



CU9700442

1-97

C I E N - R

DIRECCIONADO VIRTUAL DE PUERTOS PARA PC

PORT VIRTUAL ADDRESSING FOR PC

Bolaños, L.; Arista, E.; Osorio, J.F.

Centro de Estudios Aplicados al Desarrollo Nuclear

La Habana, Cuba

1997

**POOR QUALITY
ORIGINAL**

28 No 13

DIRECCIONADO VIRTUAL DE PUERTOS PARA PC

PORT VIRTUAL ADDRESSING FOR PC

Bolaños L., Arista E. , Osorio J.F.

Centro de Estudios Aplicados al Desarrollo Nuclear (CEADEN)

1997

La Habana, Cuba

Subject Categories: E41.40

Key words: Equipment interfaces: M1; Design: Q1; Personal computers; Radiometers

Resumen.

Es muy frecuente en la actualidad diseñar equipos para medición de señales nucleares basados en una tarjeta de adición a una computadora personal (PC). En estos casos se tropieza muchas veces con el problema del direccionado de los dispositivos de entrada o salida de datos cuyo nivel de integración o 'inteligencia' hacen imprescindible el uso de varias direcciones de puerto, las cuales son limitadas en la PC. El direccionado virtual ofrece la ventaja del empleo de muy pocas direcciones para acceder a múltiples de estos dispositivos. En el artículo se describe el principio de esta técnica y la utilización en una tarjeta radiométrica de una solución en base a lógica programada.

Abstract.

Instruments for nuclear signal measurements based on an 'add-on' card for a personal computer (PC) are designed often. Then, one faces the problem of the addressing of data input/output devices which show an integration level or 'intelligence' that makes the use of several port addresses indispensable, and these are limited in the PC. The virtual addressing offers the advantage of the occupation of few addresses to accede to many of these devices. The principles of this technique and the appliance of a solution in a radiometric card based on programmed logic are discussed in this paper.

Introducción

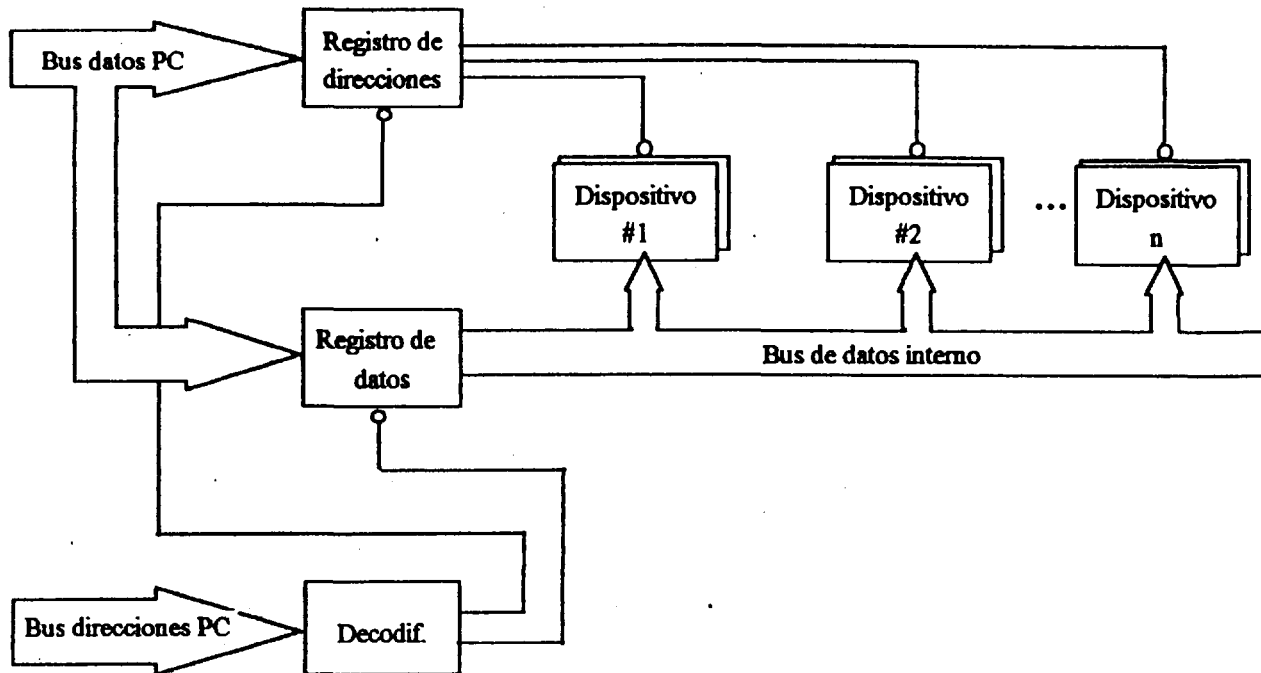
Cuando se diseñan tarjetas para adicionar a PC, frecuentemente se tropieza con el problema del direccionado de los dispositivos programables a utilizar y su inclusión en la tabla de direcciones disponible en la computadora. El problema se plantea muy claramente: se necesita una cantidad de direcciones de puertos disponibles mayor que la que ofrece como libre la microcomputadora. Esto está dado porque ya es muy común encontrar dispositivos con considerable grado de "inteligencia" o de integración controlables digitalmente (convertidores análogo-digitales múltiples en un sólo chip, temporizadores programables, amplificadores de ganancia programable, potenciómetros digitales, etc.).

Como ya se conoce, las direcciones libres de puerto que presentan las computadoras personales generalmente se ubica entre las localizaciones 300h y 31Fh. Esto da un total de 32 localizaciones de puertos que pudieran ser suficiente en muchas aplicaciones, pero que no lo resultan cuando un dispositivo programable emplea 2, 3 y hasta 8 localizaciones y se requieren varios con esas características en el mismo circuito.

Materiales y métodos

La solución que ofrece el direccionado virtual se muestra en la fig. 4. Se tienen 2 registros, llamémosles: "registro de direcciones" y "registro de datos". La función del

registro de datos, dado que su longitud de palabra es de 8 bits, es la de almacenar una dirección (que puede ser un valor desde 0 hasta 255) correspondiente al periférico que se desea direccionar. Por otra parte, el registro de datos, almacenará la información a escribir o leída del periférico en cuestión. El orden es importante: primero se pondrá la dirección en el registro de direcciones y luego se leerá el registro de datos o se le escribirá la información deseada.



Las salidas del registro de direcciones se conectarán por tanto a las señales de control de selección de los circuitos integrados (Chip Select u Output Enable).

Esta solución es aún más potente cuando se emplea para la decodificación alguna lógica programada que permita que los intervalos no sean iguales (lo cual limitaría el sistema, puesto que habría que prever como intervalo fijo el espacio que ocupe el dispositivo que necesite el mayor número de direcciones), sino que haga la selección de cada uno con diferentes espacios, dependiendo sólo de las ecuaciones lógicas que determinan cada selección. Por ejemplo, si puede programar un dispositivo de lógica pLSI o PAL con las siguientes expresiones:

$$CS_dir = a_9 \& a_8 \& a_7' \& a_6' \& a_5' \& a_0' \& AEN \& same \& ((b_4' \& a_4') \# (b_4 \& a_4)) \quad (1)$$

$$CS_dat = a_9 \& a_8 \& a_7' \& a_6' \& a_5' \& a_0 \& AEN \& same \& ((b_4' \& a_4') \# (b_4 \& a_4)) \quad (2)$$

donde:

- AEN es la señal de 'dirección activa' en el bus de direcciones de la PC
- $same = (b[1..3] == a[1..3])$ Es decir, será "1" si coinciden los bits b_1 a b_3 con a_1 a a_3 del bus de direcciones de la PC

En general el conjunto b1-b4 es un grupo de bits empleados para chequear la ocurrencia en el bus de direcciones de una combinación deseada. En la expresión dada, se manifiesta de dos maneras diferentes esta condición de igualdad. La señal same se pondrá en "1" lógico cuando b1-b3 sea igual a a1-a3; por otra parte el último producto de la expresión canónica de CS-dir y CS-dat expresa que cuando b4 sea igual a a4 ese término será "1" lógico.

Con el bit a0 se condicionan direcciones pares para el registro de direcciones e impares para el registro de datos. Las señales b1-b4 se pueden colocar a la entrada de la propia lógica programada, de manera tal que sea ella la responsable de comparar estos dos grupos de bits, y por tanto, determinar si se está accediendo a una de las direcciones válidas para estas expresiones. Recordemos que las direcciones 300h-31Fh están determinadas por los bits a9-a0 de la siguiente manera:

a9	a8	a7	a6	a5	a4	a3	a2	a1	a0	
1	1	0	0	0	X	X	X	X	X	(X = no importa, 0 ó 1)

Los bits superiores no son relevantes, por lo cual está implícito que este direccionado de puertos se repite a partir de direcciones superiores (700h-71Fh, B00h-B1Fh, F00h-F1Fh). Para evitar esto, se pueden incluir los bits más altos del bus de direcciones en este decodificador programado, con la correspondiente condición.

Resultados

Como resultado de esta combinación, se utilizan sólo 2 direcciones del conjunto libre de la microcomputadora, y se pueden direccionar hasta 255 dispositivos. El límite físico está dado realmente por la cantidad de pines de salida que tenga el decodificador en base a lógica programada, pero el límite lógico es 255, lo cual es mayor en todo caso que el que pueda suministrar un decodificador convencional.

En el ejemplo de la tarjeta que se trata, al registro de direcciones le corresponde la dirección 300h y al registro de datos la dirección 301h. Este intervalo es desplazable mediante un 'DIP switch' colocado en las entradas b1-b4 y con el cual se puede pasar al siguiente par de direcciones o al otro, y así sucesivamente, sólo es necesario poner la combinación digital necesaria en b4-b1.

La programación de la función antes expresada emplea 364 fusibles del total de 2048 de una PAL16V8. En términos de filas en su estructura interna sólo ocupa 10 de 64 disponibles (para un 16%) utilizando el compilador PLD de ORCAD 4.4. Se ha empleado en un sistema compuesto por un temporizador programable 8253 (4 localizaciones), un conversor D/A de 8 canales como el AD7228 de Analog Devices de 8 canales (8 más), y varios puertos de E/S paralela para el control de señales digitales para un motor, la habilitación de los contadores, etc. (supongamos al menos 2 localizaciones más). En la propia tarjeta es posible atender 2 canales con las mismas características, para lo cual se accede a un total de 28 direcciones.

Mostramos en la figura siguiente una sección de programa en QBasic, donde aparecen se ejemplifica el acceso a los dispositivos.

'Declaraciones en el programa principal

```
DECLARE SUB CSdese ()  
DECLARE FUNCTION leer (dir)  
DECLARE SUB mover (codi, cantp)
```

'Subrutinas para la lectura y escritura en las direcciones de puerto virtual

'Función que devuelve la lectura en un puerto designado por "dir"

```
FUNCTION leer (dir)  
OUT &H300, dir  
leer = INP(&H301)  
END FUNCTION
```

'Subrutina que escribe sobre un puerto designado por "dir"

```
SUB escribir (dir, dat)  
OUT &H300, dir  
OUT &H301, dat  
END SUB
```

'Hasta aquí las subrutinas

'Programa que chequea un bit de entrada en una dirección de puerto (BFh)

```
WHILE (leer(&HBF) AND &H4) = 0 'Chequea el bit 2 del Registro Estado  
CALL mover(codi, 50) 'Mueve el motor si no se ha cerrado  
CALL demora(150)  
WEND
```

'Programa que escribe sobre una dirección deseada

```
SUB CSdese  
'Subrutina que se mantiene escribiendo en una direccion  
'el valor que se desee  
LOCATE 16, 20: PRINT "Termina con <Ctrl+Break> "  
LOCATE 12, 20  
PRINT "Suponemos que la direccion y el dato estan en hexadecimal",  
LOCATE 10, 20: PRINT "Escriba solo DIRECCION,DATO :";  
INPUT " ", di$, da$  
di = VAL("&H" + di$): da = VAL("&H" + da$)  
WHILE 1 <> 0  
CALL escribir(di, da)  
WEND  
END SUB
```

Como se puede apreciar, son subrutinas muy simples, que se pueden llevar a cualquier otro lenguaje, y utilizar como módulos de acceso a los dispositivos, haciendo transparente para el programador la estructura del direccionado.

Discusión y conclusiones

Se puede apreciar que la selección del registro de datos se hará no sólo cuando se desee acceder al registro de datos, sino también obligatoriamente cuando se haga referencia al registro de direcciones. Es esta condición la que posibilita que la dirección (de 8 bits) del dispositivo pueda enviarse al registro de direcciones a través del bus de datos del circuito, los datos también y no haya conflicto.

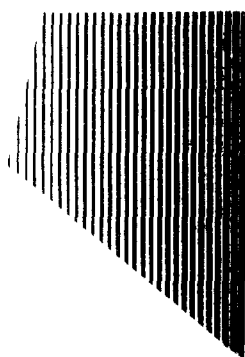
Puede parecer complejo este tipo de direccionado, pero una vez preparadas las subrutinas básicas para la lectura y escritura de estas direcciones, queda totalmente transparente y resulta cómodo para el programador el trabajo con esta configuración.

La cantidad de espacio que se salva permite la inclusión casi ilimitada de cuantos dispositivos sean necesarios en el sistema así como la conexión de otras tarjetas que necesiten direcciones de puertos.

Podemos argumentar además que la utilización de lógica programada en la decodificación de direcciones nos ofrece las ventajas de aumentar: posibilidades de conexión, confiabilidad, flexibilidad, nivel de compactación. Por otro lado, disminuye los costos del desarrollo y mucho más los de producción. Empleada como medio para el direccionado virtual, eleva también las posibilidades de reconfiguración de un sistema.

Bibliografía

1. Lehman, C. : Programmable Logic design tools. Language Guide, 1994. Capítulo 1. Pág. 1



CIE

**CENTRO DE INFORMACION
DE LA ENERGIA NUCLEAR**

Calle 20 No. 4113 e/ 18A y 47, Playa

Telf.: 22-7527. Fax: 331188.

E mail: cien @ceniai cu