .B    #%00001111,PBDDR
    
```

5.2.2. Uso de la placa IO'85.

Para poder realizar prácticas sencillas de entrada/salida digital en paralelo se dispone de 16 placas de accesorios. Cada una de estas placas ofrece, entre otros elementos, 8 interruptores y 8 leds. Unos conectores modulares tipo MOLEX de 9 vías permiten conectar los diferentes elementos de estas placas a los puertos A y B del 68230.

Dada la configuración física de la placa de accesorios y del 68fil, lo más sencillo es conectar los leds con el puerto A y los interruptores con el puerto B. Si hacemos esto y programamos el 68230 de modo que el puerto A funcione en el modo más simple de salida, y el puerto B en el más simple de entrada, entonces podremos controlar fácilmente qué leds queremos mantener encendidos en cada instante mediante operaciones de salida por el puerto A, y también saber en cualquier momento el estado de cada uno de los conmutadores de la placa de accesorios mediante una lectura del puerto B.

Como práctica:

Para programar los puertos A y B como acabamos de describir, basta elegir el modo 00 como control general de los puertos y el submodo 1X como control específico de ambos puertos A y B. Además, se han de configurar todas las líneas del puerto A como salida y todas las líneas del puerto B como entrada. Veamos cómo hacerlo:

```

PI_T    EQU    $800000    Dirección base del 68230
PGCR    EQU    PI_T+1
PADDR   EQU    PI_T+5
    
```

```

PBDDR EQU PI_T+7
PACR EQU PI_T+13
PBCR EQU PI_T+15
PADR EQU PI_T+17
PBDR EQU PI_T+19

```

```

MOVE.B 0x00000000,PGCR
MOVE.B 0x10000000,PACR
MOVE.B 0x10000000,PBCR
MOVE.B 0x11111111,PADDR
MOVE.B 0x00000000,PBDDR

```

A: puerto de salida
B: puerto de entrada

Tras estas instrucciones, ya podemos realizar operaciones de entrada/salida con los interruptores y leds de la placa de accesorios. Por ejemplo, podemos encender los 4 leds situados más a la izquierda con la instrucción:

```
MOVE.B 0x11110000,PADR
```

y si, a continuación, queremos leer el estado de los conmutadores en el registro D0, lo haremos con la instrucción:

```
MOVE.B PBDR,D0
```

Nótese que todos los movimientos a registros del 68230 son de tamaño byte. Para dar por finalizado el programa se deberá añadir un `SERMON FPROG`, o bien ejecutar paso a paso.

Una vez comprobado el funcionamiento de estas sencillas operaciones de entrada/salida, y configurando de igual modo el 68230, escríbase un programa que implemente un ciclo en el cual se encienda o apague cada uno de los leds según el estado del conmutador correspondiente (conmutador a 1 → led apagado, y viceversa). El ciclo debe terminar cuando todos los leds se apaguen, es decir, cuando todos los conmutadores estén a 1.

5.3. EJERCICIO PROPUESTO

Se trata de ir encendiendo consecutivamente un único LED, de modo que aparente moverse hacia la derecha o hacia la izquierda, y que cuando llegue a un extremo aparezca por el lado contrario, es decir, aparentando un movimiento circular continuo.

El sentido de la rotación se podrá cambiar en cualquier momento mediante el interruptor 7 (0=derecha, 1=izquierda).

En cuanto a la velocidad de movimiento del LED, ésta se controlará con los interruptores 0 a 6, mediante los cuales se codificará un valor entre 0,1 y 12,7 segundos de intervalo. Un valor cero provocará que el movimiento del LED se detenga. Cualquier cambio en la configuración de esos interruptores debe afectar de inmediato a la velocidad de rotación.

Evidentemente lo apropiado es emplear el timer del mismo 68230 para controlar la velocidad de rotación, reprogramando adecuadamente el CPR (Counter Preload Register) cuando se detecte un cambio en los interruptores del 0 al 6, y reiniciando el funcionamiento del timer para que tenga efecto inmediato (en caso de detener el LED lo lógico será dejar el timer parado). La rutina de servicio de la interrupción del timer puede limitarse a realizar el movimiento del LED. Recuérdese que la frecuencia del reloj de entrada para el timer es de 250 KHz, por lo que una cuenta inicial de 25000 equivale a una décima de segundo.