

Contenido

1

Editorial

**A Single-Chip Digital Phase Meter For Vibration Analysis Of
Rotating Machinery**

Dr. Miguel Lindig Bos, Director del CIDETEC-IPN

3

8

El Proyecto Institucional de Telecomunicaciones del IPN

Lic. Elizabeth Acosta Gonzaga, Profesora del CIDETEC-IPN

**Introducción a la Teoría de Conjuntos Difusos «Fuzzy Set»
(Compendio)**

Ing. María Elena Aguilar Jáuregui, Profesora del CIC-IPN, Ing. Cuauhtémoc Peredo Macías,
Alumno de la Maestría del CIC-IPN.

11

14

Diseño Moderno de Circuitos Digitales Usando DSP'S

M. en C. Osvaldo Espinosa Sosa, M.en C. Romeo Urbieta Parrazales,
Ing. Marco Antonio Ramírez Salinas, Profesores del CIC-IPN

**El Espacio de Configuración de Dispositivos PCI
(Desarrollo de Rutinas de Acceso)**

Ing. Israel Rivera Zarate, Profesor del CIDETEC-IPN

20

Editorial

Nuestra revista tiene como objetivo fundamental la divulgación del quehacer tecnológico que en materia de Computo se realice no solamente en el Instituto, sino también fuera del mismo. A lo largo de su existencia, hemos recibido algunas colaboraciones que tienen como origen algún desarrollo conjunto con investigadores en el extranjero o son resultado de participaciones en Congresos o eventos de carácter internacional, por lo que su idioma original no es el español, sino el inglés fundamentalmente. Una solución a sido, en la mayoría de los casos, solicitar a los autores el envío de una versión en español o, si esto no es posible, su autorización para realizar dentro del Comité Editorial de polibits una traducción. Sin embargo, esto resulta en un retraso en la edición del artículo en cuestión, en cualquiera de las dos soluciones, y la posibilidad de inexactitudes en la traducción del documento, con el consiguiente demérito en la calidad y fidelidad de la información presentada a los lectores.

Actualmente, y no solamente por el aspecto de la Globalización, sino por un simple punto de vista práctico, todo aquel que de alguna forma se involucre con temas tecnológicos de actualidad (computación, mecánica, electrónica, química, etc.) tiene que tener, por necesidad un conocimiento, al menos al nivel de comprensión, del idioma Inglés. La mayoría de las fuentes originales del conocimiento y la divulgación básica de los desarrollos se realizan en este idioma, ya sea por ser el idioma natal del o los investigadores, porque el medio o foro en que se realiza la difusión de los mismos tiene al Inglés como idioma oficial o simplemente porque es una especie de regla establecida mundialmente que actualmente el idioma universal o de comunicación común entre pueblos y personas de diferentes lenguajes es el Inglés. Por otro lado, tanto en los medios académicos como

industriales y tecnológicos, no es posible admitir como pretexto el desconocimiento del idioma Inglés, dado que es una herramienta fundamental en el mundo de hoy. Aún con sus limitaciones y evidentes debilidades, el sistema educativo en México tiene establecidas materias de idioma extranjero (típicamente Inglés) desde el nivel medio (secundaria) hasta el nivel profesional, aún cuando esto ocurra en ocasiones a nivel de materias optativas.

Tomando esto en cuenta, la revista polibits a tomado el acuerdo de aceptar como posible segundo idioma de publicación al Inglés, en los casos en que el idioma original de la colaboración sea este y no exista en el momento de la edición una versión en español escrita por los autores y además este escrito en forma lo suficientemente legible si es que el idioma natal de los autores no sea el mismo. De esta forma, polibits asegura la mayor fidelidad a la divulgación que el o los autores desea presentar a los lectores de la revista, siendo completa responsabilidad de los mismos el tratamiento o la formalidad a las reglas gramaticales que le den al idioma en cuestión. Si el Comité Editorial toma la decisión que la redacción del documento resulta demasiado confusa como para ser entendible, se reserva el derecho de rechazar la colaboración hasta que el autor realice una revisión del mismo.

En este tenor, se presenta en este número una colaboración que cumple con los considerandos anteriores: realizada como resultado de una participación internacional, escrita directamente en Inglés y con el propósito de obtener una divulgación más amplia a nivel internacional. Aún cuando al autor se le podría solicitar una versión en español, el tiempo de su escritura y revisión sobrepasarían los límites de tiempo de edición mínimos para su inclusión, en forma oportuna, en esta revista.

A Single-Chip Digital Phase Meter For Vibration Analysis Of Rotating Machinery

Michael Lindig Bös
Director del CIDETEC - IPN
e-mail: mlindig@vmredipn.ipn.mx

The design of a digital phase meter, intended for vibration monitoring and analysis of electric power generators, is discussed. The design may be implemented in a single, field-programmable gate array, operating at 32 MHz. Inputs to the device are a trigger pulse and a sampled, bandlimited vibration signal. The device measures the phase angle of the signal at 1/2, 1 or 2 times the rotation frequency with a resolution of 1 degree, referred to the trigger signal. A novel frequency multiplier, as well as a tracking filter and zero-crossing detector are incorporated into the device. The trigger frequency range is 1 to 1000 Hz. A detailed error analysis, as well as experimental results, are presented.

I. INTRODUCTION

The monitoring and performance analysis of electric power generators is of great practical interest. The process involves the real-time processing of vibration signals and the measurement of several parameters, such as peak-to-peak amplitude and phase angle of the different harmonic components of the vibration signal as referred to a trigger signal obtained from the same machine.

In this work, the design of a digital phase meter is presented. The target specifications of the design are as follows:

Rotating speed:	60 to 60,000 rev/min.
Resolution:	1 degree
Precision:	± 1 degree at 0.5 and 1 times the frequency of rotation, ± 2 degree at 2 times the frequency of rotation.

The input signal to the device is a vibration signal, sampled at 102.4, 51.2, 25.6 or 10.24 KHz and bandlimited at 1/2.56 the sampling frequency. The target specifications apply to a signal amplitude range of 10 to 1.

A key component for the monitoring process is the tracking filter. The vibration signal is sampled at N times the frequency of rotation, where N is the desired angular resolution. The sampling frequency is obtained by a digital frequency multiplier that measures the time interval between successive trigger pulses, and the sampled signal is then filtered by a phase-compensated bandpass filter whose bandpass region is centered at $2\pi k/N$, where k is the desired harmonic component. Peak-to-peak amplitude and phase measurements may be obtained from the filtered signal, the latter by means of a zero-crossing detector.

Since the lowest practical bandpass centre frequency obtainable by a digital filter is limited both by the filter order and, for fixed-precision calculations, by the word size of the processor, higher resolution phase measurements are usually obtained by means of an interpolation process.

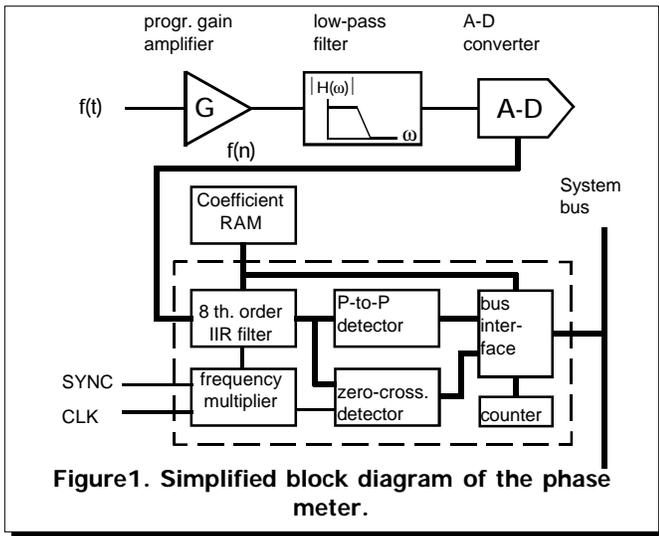
In what follows, a block diagram of the design and specifications of each functional unit are given. A performance analysis is presented, concentrating on the following aspects:

- 1.- The error generated by sampling two times the same signal at different sampling frequencies.
- 2.- The violation of the sampling theorem at low frequencies of rotation.
- 3.- The oversampling incurred in at high speeds of rotation.

Results of computer simulations for several signal qualities of the vibration signal are shown. Actual measurements validate, within the resolution of the available equipment, these simulations.

II. CIRCUIT DESCRIPTION

A block diagram of the relevant processing elements is shown in **figure 1**. The input signal, $f(t)$, is amplified and bandlimited. The output of the lowpass filter feeds a 12-bit analog to digital converter. The resulting sampled signal, $f(n)$, is one of two inputs to the phase meter. The other is a synchronization pulse (SYNC), occurring once for each rotation of the machine under study. The phase meter operates at a frequency given by a master clock (CLK), here, a 32 MHz oscillator. The main functional elements of the phase meter (enclosed by a dashed line) are a frequency multiplier, an 8th. order IIR filter, a zero-crossing detector and a counter. A peak-to-peak detector is included in the design, as well as a system bus interface. The filter coefficients and partial processing results are stored in a static RAM, which in the prototype is an external device. The filter coefficients are completely programmable by means of the system bus. In what follows, the main functional elements will be described in more detail.



II.1 THE FREQUENCY MULTIPLIER.

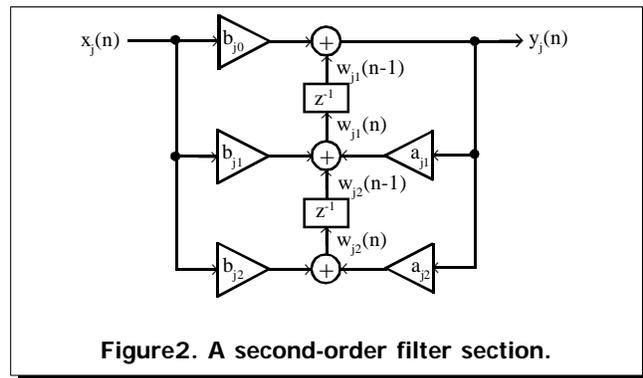
Ideally, this functional element generates an output pulse train whose frequency is given by $f_{OUT} = Nf_{IN}$, where f_{IN} is the input signal frequency and N a positive integer [1], [2]. Let τ_{OUT} be the period of the output signal. Since τ_{OUT} is derived from the system clock and constraint to be an integer number of system clock periods, the multiplication can, in general, not be realized exactly. A detailed description of the design solution used can be found in [3]. For applications where the output signal is used to sample a periodic signal harmonically related to the input signal of

the frequency multiplier, it has been shown that the maximum amplitude error is given by $\epsilon(n)_{max} < 2\sin(\pi/K)$, where $K = f_{CLK}/f_{IN}$. Also, the frequency multiplier introduces a delay of, at most, 1 sample interval.

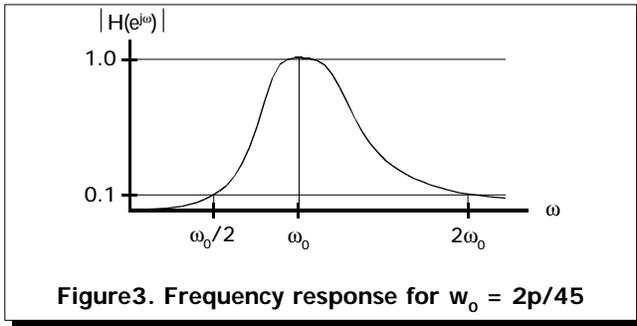
In the present application, the frequency multiplier operates at 1/2 the system clock frequency (16 MHz), and the maximum input frequency is 2KHz. For these values, the maximum amplitude error is 0.0393%, and the phase error introduced is at most 0.045 degrees. These errors are more than one order of magnitude smaller than the target specifications, and, in what follows, will be neglected.

II.2 THE IIR FILTER.

The device implements 4 cascaded second-order sections, as shown in **figure 2**. The index j designates the section number, a_{jk} and b_{jk} are filter coefficients and $w_{jk}(n)$ represent intermediate sequences. The structure is known as the transposed direct form II [4] and has the property of minimizing the storage requirements for a given filter order. The section is general, in the sense that lowpass, bandpass, highpass and allpass transfer functions can be realized. The design is based on fixed 16-bit precision, two's complement arithmetic. Values in the range $-2 \leq x \leq 2 - 2^{-14}$ may be represented.



In the present application, it is assumed that a fourth order bandpass Butterworth filter, followed by an allpass filter of the same order, is implemented. For the fundamental frequency component of the input signal, the filter was designed with a passband of $1.9\pi/45 \leq \omega \leq 2.1\pi/45$, using a commercial software package [5]. The allpass network was designed for a constant overall delay of 45 samples. **Figure 3** shows the frequency response of the filter. The maximum amplitude error in the passband region is 0.77%. No measurable phase error in the passband was observed.



II.3 PEAK-TO-PEAK AND ZERO CROSSING DETECTORS.

The peak-to-peak amplitude and phase measurements are derived from the output samples of the filter, $y(n)$. For a given input period, and for each sample interval, the value $p(n) = y(n) - y(n-1)$ is calculated. If $p(n)$, $y(n)$ and $y(n-1)$ are all positive and $p(n+1)$ is negative, $y(n)$ is stored as the positive peak value. Similarly, if $p(n)$, $y(n)$ and $y(n-1)$ are all negative and $p(n+1)$ is positive, $y(n)$ is stored as the negative peak value. The sum of the magnitudes of both peak values is reported at the occurrence of the next SYNC pulse.

The event of zero crossing is defined to occur during the sample period between samples $y(n)$ and $y(n-1)$, such that $y(n) \geq 0$ and $y(n-1) < 0$. Note that the sampling intervals are synchronized to the SYNC pulse. Since the sampling period equals 8 degrees, linear interpolation is used to achieve the desired 1-degree resolution. Specifically, the device implements the recursion:

$$\delta(k+1) = \delta(k) - [y(n) + |y(n-1)|], \text{ with}$$

$$\delta(0) = 8 |y(n-1)|$$

for all values of k such that $\delta(k) > 0$.

That is, the expression $[8 |y(n-1)|] / [y(n) + |y(n-1)|]$ is evaluated by successive subtractions. Clearly, the final value of k equals the integer part of the expression given above. This value is added to 8 times the count of sampling intervals between the occurrence of a SYNC pulse and the sample $y(n-1)$, thus yielding the phase angle in degrees.

III. ERROR ANALYSIS

Let f_R be the frequency of rotation of the machine. Then, $Nf_R = f_M$ is the output frequency of the frequency multiplier. Denote by f_m the sampling frequency. Two situations may be distinguished, namely either $f_m \leq f_M$, or

$f_m < f_M$. If $f_m \leq f_M$, the vibrations signal is, in general, not bandlimited in terms of f_M . Also, since both sampling frequencies are uncorrelated, there will be an error due to the uncertainty of the precise sampling instant. On the other hand, if $f_m < f_M$, the (sampled) vibration signal is oversampled. In what follows, bounds on these sources of error will be established. Because of space limitations, only the fundamental frequency component of the vibration signal will be considered, that is, $f(t) = \sin(2\pi f_R t)$.

III.1 ERROR DUE TO UNCERTAINTY IN THE SAMPLING INSTANTS

If $f(t)$ were sampled by the output of the frequency multiplier, the sampling interval would be $(f_m/f_M)\tau_m$, where $\tau_m = 1/f_m$. Since $f(t)$ is already sampled at τ_m intervals, the signal can be observed only at integer multiples of (f_m/f_M) . Denote by $\lfloor x \rfloor$ the greatest integer less or equal to x . Define an error signal, $\epsilon(n)$:

$$\epsilon(n) = \sin 2\pi(nf_m/f_M)\tau_m - \sin 2\pi \lfloor nf_m/f_M \rfloor \tau_m \dots\dots\dots 1$$

for n a positive integer.

Since $(nf_m/f_M) < \lfloor nf_m/f_M \rfloor + 1$ for all n , and using a trigonometric identity, from (1) results:

$$\epsilon(n) < 2\sin(\pi\tau_m)\{\cos\pi\tau_m[(2nf_m/f_M) - 1]\} 2$$

and it follows that:

$$\epsilon(n)_{\max} < 2\sin\pi\tau_m \dots\dots\dots 3$$

At the lowest sampling frequency (10.24 KHz), the amplitude error given by (3) is 0.0614%. The delay introduced by the double sampling process is clearly less than τ_m . Since the sampled signal is further processed by a bandpass filter, it is reasonable to assume that at its output the delay tends toward a mean value. This has been confirmed by experiment. The mean delay is, of course:

$$\delta(n)_{AV} = (\tau_m/N) [(nf_m/f_M) - \lfloor nf_m/f_M \rfloor] \dots\dots\dots 4$$

It can be shown that, for f_m/f_M not an integer,

$$0.25\tau_m \leq \delta(n)_{AV} < 0.5\tau_m$$

Taking the lower bound as a systematic error, it follows that for phase errors of less than 1 degree, $f_R < 360/4f_m$. For the sampling frequencies specified, the upper limit on the rotating speed is, respectively, 1138, 569, 284 and 114 rev/sec.

III.2 ERROR DUE TO ALIASING

Assume the input to a bandpass filter with center frequency $2\pi/N$ is a periodic, non-bandlimited signal, with frequency components located at $2\pi n/N$, $n = 0, 1, \dots$. Then, the output of the filter contains frequency components located at $2\pi/N$, $2k\pi(N-1)/N$ and $2k\pi(N+1)/N$, where $k = 1, 2, \dots$.

The amplitude spectrum of physical signals decreases at increasing signal frequencies, according to a rate of the form $1/\omega^\alpha$. That is, the normalized amplitude of the aliased frequency components is $1/(kN-1)^\alpha$ and $1/(kN+1)^\alpha$, respectively. The sum of these amplitudes is, then:

$$\sum_{k=1}^{\infty} \left[\frac{1}{(kN-1)^\alpha} + \frac{1}{(kN+1)^\alpha} \right] \approx \frac{2}{N^\alpha} \sum_{k=1}^{\infty} \frac{1}{(k)^\alpha} \dots\dots\dots 5$$

Now, the sum in (5) converges for $\alpha > 1$. For vibration signals, α can be taken to be ≥ 2 . In particular, for $\alpha = 2$, the sum converges to $\pi^2/6 \approx 1.645$ [6]. For $N = 45$, the sum of the amplitudes of the aliased frequency components is then 0.00162, and the worst-case phase error is 0.093 degrees.

III.3 ERROR DUE TO OVERSAMPLING THE SAMPLED SIGNAL

If $f_m < f_M$, during a sample period $1/f_m$ a certain number of consecutive samples taken at $1/f_m$ intervals will have equal values. Assume $f_M/f_m = R$ to be an integer. Then, the oversampling process may be modelled as a simple FIR filter of order $R-1$, shown in figure 4.

The Z-transform of the transfer function is given by:

$$H(z) = \sum_{r=0}^{R-1} z^{-r} = \frac{1-z^{-R}}{1-z^{-1}} \dots\dots\dots 6$$

And the frequency response is easily obtained as:

$$H(e^{j\omega}) = e^{-j\omega(R-1)/2} \frac{\text{sen}\omega R/2}{\text{sen}\omega/2} \dots\dots\dots 7$$

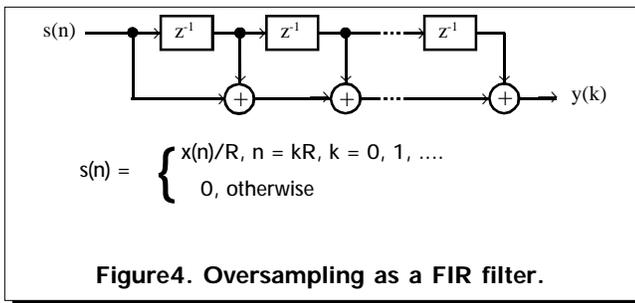
That is, oversampling has a lowpass filter effect and introduces a delay of $(R-1)/2$ samples. Ec. (7) gives average values if R is not an integer, because for any given input sample period $1/f_m$, the number of equal samples obtained at the rate $1/f_M$ is an integer (either $\lfloor R \rfloor$, or $\lfloor R \rfloor + 1$).

In the present application, the maximum oversampling rate is $N(f_{r,max})/(f_{m,min})$, that is, 4.395. From (7), and for $\omega = 2\pi/45$, the attenuation is $|H(e^{j\omega})|/4.395 = 0.985$, and the delay 1.697 samples, or 13.578 degrees. This delay can be corrected, since the sampling frequency is known, and the rotation frequency is measured by the frequency multiplier.

III.4 ERROR DUE TO NOISE

The phase angle measurement error due to this source depends on the type and specifications of the bandpass filter. Clearly, in order to reduce this error, a narrow bandpass region is desirable. On the other hand, the filter parameters are influenced by the maximum expected acceleration of the rotation frequency. Let $\tau_{IN}(n)$ be the period of rotation. The output of the frequency multiplier is, then, $\tau_{IN}(n-1)/N$, that is, the output frequency is defined by the previous input period. Hence, if the machine accelerates, the centre frequency of the passband filter is no longer coincident with the frequency of rotation and the filter cannot be too narrow in the passband.

Computer simulations have been performed for various synthetic signals, and for the filter specified in II.2. Specifically, the phase error due to added noise signals of uniform distribution and zero mean was measured (figure 5). The observed variance of the phase error increases linearly with the variance of the noise signal, reaching 1 degree for a noise with a standard deviation of 0.212.



IV. PHYSICAL IMPLEMENTATION

A 4-channel prototype was built, based on 2 field-programmable gate arrays type A1280XL from ACTEL [7], providing each the equivalent of 20,000 gates. The functional unit requiring most resources is, of course, the multiplier used in the implementation of the digital filter. In order to minimize these resources, a simple shift-add

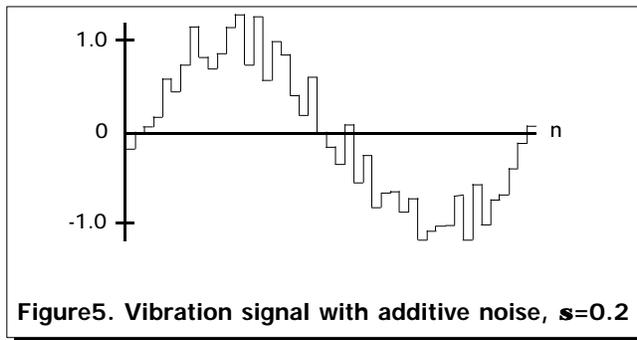


Figure 5. Vibration signal with additive noise, $\sigma=0.2$

algorithm was used. Each multiplication is executed in 16 clock cycles, that is, the second-order section shown in fig. 2 executes in 80 clock cycles. It is estimated that a single-channel design, operating at 32 MHz, could be implemented in a single FPGA and one external static RAM. The storage requirements depend on the filter order. For each second-order section, 7 16-bit words, or 14 bytes, are required. The prototype implements an 8-bit memory interface, and, for ease of design, uses 16 bytes for each second-order section.

V. CONCLUSIONS

The design of a digital phase meter has been presented, that lends itself well to single-chip implementations. The design objectives have, in general, been satisfied. The interpolation algorithm of the phase measurement has yet to be refined by increasing its resolution, and providing rounding instead of simple truncation. The most important source of error is the delay introduced by the double sampling process. This delay can, however, be calculated, since both the sampling frequency and the output frequency of the frequency multiplier are known for each input signal period.

Another important error is the aliasing error, especially at low input frequencies. This suggests the use of higher values of the multiplication factor N (see eq. 5), and, hence, higher-order passband filters. If a 5th-order filter operating at a bandpass centre frequency of $2\pi/90$ were implemented, the design objectives would be exceeded. For the design solution discussed, this means an increase of the operating frequency from 32, to 40 MHz.

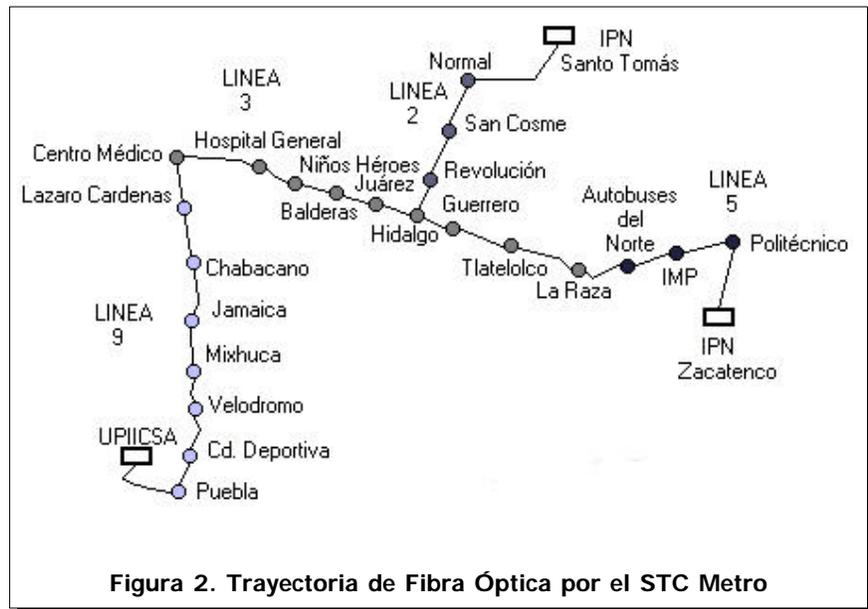
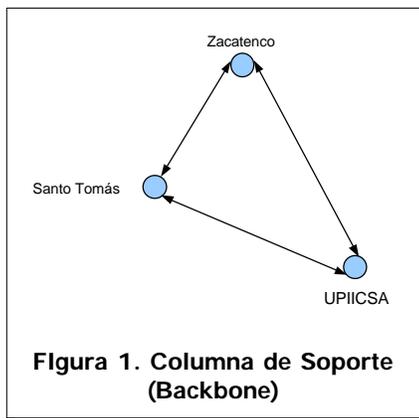
REFERENCES

- [1] H.Y. Lo, J.H. Lu. "A simple design for a digital programmable frequency multiplier". *Int. J. Electron.*, vol 46 no. 5, pp.535-5542, Dec. 1979.
- [2] N. Boutin, A. Boucher. "A novel digital frequency multiplier". *IEEE Trans. Instrum. Means.*, vol 35 no. 4, pp.556-570, Dec. 1986.
- [3] M. Lindig Bös. "A simple, high-precision, high-speed digital frequency multiplier". *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, Sacramento, Cal., Aug.3-6, 1997.
- [4] Oppenheim, A.V., Schafer, R.W. "Digital Signal Processing". *Prentice-Hall*, 1975.
- [5] DFDP3/plus Instruction Manual. Atlanta Signal Processors, Inc., 1992.
- [6] Knopp, K. "Infinite Sequences and Series". *Dover*, 1956.
- [7] FPGA Data Book and Design Guide. ACTEL Corp., 1995.

El Proyecto Institucional de Telecomunicaciones del IPN

Lic. Elizabeth Acosta Gonzaga
Profesora e Investigadora del CIDETEC-IPN

Al principio de la presente década, las redes de cómputo en el Instituto Politécnico Nacional se encontraban aisladas, es decir, cada unidad contaba con una red propia (en el mejor de los casos) y no existía comunicación entre ellas, por lo que se tuvo la necesidad de plantear una estructura que enlazara a todas las escuelas, centros y unidades. Así, en el año de 1995 se propone el Proyecto Institucional de Telecomunicaciones, con base en una estructura moderna para asegurar un sistema de comunicación involucrando la transmisión de voz, datos y video entre las 22 Escuelas de Nivel Superior, 16 Escuelas de nivel Medio Superior, 6 Centros de Investigación y 11 Unidades de Areas Centrales.



IMPLEMENTACIÓN

En 1996 se inicia la construcción de la llamada Red Institucional, con una infraestructura física constituida por tres nodos conectados entre sí por medio de 96 km. de fibra óptica (formando un triángulo), además de 21 enlaces de microondas para completar y respaldar la conexión de fibra, permitiendo al IPN integrarse al directorio mundial de Instituciones de Nivel Superior a la vanguardia en Telecomunicaciones. De esta forma, la columna de soporte (backbone) del proyecto de telecomunicaciones se ve en la figura 1.

Con este esquema se asegura que siempre existe comunicación entre los nodos, es decir:

- UPIICSA-ZACATENCO
- UPIICSA-SANTO TOMÁS
- ZACATENCO-SANTO DE TOMÁS
- ZACATENCO-UPIICSA
- SANTO TOMÁS-UPIICSA
- SANTO TOMÁS-ZACATENCO

Para unir a los nodos del backbone principal, la trayectoria de fibra óptica sigue principalmente la ruta de las líneas del Sistema de Transporte Colectivo (metro), con canalización privada hasta cada nodo. Así, la trayectoria de fibra que une a los tres nodos se muestra en la figura 2.

La comunicación entre los nodos se realiza en base a enlaces con una capacidad de 288E1 (1E1=2.848 Mbps) cada uno. De igual forma, como parte de las estrategias de seguridad y redundancia existen 3 enlaces

de microondas que respaldan a los enlaces de fibra, cada uno con capacidad de 8E1.

De esta manera se asegura que exista siempre comunicación entre nodos, consiguiendo doble redundancia, ya sea siguiendo cualquiera de las trayectorias alternas o por un medio diferente, que puede ser fibra óptica o vía microondas; por ejemplo si el nodo UPIICSA desea comunicarse con el nodo Zacatenco tiene varias alternativas:

- fibra óptica directamente a Zacatenco
- microondas directamente a Zacatenco
- fibra óptica a través de Santo Tomás
- microondas a través de Santo Tomás

Esta estructura sirve de soporte para enlazar a las diferentes unidades del Instituto a su correspondiente nodo más cercano, y de esta manera asegurar que exista comunicación entre todas las escuelas del IPN. Las instalaciones que se encuentra cercanas a los nodos principales y que no pueden ser enlazadas a través de fibra óptica se comunican al nodo más cercano utilizando microondas, por ejemplo para el caso del nodo UPIICSA, éste mantiene enlaces con nueve escuelas (redes secundarias) para lo cual se cuenta con módems DMS para enlaces microondas con capacidad de 8E1 y 4E1; las redes secundarias que forman parte del nodo UPIICSA se muestran en la **figura 3** y sus respectivas capacidades de transmisión se muestran en la **tabla 1**.

De esta forma las nueve escuelas correspondientes al nodo UPIICSA intercambian información, ya sea voz o datos con las otras escuelas o unidades del IPN. Por ejemplo, si la Esca Tepepan desea comunicarse con Zacatenco, dicha comunicación se

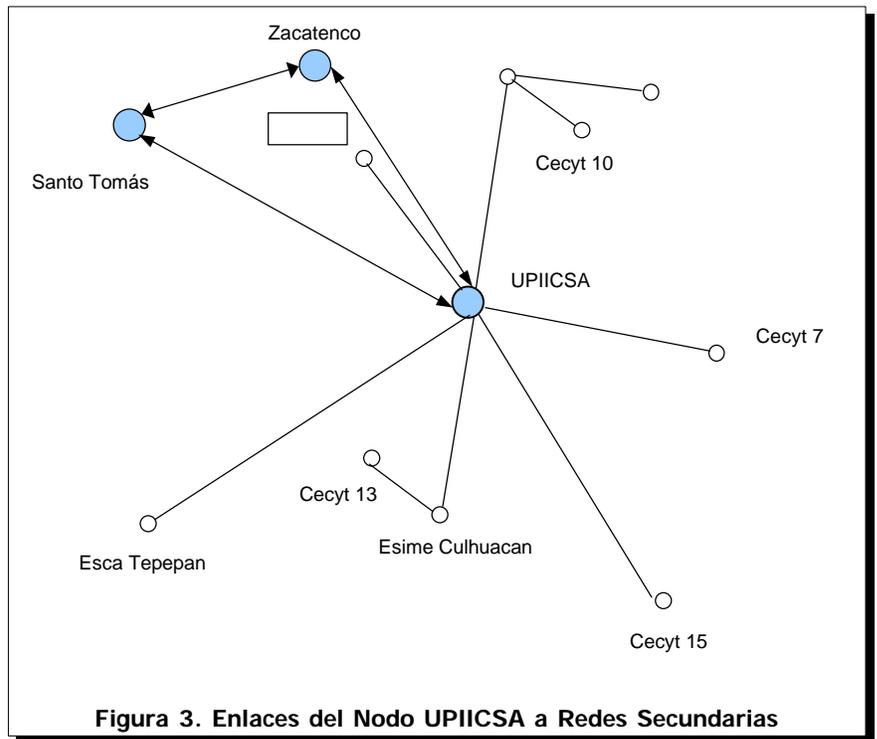


Figura 3. Enlaces del Nodo UPIICSA a Redes Secundarias

realiza vía microondas hasta el nodo UPIICSA, y de éste hacia Zacatenco ya sea por fibra óptica o por microondas; es decir,

Esca Tepepan-UPIICSA, UPIICSA-Zacatenco.

Para los casos del Cecyt 13, Cecyt 10 y Cet 1, la comunicación se realiza a través de un nodo intermedio; en el caso del Cecyt 13 dicho nodo es Esime Culhuacan; así, si desea comunicarse con Santo Tomás, la comunicación se realiza vía microondas hasta el nodo intermedio Esime Culhuacan, de Esime Culhuacan vía micro-

ondas hasta el nodo UPIICSA, y de éste hacia Santo Tomás por alguno de los dos medios disponibles.

En los casos del Cecyt 10 y del Cet 1, la comunicación es a través del nodo intermedio Cecyt 1; así, la comunicación del Cecyt 10 hacia Santo Tomás es Cecyt 10 vía microondas al Cecyt 1, del Cecyt 1 a UPIICSA también por microondas y del nodo UPIICSA al nodo Santo Tomás por cualquiera de los dos medios disponibles. Una situación similar aplica para el caso de Cet 1.

ENLACES	CAPACIDAD	CAPACIDAD	CAPACIDAD
	Actual	En uso	Libre
UPIICSA-Esca Tepepan	4E1	2E1	2E1
UPIICSA-Esime Culhuacan	4E1	4E1	2E1
Esime Culhuacan-Cecyt 13	4E1	4E1	-
UPIICSA-Cecyt 15	4E1	2E1	2E1
UPIICSA-Cecyt 7	4E1	2E1	2E1
UPIICSA-Cecyt 1	8E1	6E1	2E1
Cecyt 1-Cet 1	4E1	4E1	-
Cecyt 1-Cecyt 10	4E1	4E1	-

Tabla 1. Capacidades de redes secundarias del nodo UPIICSA

El nodo Santo Tomás une a varias escuelas y unidades por medio de enlaces de microondas; las escuelas cercanas al mismo (Canal 11) se enlazan por un anillo FDDI* (Fiber Distributed Data Interface). Para el caso del nodo Zacatenco, éste cuenta con otro anillo de FDDI, el cual une todas las escuelas y unidades correspondientes a esa área.

Como parte de la red de IPN se tendieron 1,200 Km de cableado estructurado, el cual incluye a todas las escuelas, centros y unidades, utilizando 7,500 salidas de ellas corresponden 5,000 para voz y 2,500 de datos. Para las salidas de datos se usó cable UTP categoría 5, mientras que para las salidas de voz se aplicó UTP categoría 3.

Finalmente, una porción de la red institucional está formada por enlaces satelitales; la estación satelital maestra se encuentra ubicada en el Edificio Inteligente en Zacatenco. En un inicio el propósito de éstos era comunicar a siete centros foráneos ubicados en el interior del país para compartir voz, datos y video; debido a las necesidades actuales dichos enlaces se han expandido hasta llegar a un número de 13, comunicando las unidades de:

- Vicente Guerrero, Durango.
- Jiquilpan, Michoacán.
- La Paz Baja, California Sur.
- Oaxaca, Oaxaca.
- Tijuana, Baja California Norte.
- Yautepec, Morelos.
- Milpa Alta, DF.
- Morelia, Michoacán.
- Cancún, Quintana Roo.
- Reynosa, Tamaulipas.
- Tampico, Tamaulipas.
- Culiacán, Sinaloa.
- Guasave, Sinaloa.
- Altamira, Sinaloa (próxima instalación).

Para la administración de la Red se cuenta con un Sistema de Monitoreo, el cual proporciona soporte a cada uno de los servicios, teniéndose atención para:

- Telefonía
- Datos
- Videoconferencia
- Microondas
- Fibra óptica
- Satélite

El IPN se comunica a las Redes de Datos Nacionales e Internacionales y con Internet, a través de la Red Digital Integrada de Teléfonos de México y Avantel, con una capacidad de 3E1 por cada enlace.

Algunas unidades incluyen varias redes locales; así, en el caso del nodo UPIICSA éste enlaza ocho redes locales instaladas dentro de la misma unidad, para lo cual se tiene un sistema de cableado estructurado para cada red. Dichas redes se unen por medio de fibra óptica a un concentrador inteligente Bytex y a un Ruteador CISCO, para el intercambio de datos con los otros nodos del backbone.

En relación a las otras prestaciones, la Red de Telefonía Digital proporciona Servicios Telefónicos de alta calidad a toda la comunidad de IPN, con 5,000 puertos telefónicos operados con 53 conmutadores. Por su parte, los servicios de videoconferencia en el Distrito Federal cuentan con equipos y salas acondicionadas ubicadas en:

- El Edificio Inteligente de Cómputo y Comunicaciones.
- El Centro Cultural Jaime Torres Bodet en Zacatenco.
- La Escuela Superior de Comercio y Administración.
- El Casco de Santo Tomás.

Y en el interior de la República en:

- Tijuana, BCS.
- Reynosa, Tamps.
- Morelia, Mich.

* Estándar para transmisión de datos en redes locales utilizando fibra óptica a una velocidad de 100 Mbps. Utiliza un anillo doble similar a Token Ring.

Introducción a la Teoría de Conjuntos Difusos «Fuzzy Set» (Compendio)

*Ing. María Elena Aguilar Jáuregui
Profesora Investigadora del CIC-IPN.
Ing. Cuauhtémoc Peredo Macías
Alumno de la Maestría del CIC-IPN.*

El presente trabajo tiene como objetivo introducir al lector en los orígenes, conceptos y operaciones básicas de la Teoría de Conjuntos Difusos. Dado que el lector puede estar más familiarizado con el término «Fuzzy», a lo largo del trabajo se empleará ésta palabra o bien, su traducción «difuso». El uso de cualquiera de los dos términos hace referencia al mismo concepto.

INTRODUCCIÓN

Al paso del tiempo, los seres humanos hemos aprendido a utilizar la experiencia derivada de la interacción diaria con nuestro medio ambiente. Esta experiencia traducida en conocimiento nos proporciona la habilidad de predecir eventos futuros, tomando como referencia evidencias básicas obtenidas del devenir diario.

De esta forma podemos tomar decisiones en el manejo y análisis del cúmulo de datos a nuestra disposición. Existiendo una gran variedad de fuentes de información, surge la necesidad de establecer cierto orden para lograr un manejo eficaz de los recursos informativos, trayendo como consecuencia el hecho de efectuar mediciones. Sin embargo, todas las mediciones involucradas en cualquier

de los ámbitos posibles, sin importar que tan precisas sean, admiten la posibilidad de error, por lo que existe cierta incertidumbre. Esta incertidumbre se deriva de las limitantes que se tienen en cuanto a la forma de percibir nuestro medio, y que tanto o tan profundo se puede razonar acerca de él; además de que todas nuestras mediciones están hechas tomando como base un estándar artificial.

La incertidumbre es uno de los factores principales por los que el conocimiento pierde precisión. Otro factor es el hecho de que para describir y comunicarlo, se hace uso del lenguaje natural. La forma en que éste afecta a la transmisión del conocimiento, es la interpretación que se le da a las palabras, provocando una asociación distinta para diferentes personas. Por lo tanto, el lenguaje natural tiene hasta cierto punto, la característica de ambigüedad.

SURGIMIENTO DE LA TEORÍA DE CONJUNTOS DIFUSOS

Desde el comienzo de la ciencia moderna, hasta el final del siglo XIX, la incertidumbre se consideró como un elemento indeseable, y la idea principal fué evitarla. Esta actitud cambió gradualmente con el surgimiento de la mecánica estadística al comienzo del siglo XX. Para tratar con la inmanejable complejidad de los procesos mecánicos a nivel molecular, la

mecánica estadística recurrió al uso de promedios estadísticos y a la Teoría de Probabilidad. Sin embargo esta última no es capaz de manejar la incertidumbre en todas sus manifestaciones, en particular, no puede manejar la incertidumbre resultante de la ambigüedad producida al utilizar términos lingüísticos en el lenguaje natural.

Estas limitaciones son parte de las razones por las cuales se concibió una nueva teoría de incertidumbre, capaz de tratar con elementos como ambigüedad e imprecisión.

El origen de ésta teoría se puede considerar como la publicación de un artículo de Lofti A. Zadeh en 1965. Zadeh se interesó en los problemas de sistemas complejos y en cómo representarlos utilizando modelos simples. Dado que las herramientas matemáticas tradicionales resultaron inadecuadas para este propósito, Zadeh introdujo el concepto de «Conjunto Difuso (Fuzzy)»; un conjunto cuyo contorno no es definido o preciso.

EVOLUCIÓN DE LA TEORÍA DE CONJUNTOS DIFUSOS

Al inicio de los años 70's surgió la idea de realizar «Control Fuzzy», diseñando y construyendo los primeros controladores fuzzy en esa misma década. Estos controladores se basan

en reglas de inferencia establecidas en lenguaje natural y representadas por conjuntos fuzzy. En esta década también evolucionaron ideas concernientes al uso de conjuntos fuzzy en reconocimiento de patrones y se formó la Asociación Internacional de Sistemas Fuzzy (IFSA; International Fuzzy Systems Association), llevando a cabo la publicación de la revista de esta asociación, «Fuzzy Sets and Systems».

Durante los 80's además de darse un desarrollo en la Teoría de los Conjuntos Fuzzy, se realizaron aplicaciones prácticas de los descubrimientos teóricos. Como resultado de las investigaciones, especialmente en Japón, se han fabricado productos cuyo diseño funcional está basado en la Teoría de Conjuntos Fuzzy. Esta tendencia ha evolucionado de tal manera que ahora existen, por ejemplo, automóviles cuyas transmisiones y frenos antiderrapantes son controladas por circuitos lógicos fuzzy.

Existen diversas áreas donde se ha descubierto la utilidad de aplicar la teoría Fuzzy a sus actividades; por ejemplo, finanzas, psicología, química, ecología, economía, etc.; además de aquellas actividades de ingeniería tales como sistemas expertos, sistemas de información, robótica, etc. entre otras.

**CARACTERÍSTICAS DE LOS
CONJUNTOS FUZZY**

El concepto de un conjunto cuyo contorno no es definido, contrasta con el concepto clásico de conjuntos, actualmente denominado conjunto preciso o claro (Crisp Set), cuyo contorno se requiere que sea bien definido. Esto es, un conjunto preciso o crisp, es un grupo de objetos para los cuales es sabido si cualquier objeto dado pertenece al conjunto o no.

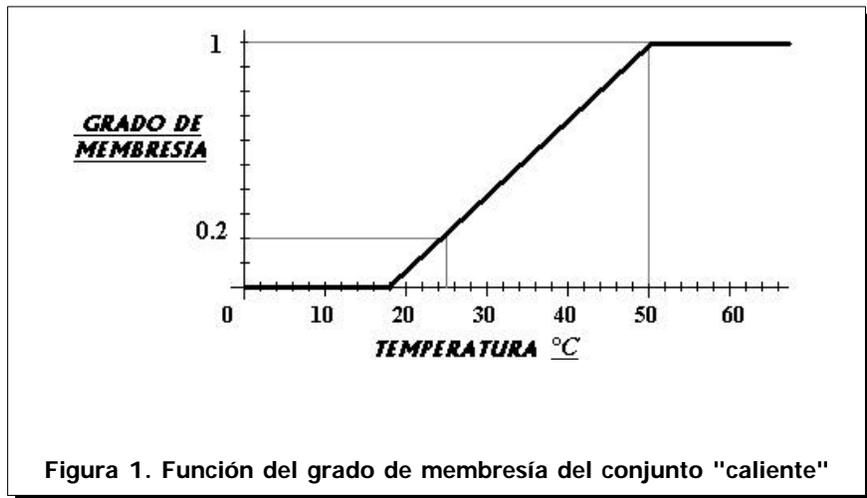


Figura 1. Función del grado de membresía del conjunto "caliente"

Contrarios a los conjuntos precisos clásicos, los conjuntos fuzzy (o difusos) no tienen contornos definidos. El ser miembro de un conjunto fuzzy no significa que se pertenezca o no de una manera definitiva al mismo. Un miembro puede pertenecer al conjunto en un mayor o menor grado.

Se puede definir a un conjunto fuzzy asignando a cada elemento del mismo, un número entre 0 y 1, el cual indica el grado de membresía o pertenencia al conjunto, dependiendo del contexto en el cual se establezcan las características del conjunto fuzzy.

Por ejemplo, supóngase un conjunto fuzzy denominado «caliente», cuyos miembros son las temperaturas a las cuales se considera que un objeto está caliente. Se establece que ese objeto deja de estar frío a los 25°C y llega a su máxima temperatura a los 50°C.

De esta forma, a una temperatura de 0°C, el objeto definitivamente no está caliente, por lo que a 0°C se le asigna un grado de membresía de 0. A una temperatura de 25°C el objeto ya no está frío, pero tampoco está totalmente caliente, por lo que para esta temperatura el grado de membresía es de 0.2. A una temperatura

de 50°C en adelante, el objeto está totalmente caliente, por lo que a esas temperaturas se les asigna el máximo grado de membresía, que es de 1.

En general, el grado de membresía de un elemento en un conjunto fuzzy, se incrementa conforme el elemento satisface las características establecidas para el conjunto fuzzy dado. A la asignación del grado de membresía a cada elemento en el conjunto fuzzy, se le denomina *Función del Grado de Membresía* del conjunto fuzzy.

Un ejemplo de esta función, reflejando el contexto particular del conjunto «caliente» se muestra en la **figura 1**.

Finalmente, cada conjunto fuzzy es determinado por una función particular del grado de membresía, la cual asigna a cada elemento de interés su grado de membresía en el conjunto. Aunque no es necesario, es conveniente expresar el grado de membresía por números entre 0 y 1.

BIBLIOGRAFÍA

- [1] George J. Klir, Bo Yuan, Ute H. St. Clair. *«Fuzzy Set Theory»*. Prentice Hall, 1997.
- [2] Timothy J. Ross. *«Fuzzy Logic with Engineering Applications»*, McGraw Hill, 1995.
- [3] M. J. Patyra, D. M. Mlynek. *«Fuzzy Logic, Implementation and Applications»*, Wiley Teubner, 1996

Diseño Moderno de Circuitos Digitales Usando DSP'S

M. en C. Osvaldo Espinosa Sosa
M. en C. Romeo Urbieto Parrazales
Ing. Marco Antonio Ramírez Salinas
 Profesores del CIC-IPN

Los seres humanos interactuamos con el mundo real en base a señales analógicas, las cuales pueden ser presión, intensidad luminosa, calor, sonidos etc. Dichas señales son variables continuas en el tiempo. Cada uno de nuestros sentidos es sensible a diferentes tipos de señales analógicas. Nuestros oídos son sensibles al sonido, nuestros ojos a la luz etc., una vez que recibimos una señal, nuestros órganos sensoriales se encargan de convertirla en una señal eléctrica y la mandan al cerebro, nuestra gran computadora analógica. El cerebro es una computadora sumamente poderosa cuya capacidad no es posible alcanzar con ninguna de las computadoras digitales actuales, ya que no solo analiza la información recibida, sino que también es capaz de tomar decisiones usando esos datos.

Desafortunadamente las computadoras trabajan con señales digitales, y no analógicas, es por eso que para poder aprovechar la capacidad y versatilidad que nos ofrecen, debemos hacer lo siguiente:

1. Convertir las señales analógicas en señales eléctricas, utilizando un transductor (como un micrófono).
2. Digitalizar esas señales, es decir convertirlas de su forma análoga a

digital por medio de un dispositivo llamado ADC (Convertidor Analógico – Digital).

Una vez que la señal se ha convertido a su forma digital, la computadora puede procesarla. Dado que el sistema procesa señales digitales, este es llamado Procesador Digital de Señales (DSP), y este hecho marca una diferencia respecto a otros microprocesadores y microcontroladores de propósito general. Una vez que el DSP ha procesado a la señal, se debe volver a su forma original (analógica) para que podamos percibir el efecto. Este último proceso lo realiza un Convertidor Digital – Analógico (DAC). Un altavoz, por ejemplo, puede reproducir señales audibles en base a las señales eléctricas que nos entrega un DAC. Para procesar una señal en forma digital debemos convertirla al menos dos veces (en este esquema general de procesamiento).

¿DE QUÉ CONSTA UN SISTEMA TÍPICO DE PROCESAMIENTO DE SEÑALES?

Básicamente consta de un chip al que denominamos DSP, elementos de memoria, posiblemente un convertidor digital – analógico (DAC), uno analógico – digital (ADC) y canales de comunicación con el mundo externo (puertos). No todos los sistemas de procesamiento de señales deben tener la misma arquitectura o

el mismo número de componentes, esa selección depende de la aplicación. Por ejemplo un sistema de audio puede requerir convertidores ADC's y DAC's mientras que uno para procesar imágenes no.

El DSP es un chip que contiene diferentes elementos de hardware, algunos de los elementos más importantes se mencionan a continuación:

Unidad central

aritmético – lógica. Esta es la parte del DSP que ejecuta la mayor parte de las operaciones o instrucciones como pueden ser sumas o multiplicaciones. De hecho es la parte que hace más rápidos a los DSP's en comparación con otros procesadores de propósito general.

Unidad aritmética auxiliar. Los DSP's tienen frecuentemente una unidad aritmética auxiliar, que se utiliza para realizar cálculos matemáticos y operaciones lógicas en el caso de estar ocupada la unidad central.

Puertos. Los DSP's normalmente tienen puertos de comunicación seriales y paralelos para comunicación rápida con otros procesadores y/o convertidores.

Memoria. La memoria retiene información e instrucciones para los DSP's. Los DSP's leen y/o escriben información en ella, es muy

frecuente que se encuentren bancos de ésta en el interior del chip ya que suele ser más rápida que la memoria externa.

Convertidores. Estos dispositivos proveen la función de transformación para el DSP, ya que solo trabajan con señales digitales, requieren elementos para transformar una señal analógica en digital y viceversa.

LOS PRODUCTOS Y LAS SUMAS

Hasta este punto nos podríamos preguntar: ¿porqué necesitamos un DSP?, ¿porqué no utilizar un procesador de propósito general para el tipo de tareas que estamos tratando?. La respuesta se basa en la naturaleza de las operaciones más comunes en la labor del procesamiento de señales: los productos y las sumas. Las sumas y las restas, son realizadas en pocos ciclos de reloj en la mayoría de los procesadores, pero la multiplicación y la división son mucho más complejas operaciones. Una operación de multiplicación consiste de una serie de sumas y desplazamientos. Los microprocesadores de propósito general son muy lentos para ejecutar operaciones de multiplicación y división, en donde a través de microcódigo se implementan una serie de sumas, restas y desplazamientos para una simple multiplicación. Como ejemplo, el 68000 (un microprocesador de Motorola) requiere 10 ciclos de reloj para sumar, y alrededor de 74 para multiplicar. Por la naturaleza del procesamiento digital de señales requerimos realizar múltiples cálculos de la forma:

$$A = B * C + D * E$$

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

Convolución

$$X[\Omega] = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}$$

Transformada de Fourier

Es obvio que necesitamos un hardware especializado en ejecutar sumas y multiplicaciones en un solo ciclo de reloj para el procesamiento de señales. Es por esto que la mayoría de los DSP's tienen instrucciones específicas para multiplicar, sumar y almacenar resultados en un solo ciclo. Este tipo de instrucciones son llamadas con frecuencia MAC.

Hemos enfatizado que necesitamos software especializado para desempeñar la suma y el producto en el menor tiempo posible, pero ¿cómo es posible obtener una operación de multiplicación lo más rápida posible?, sin un multiplicador lo suficientemente rápido el procesamiento de señales sería un sueño. El problema de diseñar multiplicadores rápidos empezó en los inicios de los años 70's cuando en 1971 Lincoln Laboratories diseño un multiplicador usando 10,000 circuitos integrados realizando la multiplicación en tan solo 600ns. A mitades de los años 70's la multiplicación se podía realizar en tan solo 200ns. Este hecho hizo posible diseñar procesadores de aceptable desempeño, aunque grandes y caros, pero que demostraron la viabilidad de la multiplicación rápida. A principios de los años 80's, DSP's en un solo chip aparecieron con un razonable desempeño y a medida que han transcurrido los años, el tiempo necesario para realizar una multiplicación ha disminuido mucho. En la actualidad existen dispositivos para operar con enteros que pueden multiplicar en tiempos tan cortos como menores a 5ns, dados los orígenes de esta tecnología el mejoramiento alcanzado ha sido notable. Ahora veamos más de cerca la arquitectura de los procesadores para ver como se ha afectado el diseño de los DSP's.

Los procesadores necesitan instrucciones para operar, a cada ciclo de reloj debe indicárseles lo que deben hacer, si tenemos el conjunto de instrucciones almacenadas de alguna forma, esto es más sencillo, ya que solo debe buscar y ejecutar instrucciones. De esta forma hablamos de máquinas con programa almacenado, las cuales buscan instrucciones, después datos y al ejecutar la instrucción devuelven datos para almacenar. Existen dos tipos de arquitecturas bien conocidas, la Von Neuman y la Harvard.

Las máquinas von Neuman almacenan tanto programas como datos en una misma área de memoria. En este tipo de máquinas la instrucción contiene la operación a realizar así como la dirección del dato necesario para la operación. Esto hace necesario leer primero la instrucción y posteriormente el dato. En una arquitectura Harvard la única diferencia existente es el hecho de tener programa y datos en dos memorias separadas, y que se accesan por dos vías distintas, esto permite que instrucciones y datos puedan ser cargados en forma simultánea.

La arquitectura von Neuman se utiliza en sistemas computadores de propósito general, así como la arquitectura Harvard es más común en procesadores especializados, procesamiento en tiempo real o bien aplicaciones dedicadas. La mayoría de los DSP's tienen arquitectura Harvard ya que muchas aplicaciones requieren máquinas rápidas para procesamiento en tiempo real. Por lo tanto internamente existen en los chips rutas o buses para datos y para programa en forma separada, aunque por cuestiones de economía de los circuitos, para accesos a espacios externos se tiene un solo bus multiplexado para el doble propósito. El esquema anterior se denomina "Harvard modificado". Un ejemplo de pro-

cesador que utiliza esta arquitectura es el TMS320C50 de Texas Instruments. Este chip aprovecha las características anteriores para realizar dos trabajos en forma simultánea: la búsqueda de instrucciones y datos sin interrupción uno del otro, lo cual provee un alto desempeño a bajo costo. Con lo anterior queremos decir que existe en esta arquitectura un alto grado de paralelismo.

SISTEMAS DE DESARROLLO CON DSP'S

Un DSP es un circuito integrado que no puede funcionar sin la "inteligencia" de un programa, que no es mas que un conjunto de instrucciones para realizar ciertas tareas, y para lograr lo anterior hacemos uso de algunas herramientas como son:

Ensambladores. Son programas que generan código a nivel del microprocesador usando instrucciones a nivel texto. Debido a que nosotros entendemos mejor palabras escritas que series de unos y ceros, utilizamos los ensambladores para convertir instrucciones en texto a lenguaje de máquina. Esto evita la necesidad de aprender largas y tediosas series de números binarios.

"ADD A,B" ————>
 "111000100101010001001"

Lenguajes de alto nivel. Son similares a los ensambladores pero mucho mas amigables. Los ensambladores constan de instrucciones en verdad muy básicas, como sumar, multiplicar o bien comparar. Los lenguajes de alto nivel tienen instrucciones mas complejas como imprimir o implementar bucles de repetición, por lo que es mas fácil escribir programas en lenguajes de alto nivel. Aunque los lenguajes de alto nivel permiten escribir programas en forma más fácil, los

ensambladores generan código que se ejecuta más rápido, por eso los dos son muy usados en los DSP's, Combinando las ventajas de ambos, podemos escribir el grueso de una aplicación en alto nivel, y las secciones críticas en cuanto a velocidad en ensamblador.

Simuladores. Estos programas son implementaciones del chip en software. Un simulador permite correr un programa, y simular casi toda la funcionalidad del procesador. Éstos son usados par analizar los diseños antes de implementar en hardware y son muy útiles para predecir cuando un diseño trabajará o no.

Emuladores. Un emulador es un circuito que trabaja justamente como un DSP, pero nos permite controlar y ver el resultado de la ejecución de instrucciones, siendo el modo normal de trabajo la ejecución paso a paso, lo que permite ver como se modifican los voltajes de los terminales del circuito involucrados en la ejecución de la instrucción corriente. Esta es una pieza invaluable en el desarrollo de sistemas.

El ciclo de desarrollo de una aplicación para DSP's comienza con la escritura de un programa para su posterior simulación. Si todo trabaja como es deseado se prueba el software en el hardware diseñado y se puede llegar a una primera versión del producto final. Cuando mejoras o adiciones son requeridas, se puede llagar a una versión posterior por el mismo procedimiento anterior.

¿PORQUÉ DIGITALMENTE?

Hasta este punto nos preguntáramos: ¿porqué es preferible convertir las señales de su forma analógica a digital?. Como vamos a discutir en la presente sección, el procesamiento

digital de señales ofrece claras ventajas que lo hacen ideal para cierto tipo de aplicaciones. Lo que es cierto es que la revolución digital ha cambiado y seguirá cambiando nuestras vidas. Para aplicaciones del procesamiento de señales las ventajas que se alcanzan en los resultados pueden ser:

Programabilidad. Como se puede ver, un mismo hardware puede realizar diversas tareas, por ejemplo un mismo sistema puede funcionar como un filtro, como un sintetizador de música, como control de motores, etc. La función depende del conjunto de programas que alimenten al sistema, y este hecho le confiere gran flexibilidad.

Escalabilidad. Una vez diseñado el sistema, si al encontrarse funcionando queremos modificarlo o bien agregar nuevas funciones, solo requerimos modificar el programa que alimenta a dicho sistema. Esto es deseable cuando los sistemas cambian a un nuevo ambiente o bien condiciones de trabajo. Con sistemas analógicos lo anterior requeriría el cambio de componentes o agregar nuevos elementos al diseño.

Flexibilidad. Una sola tarjeta con DSP puede desempeñar muchas funciones simplemente cargando los nuevos programas en ella.

Las ventajas presentadas anteriormente permiten reducir el tiempo de diseño y la complejidad del mismo. Con circuitos analógicos, un nuevo circuito debe ser diseñado para cada función.

Estabilidad y repetibilidad. Los componentes analógicos como resistencias, capacitores, diodos, amplificadores operacionales etc. Son afectados todos ellos por la temperatura. Su desempeño y características cambian con los cambios debidos a la temperatura. El desempeño de un

equipo no es el mismo en climas fríos que en el desierto, y para ciertas tareas esto puede ser catastrófico. Los circuitos digitales pueden operar en un amplio rango de temperaturas sin cambio aparente en su desempeño.

Envejecimiento. Como los componentes envejecen, sus características se modifican, esto afecta el desempeño y la operación de los circuitos analógicos, el fenómeno anterior hace que un circuito que hoy día funciona perfectamente, dentro de unos años no funcionará del todo.

Tolerancias. Los componentes como resistencias y capacitores tienen tolerancias, sus valores no son 100% precisos, es decir nos pueden vender un componente de cierto valor con 10% de variación. Dos circuitos analógicos que tienen un mismo diseño operaran en forma algo distinta. Este puede ser un problema grande, pues los fabricantes de equipo pueden enfrentarse con que varios de sus equipos fabricados no funcionen adecuadamente debido a este problema.

Repetibilidad. El desempeño de un circuito digital no cambia de una unidad a otra. Ya que el procesamiento de la información es en forma numérica, una multiplicación genera el mismo valor en cualquier procesador. El alimentar dos voltajes a la entrada de diversos circuitos analógicos producirá siempre distintos resultados. La tolerancia de los componentes no afecta el desempeño de los circuitos digitales, siempre producen el mismo resultado. Como los factores físicos no afectan el desempeño de los circuitos digitales, estos trabajarán igual en Asia o en Sudamérica. El nivel de precisión será determinado por el número de bits usado, y siempre será el mismo. Con circuitos analógicos la precisión de los resultados puede variar ampliamente de un cir-

cuito a otro. Después de todo, un reproductor de CD tiene la misma calidad de reproducción y solo es afectada por la mínima circuitería analógica que lo compone.

Funciones especiales. Algunas funciones especiales solo pueden implementarse digitalmente, ya que los circuitos analógicos no pueden hacerlo, o lo hacen ineficientemente. Un ejemplo de lo anterior es la compresión de la información. Los canales de comunicación son realmente caros, existe una sustancial diferencia en costo en una llamada de telefónica de dos minutos a una de 20 minutos entre la misma fuente y destino. El almacenamiento de la información también puede ser caro, por lo que debemos tener cuidado en la forma de almacenarla. La compresión sin pérdida es una respuesta a los dos problemas anteriores. Una cierta cantidad de información puede ser comprimida a la mitad de su tamaño original, por lo tanto solo ocupa la mitad del espacio que requeriría en un disco en su versión original, y segundo tardaríamos solo la mitad de tiempo para transmitirla en un canal de comunicación. Este tipo de "magia" es imposible en circuitos analógicos. Otro ejemplo es la realización de filtros de ranura, un caso especial de rechaza banda con una selectividad muy estrecha. Este tipo de filtros puede implementarse fácilmente en forma digital, ya que en forma analógica es prácticamente imposible producir resultados similares. Este tipo de filtros se usan a menudo para la cancelación de ruido debido a maquinaria u otra fuente de ruido. Además, la obtención de filtros de fase lineal (con mínima distorsión) es más sencillo en forma digital.

APLICACIONES PRÁCTICAS UTILIZANDO PROCESAMIENTO DIGITAL DE SEÑALES

Desde su introducción en el mercado, los DSP's han encontrado una amplia variedad de aplicaciones, ya que generalmente la tecnología de los DSP's no es cara actualmente.

Equipos musicales de alta fidelidad. Como hemos mencionado anteriormente el procesamiento de señales juega un papel importante en equipos de alta fidelidad, por ejemplo en sistema reproductor de CD, o bien en ecualizadores gráficos digitales, o el manejo de sistemas sonoros de muy alta calidad. El precio actual de estos equipos muestra que esta tecnología no es cara.

Juguetes. Los niños de hoy pueden jugar con muñecos parlantes, o bien al apretar un botón se reproduce el sonido una herramienta, o un animal, y por si fuera poco existen ya juguetes que utilizan el reconocimiento de voz.

Modems. Los modems cada vez son más eficientes y rápidos en su operación y su uso se ha ido extendiendo poco a poco. Los DSP's juegan un papel importante en las funciones de estos equipos ya que permiten la cancelación de ecos, o bien la compresión y descompresión de la información.

Sistemas telefónicos. La telefonía celular utiliza a los DSP's para codificar y decodificar la información, asimismo es muy frecuente encontrar máquinas contestadoras que nos indican: "para ventas presione 1", "para soporte técnico presione 2", etc. Este trabajo lo hacen los DSP's.

Procesamiento de imágenes. Los DSP's son usados ampliamente en procesamiento de imágenes, como

en simuladores tridimensionales para aumentar el realismo, estas aplicaciones pueden incluir visión para robots, compresión de la información, cálculo de detalles tridimensionales, inspección de imágenes, reconocimiento de huellas dactilares, etc.

Además de las anteriores, existen otros campos de aplicación, como en la medicina, la industria, la instrumentación etc. Como se muestra en la **tabla 1**.

LAS FAMILIAS ACTUALES DE DSP'S

Existen en la actualidad dos clasificaciones para los DSP's: de punto fijo (enteros) y de punto flotante. El rango de cantidades que se pueden expresar es muy variado. Mientras que en uno de punto fijo se pueden representar típicamente valores entre:

$\pm 2^{15} (\pm 32768)$ y $\pm 2^{-15} (\pm 0.000031)$,
 en uno de punto flotante se tienen rangos más amplios:

$\pm 2^{128}$ y $\pm 2^{-128}$

Esta diferencia es considerable, y es importante en el momento de elegir un dispositivo para una aplicación en particular. Por ejemplo, un sistema profesional de sonido empleará dispositivos de punto flotante mientras que un juguete utilizara uno de punto fijo.

El ancho del bus de datos en un procesador de punto fijo es típicamente de 16 y 24 bits, casi todos tienen arquitectura Harvard modificada, y son ampliamente usados en juguetes, discos duros, modems o bien el sistema de suspensión activa de un auto.

En los procesadores de punto flotante el ancho del bus de datos es de 32 bits típicamente, dadas las características de éstos, tienen un rango dinámico más amplio. Es precisamente esta mayor precisión la que los hace ideales para aplicaciones en sistemas de alta fidelidad, sistemas de correo de voz, procesamiento de imágenes tridimensionales etc.

Cada vez salen al mercado procesadores más potentes que aprovechan las características de la tecnología actual, esto permite incorporar varios DSP's trabajando en paralelo en el interior de un solo chip, o procesadores que trabajan a frecuencias superiores a los 200Mhz. La exigencia de las tecnologías y las aplicaciones actuales implican el conocimiento y uso intensivo de los DSP's como elemento de trabajo.

TELECOMUNICACIONES

Modem	Repetidores de línea	Encriptación de datos	Telefonía celular
Modem multicanal	Digital PABx	Cancelación de ecos	Estaciones Base
DTAD	Redes Digital	Ecuilibradores adaptivos	Paginadores
Telefonía	Telefonía por pago	Correo de voz	

CONSUMIDORES / COMPUTADORAS

TV digital	Tarjetas mezcladoras	Dictado	Discos duros
Cámara digital	Cajas de efectos	Multimedia	Máquinas de lavado
Copiadoras a color	Sintetizadores	Remotas terminal	Refrigeración
Videofono	Tarjetas de sonido PC	Fax	Impresoras
Scanners	Ecuilibradores HIFI	Juguetes	
DVD	GPS		

INDUSTRIAL

Control de Motor	Robótica	Seguridad (acceso)	Electricidad (medición)
Bombas para agua	Visión para Robots	Seguridad alarmas	Instrumentación
HVAC	Servo control	Lectores de barras	

AUTOMOTRIZ

Encendido Electrónico	Control antiderrapante	Audio Digital	Bolsa de aire
Culch Eléctrico	Control de máquinas.	Telefonía celular	GPS/Navegación
Frenos	Control adaptivo de ruta	Asientos y espejos eléctricos	Elevadores de ventanillas
Cancelación Activa	Ayuda de parqueo etc.		

Tabla 1. Campos de aplicación

CONCLUSIÓN

El uso de las nuevas tecnologías digitales permite contar con circuitería cada vez más confiable, económica y con un alto grado de eficiencia, es evidente que nuestro mundo se vuelve cada vez más digital, y por lo tanto cada vez más aplicaciones de la vida moderna se dirigen hacia esa tendencia ya sea en el hogar, en la industria, en la medicina, en la oficina, en los centros de investigación etc. Es un hecho que los tres factores que contribuyen a lo anterior son:

1. La necesidad de un mejor aprovechamiento de los recursos. (Los canales de comunicación por ejemplo)
2. La necesidad de un equipo rápido, confiable, de alta calidad y de mayor precisión.
3. La necesidad de contar en un futuro con equipo que tenga la posibilidad de crecimiento y flexibilidad.

En nuestros días, debemos explotar adecuadamente los recursos y las facilidades que nos provee la tecnología moderna a fin de mejorar y no rezagarnos en un mundo de contante cambio y actualización.

El Espacio de Configuración de Dispositivos PCI (Desarrollo de Rutinas de Acceso)

Ing. Israel Rivera Zárate.
Profesor e Investigador del CIDETEC-IPN

Este documento es continuación del artículo titulado "Diseño de una Interfaz PCI para una Tarjeta Coprocesadora Basada en el DSP TMS320C40-40", aparecido en polibits en su número 21. Debido a esto, se da por entendido que el lector debe tomar dicho artículo como referencia base para la comprensión de este documento.

INTRODUCCIÓN

Cuando el equipo de computo se energiza, el software de configuración realiza un recorrido por el bus PCI para determinar la existencia de dispositivos así como de los recursos que estos requieren para su operación. Este proceso es comunmente referido como «*proceso de descubrimiento*». El programa que realiza esta función es referido frecuentemente como el «*enumerador del bus PCI*» [1].

Con el objetivo de facilitar este proceso, todos los dispositivos PCI deben implementar un conjunto básico de registros de configuración definidos por la especificación PCI. Dependiendo de sus características operativas, un dispositivo puede imple-

mentar registros opcionales de configuración, definidos también por la especificación.

El software de configuración efectúa una serie de accesos a registros específicos para establecer la presencia y tipo de los dispositivos instalados en el bus PCI. Una vez determinada la presencia del dispositivo, el software accesa otros registros de configuración para establecer la cantidad de bloques de memoria y puertos de E/S que éste requiere para su operación. Entonces programa los decodificadores de direcciones de memoria y puertos de E/S implementados en el dispositivo para que responda en futuros accesos a estos rangos. Los rangos entre dispositivos son garantizados a ser mutuamente excluyentes.

Si el dispositivo manifiesta la necesidad de uso de una interrupción PCI (vía uno de sus registros de configuración), el software de configuración lo programa con la información de ruteo correspondiente indicándole cuál línea solicitud de interrupción del sistema (IRQ) le es asignada.

Si el dispositivo tiene capacidad de volverse maestro del bus, «*bus mastering*», el software de configuración puede acceder dos registros de configuración que permiten establecer la frecuencia con que solicitará la posesión del bus PCI y la duración promedio de la transferencia. El software de configuración emplea esta

información para programar los registros de temporización de latencia del bus PCI implementados en el árbitro de bus PCI normalmente integrado en el ChipSet PCI y de esta manera garantizar un uso óptimo del bus.

TRES ESPACIOS DE DIRECCIONES : PUERTOS DE E/S, MEMORIA Y CONFIGURACIÓN

Los procesadores Intel (sólo por referirnos a la plataforma de PC), poseen la habilidad para direccionar dos espacios distintos: puertos de E/S y memoria. Los dispositivos PCI con capacidad de ser maestro del bus (incluyendo el mismo ChipSet PCI y los puentes PCI), desarrollan ciclos de acceso PCI de E/S y memoria para acceder las regiones de E/S y memoria de dispositivos PCI respectivamente. Además un tercer tipo de acceso, «*el ciclo de configuración*», es utilizado para acceder los registros de configuración.

Los ciclos de configuración se originan como un acceso ordinario del procesador anfitrión a las direcciones de puertos: CF8h y CFCh respectivamente. Estas direcciones corresponden con registros de operación del controlador de bus del sistema implementado en el ChipSet PCI. El registro ubicado en CF8h es frecuentemente denominado CONFADDRESS

y traduce el dato enviado en una dirección dentro del espacio de configuración. El registro ubicado en CFCh es frecuentemente denominado como CONFDATA y almacena el dato a transferir a la región de configuración definida por CONFADDRESS. Rutinas dispuestas en el BIOS PCI agilizan este procedimiento permitiendo el acceso libre y transparente [2].

El espacio de configuración PCI es dividido y repartido entre cada dispositivo funcional contenido dentro de un único dispositivo físico (circuito integrado). Las primeras 16 doublewords (1 doubleword=32 bits), son referidas como el encabezado de configuración, «*configuration header*». El formato y uso de esta área es definida por la especificación PCI. Actualmente existen dos tipos básicos de encabezados de configuración: encabezado tipo 0 para todos los dispositivos, diferentes a los puen-

tes Host/PCI, PCI/PCI, o PCI/ISA, los cuales emplean el tipo 1.

ENCABEZADO DE CONFIGURACIÓN TIPO 0

Como se mencionó en el párrafo anterior, el encabezado tipo 1 es más bien asignado a los dispositivos de soporte del sistema de cómputo. Tomando como base que nuestro interés se centra en el desarrollo de aplicaciones orientadas al bus PCI, parece más conveniente describir la arquitectura del encabezado de configuración tipo 0. Ver **figura 1**.

La descripción formal respecto al significado y uso de los registros descritos en el encabezado de configuración escapa a los propósitos del artículo por lo que tan solo se indican brevemente en la **tabla 1**.

PLUG AND PLAY

Los dispositivos PCI forman parte de la arquitectura de la especificación Plug and Play (también llamada PnP). PnP fue desarrollada por Microsoft con la cooperación de Intel y muchos otros fabricantes de hardware. Como su nombre sugiere, la meta de Plug and Play es crear una máquina cuyo hardware y software trabajen conjuntamente para configurar dispositivos y asignar recursos automáticamente, de tal forma que al instalar un nuevo dispositivo pueda utilizarse inmediatamente sin mayor necesidad de configuración.

La detección automática y configuración de dispositivos no es una tarea fácil, para realizarse, requiere la cooperación de diferentes componentes hardware y software: hardware del sistema, hardware de periféricos, BIOS del sistema y sistema operativo [3].

Hardware del Sistema. El hardware del sistema a través del Chipset del sistema y los controladores del bus del sistema deben ser capaces de manejar dispositivos PnP. Los sistemas modernos basados en PCI han sido diseñados con PnP en mente.

Hardware de Periféricos. Los dispositivos que se añaden al sistema deben ser compatibles con PnP.

BIOS. El BIOS del sistema juega un papel clave, ya que las rutinas contenidas en él realizan la tarea de recolectar la información de los diferentes dispositivos instalados, así como los recursos que requieren. El BIOS comunica esta información al sistema operativo, que la utiliza para configurar sus drivers y otros programas para permitir que los dispositivos trabajen correctamente.

31		24 23		16 15		8 7		0		
DEVICE ID				VENDOR ID				00		
STATUS				COMMAND				04		
CLASS CODE						REV ID		08		
BIST		HEADER TYPE		LATENCY TIMER		CACHE L Z		0C		
BASE ADDRESS REGISTER 0										10
BASE ADDRESS REGISTER 1										14
BASE ADDRESS REGISTER 2										18
BASE ADDRESS REGISTER 3										1C
BASE ADDRESS REGISTER 4										20
BASE ADDRESS REGISTER 5										24
RESERVED = 0										28
RESERVED = 0										2C
EXPANSION ROM BASE ADDRESS										30
RESERVED = 0										34
RESERVED = 0										38
MAX_LAT		MIN_GNT		INTERRUPT PIN		INTERRUPT LINE		3C		

Figura 1. Encabezado tipo 0 del espacio de configuración de dispositivos PCI.

Desplazamiento	Abreviación	Descripción
00h-01h	VID	Identificación del fabricante del dispositivo.
02h-03h	DID	Identificación del dispositivo.
04h-05h	PCICMD	Comando de acceso PCI (acceso de lectura o escritura).
06h-07h	PCISTS	Información del estado del acceso.
08h	RID	Identificación del número de revisión del fabricante.
09h-0Bh	CLCD	Tipo de dispositivo: Clase base, Subclase, etc.
0Ch	CALN	Tamaño de la línea de cache en doublewords.
0Dh	LAT	Tiempo promedio en el que el dispositivo será maestro del bus.
0Eh	HDR	Tipo de header =0.
0Fh	BIST	Auto Prueba («Build-In Self Test»). Permite la implementación de rutinas de diagnóstico del dispositivo
10h-27h	BADR0-BADR5	Registros de dirección base. En estos registros se definen las regiones de memoria o puertos de E/S y sus tamaños.
28h-2Fh	—	Reservado.
30h	EXROM	Provee un mecanismo para la asignación de espacio en memoria física de una ROM de expansión
34h-3Bh	—	Reservado.
3Ch	INTLN	Línea IRQ del sistema asignada al dispositivo.
3Dh	INTPIN	Identifica la interrupción PCI del dispositivo (/INTA, /INTB, /INTC o /INTD).
3Eh	MINGNT	Duración promedio de un burst en modo maestro.
3Fh	MAXLAT	Frecuencia promedio de uso del bus PCI en modo maestro.

Tabla 1. Descripción de los registros del encabezado de configuración tipo 0.

Sistema Operativo. Finalmente, el sistema operativo debe estar diseñado para trabajar con el BIOS (y por lo tanto indirectamente con el hardware). El sistema operativo da de alta el software de bajo nivel (tal como device drivers) que sea necesario para el dispositivo que ha de ser utilizado por las aplicaciones. Además se comunica con el usuario, notificando los cambios en la configuración, y permite modificar la asignación de recursos si fuese necesario.

La mayor parte del trabajo que permite a Plug and Play operar correctamente la realiza el BIOS del sistema durante el proceso de arranque (boot). Si el BIOS tuviera que asignar recursos a cada dispositivo PnP en cada arranque del sistema, resultarían dos problemas. El primero, invertiría tiempo en hacer algo que acaba de realizar anteriormente, en cada arranque, sin ningún propósito. Después de todo, la mayoría de la gente no cambia el hardware del sistema constantemente. Segundo, y más importante, es posible que el BIOS no realice siempre la misma decisión cuando decide asignar recur-

sos, pudiendo realizar modificaciones incluso cuando el hardware no ha sido cambiado.

Por ello se diseñó el ESCD (Extended System Configuration Data), que es un área especial que forma parte de la memoria CMOS donde son resguardados los parámetros del BIOS. Esta área de memoria es utilizada para mantener información de configuración del hardware del sistema. Al momento del arranque el BIOS verifica esta área de memoria y si no detecta cambio alguno respecto al último arranque, asume que no requiere configurar nada y brinca esa porción del proceso de arranque.

El ESCD es utilizado también como un enlace de comunicación entre el BIOS y el sistema operativo. Ambos utilizan el área ESCD para leer el estado actual del hardware y grabar los cambios. Windows 95 lee el ESCD para determinar si hay algún cambio en el hardware y reaccionar correspondientemente. Windows 95 permite a los usuarios modificar las asignaciones de recursos realizadas por PnP, manualmente a través del Admi-

nistrador de dispositivos. Esta información es grabada en el área ESCD, así que el BIOS conocerá los cambios realizados en el arranque siguiente y no intentará modificar las asignaciones.

PROPÓSITO DEL BIOS PCI

El sistema operativo (excepto por el micro kernel específico de la plataforma), programas de aplicación y device drivers no son capaces de acceder directamente los registros de configuración de dispositivos PCI. Los métodos hardware utilizados para implementar esta capacidad son específicos de la plataforma. Cualquier software que accese directamente estos mecanismos es por lo tanto, por definición, específico de la plataforma. Esto conduce a problemas de compatibilidad, esto es, el software trabaja en algunas plataformas pero no en otras.

En vez de esto, las solicitudes de comunicación deben dirigirse al BIOS PCI. El BIOS es específico de la pla-

taforma. Es implementado en firmware y posiblemente en la capa de abstracción del hardware HAL (Hardware Abstarcción Layer) del sistema operativo [2]. El BIOS provee los siguientes servicios:

- Permite la determinación de los mecanismos de configuración soportados por el chipset PCI.
- Permite la determinación del rango de los buses PCI presentes en el sistema.
- Busca todas las instancias de un dispositivo PCI específico o un dispositivo correspondiente a una clase específica.
- Permite la lectura y escritura de los registros de configuración de dispositivos PCI

SOPORTE DE ENTORNOS DE SISTEMAS OPERATIVOS

Diferentes sistemas operativos poseen diferentes características operacionales (tales como el método que define el uso de la memoria del siste-

ma, el método utilizado para invocar los servicios del BIOS). En sistemas basados en la familia de procesadores x86, el sistema operativo que se ejecuta en una plataforma particular cae en una de las siguientes categorías:

- Sistema operativo en modo real (en otras palabras MS-DOS).
- Modo protegido 286.
- Modo protegido 386 (modelo segmentado y modelo plano).

MODO REAL

Los sistemas operativos en modo real, tal como MS-DOS, están escritos para ser ejecutados por el procesador 8088. Ese procesador es capaz de direccionar únicamente 1MB de la memoria del sistema (00000h a FFFFFh). MS-DOS realiza llamadas al BIOS PCI mediante la ejecución de la interrupción por software 1Ah. Para especificar un servicio particular deben cargarse previamente parámetros en los registros de segmento del procesador. Ver **tabla 2**.

RUTINAS DE ACCESO AL ESPACIO DE CONFIGURACIÓN

Tomando como base los servicios a interrupción para acceso al espacio de configuración dispuestos en el BIOS PCI se desarrollaron un conjunto de rutinas básicas fundamentadas en [4] y [5]; que permiten el acceso a los registros de configuración de cualquier dispositivo instalado en el bus PCI. Estas rutinas se presentan en las columnas siguientes:

FRAGMENTO DE CÓDIGO EJEMPLO

Contando con las rutinas básicas de acceso a registros de configuración se desarrolló un programa de prueba en el que se realizan únicamente operaciones de lectura a todos los registros de configuración de acuerdo con la arquitectura de encabezado tipo 0 mostrada en la **figura 1**.

FUNCIÓN	Valor en AH	Valor en AL
Identificador de funciones PCI	B1h	
Prueba la existencia del BIOS PCI		01h
Encuentra un dispositivo PCI		02h
Encuentra un dispositivo PCI según su clase de código		03h
Genera ciclo especial		06h
Lee un byte de un registro de configuración		08h
Lee una word de un registro de configuración		09h
Lee una doubleword de un registro de configuración		0Ah
Escribe un byte a un registro de configuración		0Bh
Escribe una word a un registro de configuración		0Ch
Escribe una doubleword a un registro de configuración		0Dh
Obtiene las opciones de ruteo de interrupciones PCI		0Eh
Asigna una interrupción PCI		0Fh

Tabla 2. Servicios del BIOS PCI.

El Espacio de Configuración de Dispositivos PCI (Desarrollo de Rutinas de Acceso)

```

;*****
; llamada a funciones del BIOS PCI para leer o escribir
; a registros de configuración
;*****
;
; READ_C_BYTE
; Lee un registro de configuración de 8 bits
; DI: número de registro
; CL: byte leído
; Bandera Carry: 1=error, 0=exitoso.
;
read_c_byte proc near
    push bx ;salva valores de registros
    push ax ;
    mov bh, BUS_NUM ;carga número de bus
    mov bl, DEV_FUNC ;carga número de dispositivo
    ; y numero de función
    mov ah, PCI_FUNC_ID ;carga identificador de función PCI
    mov al, READ_CONFIG_BYTE ;carga servicio de lectura de byte
    int 1ah ;interrupción del bios
    pop ax ;
    pop bx ;restablece valor a registros
    jnc good ;si tuvo exito regresa a la
    ;rutina principal
good: ret ;regreso a la rutina principal
read_c_byte endp
;
; READ_C_WORD
;
; Lee un registro de configuración de 16 bits
; DI: número de registro
; CX: word leída
; Bandera Carry: 1=error, 0=exitoso.
;
read_c_word proc near
    push bx ;salva registros
    push ax ;
    mov bh, BUS_NUM ;
    mov bl, DEV_FUNC ;carga número de dispositivo
    ; y número de función
    mov ah, PCI_FUNC_ID ;carga identificador de función PCI
    mov al, READ_CONFIG_WORD ;carga servicio de lectura de word
    int 1ah ;interrupción del bios
    pop ax ;restablece valores a registros
    pop bx ;
    ret ;
read_c_word endp
;
; READ_C_DWORD
;
; Lee un registro de configuración de 32 bits
; DI: número de registro
; ECX: byte leído
; Bandera Carry: 1=error, 0=exitoso.
;
read_c_dword proc near
    push bx ;salva registros
    push ax ;
    mov bh, BUS_NUM ;
    mov bl, DEV_FUNC ;carga número de dispositivo y
    ; número de función
    mov ah, PCI_FUNC_ID ;carga identificador de función PCI
    mov al, READ_CONFIG_DWORD ;carga servicio lee double word
    int 1ah ;interrupción del bios
    pop ax ;restablece valores a registros
    pop bx ;
    ret ;
read_c_dword endp
;
; WRITE_C_BYTE
;
; Escribe un dato de 8 bits a un registro de configuración
; DI: número de registro
; CL: valor a escribir

```

```

; Bandera Carry: 1=error, 0=exitoso.
;
write_c_byte proc near
    push ax ;salva valor de registros
    push bx ;
    mov ah, PCI_FUNC_ID ;carga identificador de función PCI
    mov al, WRITE_CONFIG_BYTE ;servicio de escritura de byte
    mov bh, BUS_NUM ;carga número de bus
    mov bl, DEV_FUNC ;carga número de función y
    ; dispositivo
    int 1ah ;interrupción del bios
    pop bx ;restablece registros
    pop ax ;
    ret ;
write_c_byte endp
;
; WRITE_C_WORD
;
; Escribe un dato de 16 bits a un registro de configuration
; DI: número de registro
; CX: valor a escribir
; Bandera Carry: 1=error, 0=exitoso.
;
write_c_word proc near
    push bx ;save registers
    push ax ;
    mov ah, PCI_FUNC_ID ;load function id
    mov al, WRITE_CONFIG_WORD ;load write function value
    mov bh, BUS_NUM ;load bus number
    mov bl, DEV_FUNC ;load device and function number
    int 1ah ;call bios
    pop ax ;
    pop bx ;restore register values
    ret ;
write_c_word endp
;
; WRITE_C_DWORD
;
; Escribe un dato de 32 bits a un registro de configuración
; DI: número de registro
; ECX: valor a escribir
; Bandera Carry: 1=error, 0=exitoso.
;
write_c_dword proc near
    push ax ;salva valor de registros
    push bx ;
    mov ah, PCI_FUNC_ID ;es una función PCI
    mov al, WRITE_CONFIG_DWORD ;servicio de función PCI
    mov bh, BUS_NUM ;numero de bus PCI
    mov bl, DEV_FUNC ;número de dispositivo y función
    int 1ah ;interrupción del BIOS
    pop bx ;restablece valor de registros
    pop ax ;
    ret ;
write_c_dword endp
;

```


El Espacio de Configuración de Dispositivos PCI (Desarrollo de Rutinas de Acceso)

```

rvid: movdi, VID_ADDR      ;carga dirección de VID
      jmp word_reg        ;
rvid: movdi, DID_ADDR      ;carga dirección de DID
      jmp word_reg        ;
rcmd: movdi, CMD_ADDR      ;carga dirección de COMMAND
      jmp word_reg        ;
rst: movdi, STS_ADDR       ;carga dirección de STATUS
      jmp word_reg        ;
rrid: movdi, RID_ADDR      ;carga dirección de REVISION ID
      jmp byte_reg        ;
rclcd:
movdi, 0008h              ;utiliza una localidad anterior a CLCD para
                          ;realizar lectura de 32 bits.
      jmp b3_reg          ;
rcaln:
movdi, CALN_ADDR          ;carga dirección de CALN
      jmp byte_reg        ;
rlat: movdi, LAT_ADDR      ;carga dirección de LATENCY
      jmp byte_reg        ;
rhdr: movdi, HDR_ADDR      ;carga dirección de HEADER
      jmp byte_reg        ;
rbist:
movdi, BIST_ADDR          ;carga dirección de BIST
      jmp byte_reg        ;
rbadr0:
movdi, BADR0_ADDR         ;carga dirección de BADR0
      jmp dword_reg       ;
rbadr1:
movdi, BADR1_ADDR         ;carga dirección de BADR1
      jmp dword_reg       ;
rbadr2:
movdi, BADR2_ADDR         ;carga dirección de BADR2
      jmp dword_reg       ;
rbadr3:
movdi, BADR3_ADDR         ;carga dirección de BADR3
      jmp dword_reg       ;
rbadr4:
movdi, BADR4_ADDR         ;carga dirección de BADR4
      jmp dword_reg       ;
rbadr5:
movdi, BADR5_ADDR         ;carga dirección de BADR5

```

```

      jmp dword_reg        ;
rexrom:
movdi, EXROM_ADDR         ;carga dirección de EXROM
      jmp dword_reg        ;
rintln:
movdi, INTLN_ADDR         ;carga dirección de INTLN
      jmp byte_reg         ;
rintpin:
movdi, INTPIN_ADDR        ;carga dirección de INTPIN
      jmp byte_reg         ;
rming:
movdi, MINGNT_ADDR        ;carga dirección de MINGNT
      jmp byte_reg         ;
rmaxl:
movdi, MAXLAT_ADDR        ;carga dirección de MAXLAT
      jmp byte_reg         ;
byte_reg:
      call read_c_byte     ;lee registro de 8 bits
      mov BYTE_VAL, cl     ;valor a imprimir en pantalla
      call print_byte      ;lo presenta en pantalla
      jmp gb_cread         ;regresa de la rutina
word_reg:
      call read_c_word     ;lee el registro
      mov WORD_VAL, cx     ;valor listo a presentar en pantalla
      call print_word      ;lo presenta en pantalla
      jmp gb_cread         ;regreso
b3_reg:
      call read_c_dword    ;operación de 24 bits (CLCD)
      mov DWORD_VAL, ecx   ;lee 24 bits como un registro de 32 bits
      call print_3bytes    ;listo a presentar en pantalla
      jmp gb_cread         ;imprime en pantalla
dword_reg:
      call read_c_dword    ;lee registro
      mov DWORD_VAL, ecx   ;listo a presentar en pantalla
      call print_dword     ;imprime en pantalla
      jmp gb_cread         ;regreso
gb_cread:
      ret
do_c_read_write endp
;
-
```

