

58  
2 ej.



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

CAMPUS ARAGON

**CONTROL DE ENERGIA EN HABITACIONES  
MEDIANTE LOGICA DIFUSA**

**T E S I S**

QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
P R E S E N T A  
A A R O N O R T I Z O R T E G A

ASESOR INTERNO: ING. BENITO ZURIGA VILLEGAS (U.N.A.M.)

ASESOR EXTERNO: ING. DANIEL VELEZ DIAZ (I.N.I.N.)

JULIO DE 1998

TESIS CON  
FALLA DE ORIGEN

265336



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## Dedicatorias

- **A mi madre:**

La conclusión de este trabajo debes verlo no como un triunfo mio, sino como un tributo a la persona que más me quiere y que me sacó adelante, dándome primero la vida, un ejemplo intachable de rectitud, honestidad y tenacidad, así como amor y apoyo incondicional. Deseo que sepas también que alabo y valoro el papel que has desempeñado como padre y madre y que por eso no soy sólo yo quien se titula, a mi me darán el título pero ambos sabemos que nos pertenece a los dos por igual. Gracias por tu apoyo durante todo este tiempo.

- **A mis hermanas, mi cuñado y mis sobrinos:**

Por todo el amor y apoyo recibidos por cada uno de ustedes, y por ser parte importante de mi vida, les comparto la satisfacción de este éxito. Muchas gracias por ser mi familia.

- **A mi Pastor Miguel Duarte Ortega:**

A usted que me ha visto crecer y formarme tomando muchas veces es lugar de mi padre, le agradezco todas sus oraciones y el apoyo recibido por usted y su familia. Infinitas gracias.

- **A todos mis amigos y compañeros:**

En cada etapa de mi vida he tenido la fortuna de rodearme de personas con las que he compartido experiencias y momentos inolvidables, a todas esas personas les reitero mi más sincera amistad y afecto. Gracias por su amistad.

ATTE. Aarón Ortiz Ortega

## Agradecimientos

- **A mis Profesores y a la Universidad:**

Son muchos los buenos maestros que me han instruido durante mi formación, y nombrarlos sería muy difícil, pero a todos ellos les agradezco muy profundamente por haberme ayudado a llegar a la meta. Agradezco también a la Universidad Nacional Autónoma de México por haberme brindado la oportunidad de estudiar en sus aulas y ser uno más de sus hijos.

- **Al Instituto Nacional de Investigaciones Nucleares:**

Agradezco merecidamente al I.N.I.N. por la ayuda que me brindó para la elaboración de este trabajo de tesis, por el apoyo económico y el acceso a sus instalaciones así también por haberme permitido conocer gente valiosa a la que recuerdo con estimación.

- **Al Ingeniero Benito Zuñiga Villegas:**

A usted que fue uno de mis profesores quiero agradecerle doblemente por haber también aceptado colaborar en la asesoría de esta tesis.

- **Al Ingeniero Daniel Vélez Díaz:**

Gracias por haberme aceptado como becario y por la asesoría y la guía que me proporcionaste durante mi estancia en el instituto, espero te satisfaga como a mi la culminación de este trabajo.

- **Al personal del Laboratorio de Automatización del I.N.I.N. :**

Fue para mi muy grato el haber conocido en este departamento a personas como ustedes, les agradezco a todos el apoyo que en su momento cada quien me brindó y en especial al Físico Eduardo Sáenz y al Ing. Tonatiuh Rivero por su paciencia y sus enseñanzas.

# Contenido

---

<b>Introducción</b>	<b>1</b>
<b>1. Sensores y controladores para Temperatura e Iluminación</b>	<b>3</b>
1.1. Medición de temperatura	4
1.1.1. Termómetro con Termistor	4
1.1.2. Termómetro con RTD	5
1.1.3. Termómetro con Semi-conductores	6
1.1.4. Termómetro con Termopares	7
1.2. Medición de iluminación	9
1.2.1. Celda fotovoltaica	9
1.2.2. Dispositivos fotoconductores	10
1.2.3. Fotodiodo	11
1.2.4. Fototransistor	12
1.2.5. Circuitos de medición para iluminación	13
1.3. Controladores	14
1.3.1. Controladores electromecánicos de temperatura	14
1.3.2. Controles electrónicos	16
1.3.3. Controles graduados	17
<b>2. Conjuntos Difusos: conceptos y operaciones básicas</b>	<b>20</b>
2.1. Conjuntos exactos y conjuntos difusos	20
2.2. Conceptos básicos	22
2.3. Operaciones con conjuntos difusos	25
2.3.1. Complemento difuso	26
2.3.2. Unión difusa	27
2.3.3. Intersección difusa	28
2.3.4. Suma inclusiva	29
2.3.5. Resta inclusiva	30
2.3.6. Suma algebraica	30
2.3.7. Producto algebraico	30
2.4. Lógica bi-valuada y n-valuada	30
<b>3. Técnicas de control difuso y aplicaciones</b>	<b>34</b>
3.1. Proceso de escalamiento o normalización	36

3.2. Difusificación	37
3.2.1. Forma de las funciones de membresía	38
3.2.2. Parámetros importantes de las funciones de membresía	39
3.2.3. Número y posición de las funciones de membresía	40
3.2.4. Modificadores (Hedges)	40
3.3. Base de reglas	42
3.3.1. Reglas tipo Mamdani	43
3.3.2. Reglas tipo Sugeno-Takagi	44
3.4. Inferencia difusa	44
3.4.1. Inferencia Max-Min	46
3.4.2. Inferencia Max-Prod	49
3.4.3. Evaluando múltiples reglas difusas	50
3.5. Desdifusificación	51
3.5.1. Desdifusificación por centro de gravedad	51
3.5.2. Desdifusificación por centro de sumas	53
3.5.3. Desdifusificación por el centro del área mayor	53
3.5.4. Desdifusificación por el inicio del máximo	54
3.5.5. Desdifusificación por altura	54
3.5.6. Comparación y evaluación de los métodos de desdifusificación	55
3.6. Controladores difusos equivalentes	56
3.6.1. Equivalente difuso P	56
3.6.2. Equivalente difuso PD	57
3.6.3. Equivalente difuso PI	57
3.6.4. Equivalente difuso PID	57
3.7. Diseñando un controlador difuso	59
3.8. Aplicaciones	66
3.8.1. Mejorando la respuesta de sensores de temperatura con técnicas de control difuso	67
3.8.2. Control difuso de un horno de cemento	69
3.8.3. Control difuso de un sistema de operación de trenes	71
<b>4. Selección y descripción del microcontrolador</b>	<b>73</b>
4.1. Características generales del microcontrolador MC68HC11A1	74
4.1.1. Descripción de los pines	74
4.1.2. CPU y registros	78
4.1.3. Tipos de direccionamiento	80
4.1.4. Modos de operación	80
4.1.5. Mapa de memoria	81
4.2. Memoria EEPROM	85
4.3. Interface serial de comunicaciones (SCI)	88
4.3.1. Transmisión	88
4.3.2. Recepción	88
4.3.3. Registros de datos de comunicación serial	89

4.4. Sistema de temporización	92
4.4.1. Captura de entradas	92
4.4.2. Comparadores de salida	93
4.4.3. Interrupción de tiempo real	96
4.4.4. Acumulador de pulsos	96
4.5. Convertidor Analógico-Digital	97
<b>5. Diseño del algoritmo difuso de control</b>	<b>100</b>
5.1. Creación de las funciones difusas	100
5.1.1. Selección de la forma de las funciones de membresía	100
5.1.2. Selección de los universos de discurso y los valores de las variables difusas	101
5.2. Algoritmo de difusificación	106
5.3. Creación de la base de conocimientos	110
5.4. Algoritmo de evaluación de reglas por el método de inferencia Max-Min	113
5.5. Algoritmo de desdifusificación	115
<b>6. Implementación y pruebas</b>	<b>118</b>
6.1. Construcción del sistema mínimo	119
6.2. Diseño de un circuito de detección de cruce por cero	131
6.3. Diseño del circuito de potencia	131
6.4. Acondicionamiento de las señales de los sensores	132
6.5. Diseño del programa para el controlador difuso	133
<b>7. Resultados y conclusiones</b>	<b>143</b>
7.1. Resultados	143
7.1.1. Resultados de las pruebas al controlador difuso básico	143
7.1.2. Resultados de las pruebas físicas hechas al controlador a nivel de base de datos	151
7.1.3. Pruebas al control de iluminación	162
7.1.4. Pruebas al control de temperatura	165
7.2. Conclusiones	168
<b>Anexo 1</b>	<b>170</b>
<b>Anexo 2</b>	<b>184</b>
<b>Bibliografía</b>	<b>188</b>

## Introducción

---

A mediados de la década de los 60's, Lotfi A. Zadeh de la Universidad de Berkley en California, da a conocer los resultados de sus trabajos de investigación sobre lo que desde entonces llamó Teoría de los Conjuntos Difusos. El propósito de Zadeh era encontrar la manera de representar y manipular matemáticamente todo aquello que presentara grados de imprecisión o ambigüedad. En la teoría clásica de conjuntos (a la cual por distinción se le denomina exacta), un elemento 'a1' se dice que tiene un grado de pertenencia bien definido con respecto a un conjunto A, en estos términos podemos asegurar que una proposición tal como 'a1' pertenece al conjunto A puede resultar únicamente verdadera o falsa. Al igual que la teoría de conjuntos, las demás ramas de las matemáticas exactas trabajan con conceptos y valores bien definidos. Regresando al ejemplo arriba mencionado, si la proposición fuera 'a1' pertenece poco al conjunto A, entonces tal enunciado no podría ser valorado de manera exacta debido al grado de imprecisión que introduce el adverbio 'poco'.

La Lógica Difusa por su parte, no tiene problemas para manipular este tipo de situaciones, debido a que por naturaleza se trata de una teoría no bi-valuada sino n-valuada, es decir, ahora un elemento puede tener más de un grado de pertenencia a un conjunto determinado, así podemos asignar valores tales como poco perteneciente, muy perteneciente, mucho muy perteneciente, etc. sin importar el grado de ambigüedad que se maneje en cada caso. Al igual que la teoría exacta de conjuntos, la difusa posee una cierta cantidad de operaciones básicas que permiten la manipulación de cantidades imprecisas, como podrían ser: "muy caro", "poco caliente", "casi bajo", etc. las cuales son representadas numéricamente mediante valores obtenidos de algún intervalo que por lo general es  $[0,1]$ . Esto es, pueden ser asignados un número infinito de valores de verdad o de pertenencia, por ejemplo 0.1, 0.55, 0.799, etc. Las operaciones difusas entonces trabajan manipulando este tipo de cantidades, que sin embargo son representaciones de conceptos difusos.

En esencia, podemos decir que los valores con los que trabaja la matemática difusa son proporcionados en forma de palabras o términos lingüísticos. Mientras que ciencias exactas como son las matemáticas tradicionales, la física y la química trabajan con modelos matemáticos exactos, la lógica difusa emplea descripciones lingüísticas para construir sus modelos. Esto es muy importante, ya que en nuestro entorno existen infinidad de fenómenos que son muy difíciles sino imposibles de representar matemáticamente, debido a la incertidumbre intrínseca que los gobierna, pero que sin embargo es posible hacer una descripción verbal de ellos. Quizá por esta razón la primera aplicación de la lógica difusa se da en el campo del control, materia en la que es muy frecuente encontrar sistemas realmente complejos, cuyos elementos no siguen patrones definidos y que por el contrario presentan estados ambigüos.

Una razón que justifica el concepto de control difuso, lo proporciona el mismo Zadeh al observar que mientras más complejo es un sistema, la necesidad de describirlo con precisión se hace menos importante, sin mencionar que en algunos casos la precisión no es tan necesaria. De hecho la premisa de Zadeh es que es posible describir la conducta de un sistema y sus leyes de control lingüísticamente. La precisión es cara, especialmente cuando se utiliza para resolver problemas, y conseguir una simple aproximación frecuentemente es más que suficiente. La lógica difusa explota este "más que suficiente" a través de la imprecisión, a la cual usa de la misma manera que lo hace

el ser humano para resolver problemas. Por ejemplo, si bien es posible controlar la temperatura de un cuarto con una precisión de centésimas de grado, mediante un controlador difuso puede controlarse con una precisión de 1 o ½ grado sin que ello represente incomodidad para las personas dentro de la habitación, éste es un claro caso en donde un alto grado de precisión más bien resulta exorbitante. La lógica difusa es una excelente opción cuando no se requiere un alto grado de precisión para controlar el sistema, y sobre todo cuando mediante una buena aproximación se obtienen resultados que no rebasen límites de tolerancia.

Por otra parte, el control difuso también es conocido como control basado en el conocimiento, ya que como se mencionó en párrafos anteriores no se requiere un modelo matemático sino una descripción verbal del sistema. La lógica difusa se vale de las reglas del tipo SI .... ENTONCES para representar artificialmente los modelos lingüísticos obtenidos a través de experiencias y conocimiento humano, y al mismo tiempo para producir la salida de control del sistema mediante la relación que guarda la información contenida en los antecedentes (parte SI), con la conclusión (parte ENTONCES) . Por ejemplo, pensemos en la variable "Altura" que podría tener los siguientes valores: "alto", "medio", "pequeño", y adicionalmente consideremos otra variable denominada "Cantidad de tela", con valores como: "poca", "regular", "mucho". Con esta información hipotéticamente podríamos construir por ejemplo la regla SI Altura es alto ENTONCES Cantidad de tela es mucha. Como se puede ver esta regla almacena la información que indica que cuando se trata de una persona alta debe utilizarse mucha cantidad de tela. Esta aseveración puede tener su origen en el conocimiento del experto de producción que contribuyó en la generación del conjunto de reglas, al que se le denomina "base de conocimientos".

En el capítulo 3 titulado "Técnicas de control difuso y aplicaciones" se describe de manera más detallada cada uno de los procesos que intervienen en el controlador, y allí se presenta un método para la construcción de la base de conocimientos que facilita su representación y manejo en sistemas basados en microprocesadores.

El algoritmo desarrollado en el capítulo 5 cubre uno de los objetivos principales de este proyecto, que es precisamente el desarrollo de un algoritmo genérico de control difuso, de tal forma que pueda ser adaptado para diferentes aplicaciones. Tal algoritmo puede entonces ser codificado para algún lenguaje de computadora o como en nuestro caso para un sistema basado en microcontrolador.

Debido a que el proyecto fue desarrollado totalmente en el departamento de automatización del Instituto Nacional de Investigaciones Nucleares (I.N.I.N.), los requerimientos exigían un controlador difuso de temperatura e iluminación diseñado para trabajar en una arquitectura basada en el microcontrolador MC68HC11A1, motivo por el cual el capítulo 4 se dedica por completo a este circuito integrado, presentando un resumen de su funcionamiento, principales características y subsistemas que lo forman.

Finalmente en los capítulos 6 y 7 se presenta un reporte de la implementación y otro de resultados respectivamente. Para este último fueron obtenidas una serie de gráficas que muestran el comportamiento de las variables de control, observándose claramente el efecto que sobre estas tienen las acciones del controlador difuso.

## CAPITULO 1:

### ***Controladores y sensores para Iluminación y Temperatura***

---

La medición y el control son dos tareas que se encuentran completamente relacionadas. Concretamente, un controlador actúa sobre un sistema en base al conocimiento o a la información que tiene sobre las variables de entrada, información que es producto de una etapa de medición. En base a esto, es sencillo observar la importancia que tiene la correcta selección del elemento sensor. La precisión requerida en el medidor por ejemplo, es un parámetro a considerar al momento de hacer la selección, al igual que la sensibilidad, estabilidad, linealidad y costo. Una buena combinación de éstos parámetros asegura la obtención de un diseño óptimo. La aplicación en sí también determina el tipo de transductor necesario, en un ambiente hostil de trabajo por ejemplo, tanto gases como líquidos pueden causar daño físico al dispositivo, el diseñador entonces debe conocer los tipos de encapsulados disponibles en el mercado, de tal forma que las lecturas obtenidas del sensor sean lo más confiables posible.

La velocidad de respuesta en muchas ocasiones es un detalle a contemplar, ya que existen transductores con una mejor característica de velocidad que otros, aunque en éstos por lo general se sacrifica precisión, y nuevamente, conocer cuanto se puede ceder en precisión para obtener lecturas más rápidas, es tarea del diseñador. En un sistema de control, esta situación se ve más acentuada, debido a la necesidad de obtener una acción muchas veces inmediata ante los cambios repentinos de la variable de control. Afortunadamente para nuestro caso, el control de temperatura e iluminación no es tan exigente, como puede serlo en otras aplicaciones en las que el proceso tolera pequeños márgenes de variación de éstos parámetros.

El presente trabajo muestra un breve estudio tanto de transductores como de circuitos de control, por lo cual se divide en dos partes correspondientes a medición y circuitos controladores. En la parte de medición se presentan los tipos de transductores más representativos, junto con configuraciones sencillas de algunos circuitos medidores. En la parte de control se hace una descripción de algunos controladores ON-OFF más comunes, así también se presenta el caso de un controlador graduado, mediante el cual se pueden obtener diferentes niveles de respuesta en su salida como resultado de los distintos grados de incremento o decremento en su señal de entrada.

## 1.1 Medición de Temperatura .

Existen distintos métodos y principios en base a los cuales es posible implementar medidores de temperatura. Así también, se pueden realizar diferentes clasificaciones de estos, considerando cuestiones tales como su precisión, resolución, tecnología que utilizan, materiales de fabricación, costo, etc. En los puntos siguientes, se presentan los tipos de medidores de temperatura más comunes, poniendo especial énfasis en el sensor que se utiliza en cada caso.

**1.1.1 Termómetro con Termistor.** Los termistores son resistores sensibles a la temperatura, cuya resistencia varía inversamente con los cambios de temperatura. Por ejemplo, la resistencia de un termistor de 5000W, puede decrementarse hasta 20W por cada grado centígrado que aumente la temperatura. De esta forma, si el termistor es conectado a una fuente de corriente de 1.0 mA, podemos obtener una señal de voltaje que aumente en 200 mV/ °C. Por otra parte, este tipo de resistores presentan un alto coeficiente de temperatura resistivo, y en base a este parámetro se les clasifica en dos tipos: el de Coeficiente de Temperatura Negativo (NTC) y el de Coeficiente de Temperatura Positivo (PTC). En la práctica, sólo los Termistores del tipo NTC son usados para construir medidores de temperatura. Los materiales con los que se construyen estos dispositivos pueden ser por ejemplo cobre, constantán, nicromel, manganina, carbón y materiales cerámicos. Dependiendo del tipo del termistor, a 20°C puede presentar valores resistivos que van desde unos cuantos kW hasta 40 MW. A continuación se muestra el circuito básico de un medidor de temperatura empleando como sensor un termistor, el cual forma una de las piernas del puente de Wheatstone.

Este circuito se equilibra mediante el ajuste de la resistencia variable  $R_3$ , considerando la temperatura ambiente de trabajo, en tanto que el resto de las resistencias son de valor fijo. Una vez que cambia la temperatura, el termistor cambia su valor resistivo, y debido a que el dispositivo presenta un coeficiente de temperatura negativo, la resistencia del termistor disminuirá si la temperatura aumenta y viceversa; ésta situación provocará una variación en el voltaje presente en los puntos marcados con las letras "a" y "b", lugar en donde es conectado el voltmetro.

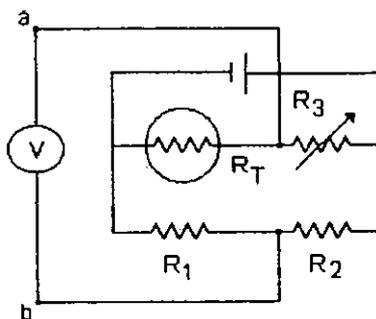


Fig. 1.1 Termómetro con Termistor

**1.1.2 Termómetro con RTD.** Las RTD son otra variante de los dispositivos termoresistivos, éstas al igual que los termistores, ven afectado su valor resistivo de acuerdo a los cambios de temperatura, pero la diferencia entre éstos es que las RTD cuentan con un coeficiente positivo de temperatura, lo cual implica que su resistencia se incrementará sólo si la temperatura aumenta, e igualmente su resistencia disminuirá si la temperatura decae. Existen diversos materiales utilizados para su fabricación, siendo la mayoría metales. Algunos de estos materiales pueden ser : plata, platino, níquel, hierro, cobre y alumel. De todos los materiales actualmente utilizados en la fabricación de dispositivos termoresistivos, el platino tiene las características óptimas para dar servicio en un amplio rango de temperaturas, y ha sido usado como un estándar para termómetros industriales en muchos países. Las RTD de platino de tipo industrial se encuentran disponibles con una variedad de valores básicos de resistencia los cuales se encuentran en el rango de 100 a 500  $\Omega$ , abarcando temperaturas que van desde -180 hasta 650°C.

Para la medición de temperaturas en el rango de -70 a 150°C, las RTD de níquel con resistencia de 10 a 1000  $\Omega$  han sido extensiva y exitosamente usadas. Comparadas con las de platino, las de níquel resultan tener mayor sensibilidad a los cambios de temperatura, no obstante, recientemente las RTD de platino completamente encapsuladas con su rendimiento superior y bajo costo han ganado mercado a sus contrapartes de níquel en la mayoría de las aplicaciones industriales. Actualmente la producción de RTD's de níquel está dedicada al remplazo de los dispositivos existentes que utilizan este tipo de sensor ya sea para control o medición.

El cobre electrolítico de alta pureza está actualmente disponible comercialmente, el cual también es utilizado para la fabricación de RTD's, este material tiene un coeficiente resistivo de temperatura ligeramente mayor que el platino. El rango útil de temperatura de las RTD de cobre está confinado al rango de -200 a 150°C, y son fabricadas con una resistencia base de 10 y 25  $\Omega$  y generalmente no exceden los 100  $\Omega$ .

El siguiente es un diagrama que muestra la implementación de un medidor digital de temperatura utilizando como elemento sensor una RTD. En éste circuito la RTD es parte de un divisor de voltaje; la variación de la temperatura provoca igualmente una variación de la resistencia de la RTD, lo cual se refleja como una variación de voltaje a la entrada no inversora del OPAM. Este voltaje es preamplificado e introducido a un circuito convertidor analógico-digital. Finalmente el valor de la temperatura es mostrado en un display.

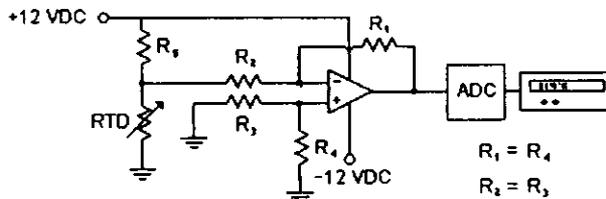


Fig. 1.2 Medidor digital de temperatura con RTD

**1.1.3 Termómetro con Semi-conductores.** Los diodos y los transistores son dispositivos formados a base de uniones de materiales semiconductores, cuyas relaciones corriente-voltaje son substancialmente determinadas por las relaciones entre los portadores de cada lado de la unión. Por otra parte, la densidad de portadores es fuertemente dependiente de la temperatura, por lo que un cambio en este parámetro, afectará la cacterística corriente-voltaje. Es muy común encontrar configuraciones de sensores a base de transistores bipolares con la base y el colector cortocircuitados, es decir, funcionando como un diodo. Esto se hace ya que el voltaje entre la base y el emisor es muy sensible a los cambios de temperatura.

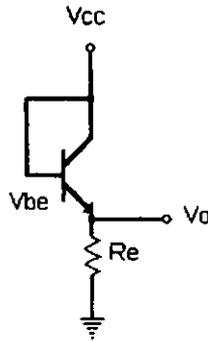
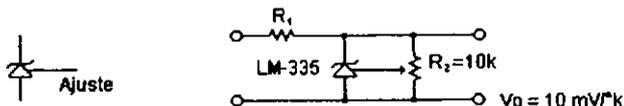


Fig. 1.3 Configuración de transistor como diodo

A continuación se muestran dos circuitos integrados muy empleados como sensores de temperatura, y cuyos principios de operación, se basan en lo ya expuesto.

- **Circuito Integrado LM-335.** Este dispositivo es un sensor de temperatura de tres terminales. Las dos principales son para la alimentación, así como para la salida, mientras que la tercera es para ajuste y calibración. El LM-335 es básicamente un diodo zener, en el cual, el voltaje de ruptura es directamente proporcional a la temperatura. El dispositivo entrega  $10\text{mV}^\circ\text{K}$  y abarca un rango de temperaturas desde  $-10$  hasta  $+100^\circ\text{C}$ . Los siguientes diagramas muestran el símbolo esquemático del dispositivo, así como algunos circuitos construidos a base de este circuito integrado.



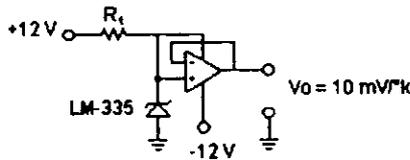


Fig. 1.4 Medidor de temperatura con el circuito LM-335

- **Circuito Integrado AD-590.** Este circuito es otra versión de sensores de temperatura de estado sólido. Este dispositivo particular, es un sensor a base de dos electrodos que opera como fuente de corriente con una característica de  $1\text{mA}/^\circ\text{K}$ . El AD-590 opera en el rango de temperatura de  $-55$  hasta  $+150^\circ\text{C}$ . Se le puede alimentar con un amplio rango de valores de voltaje que van desde 4 hasta 30 V DC.

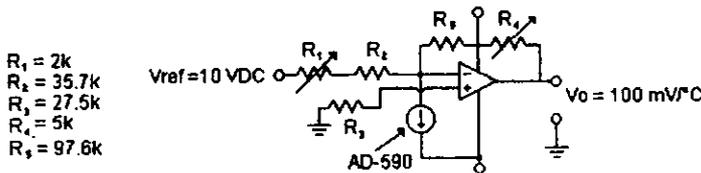


Fig. 1.5 Termómetro con el C.I. AD-590 como sensor

**1.1.4 Termómetro con Termopares.** Los Termopares entran en la categoría de transductores generadores, debido a que producen una señal eléctrica a partir de la magnitud que miden sin necesidad de alimentación. Históricamente fue primero Thomas J. Seebeck quien descubrió en 1822, que en un circuito de dos metales distintos, A y B, con dos uniones a diferente temperatura, aparece una corriente eléctrica. Es decir, hay una conversión de energía térmica a energía eléctrica, o bien, si se abre el circuito, una fuerza termoelectromotriz (f.t.e.m.) que depende de los metales y de la diferencia de temperaturas entre las dos uniones, aparece en los extremos del circuito abierto. Al conjunto de estos metales distintos con una unión firme en un punto o una zona, se le denomina Termopar. El empleo del Termopar exige que una de las uniones se encuentre bajo la temperatura a medir, mientras que la otra unión debe estar bajo una temperatura de referencia, la cual debe permanecer constante con el fin de evitar errores en la medición.

Respecto a las uniones del Termopar interesa que presenten las siguientes características: resistividad alta, coeficiente de temperatura bajo, resistencia a la oxidación a temperaturas altas y la mayor linealidad posible. Para lograr estas propiedades se emplean aleaciones especiales como son: níquel (90)/cromo (10), cobre (57)/níquel (43), níquel (94)/aluminio (20) / manganeso (3)/silicio (1), etc. Existen varios tipos de termopares, los cuales se clasifican por las aleaciones empleadas para su fabricación, asignando una letra del alfabeto para cada uno de los diferentes tipos; la aplicación específica del termopar depende del rango de temperaturas que se desee medir. En

la siguiente tabla se recogen las características de algunos de los termopares más comunes y su designación de acuerdo con las normas ANSI.

ANSI	Composición	Rango de temps.	Margen
B	Pt(6%)/Rodio-Pt(30%)/Rodio	138 a 1800	113.6
C	W(5%)/Renio-W(26%)/Renio	10 a 2300	137.0
E	Cromel-Constantán	10 a 982	175.0
J	Hierro-Constantán	-184 a 760	150.0
K	Cromel-Alumel	-184 a 1260	156.0
R	Pt(13%)/Rodio-Pt	10 a 1593	118.7
S	Pt(10%)/Rodio-Pt	10 a 1538	116.0
T	Cobre-Constantán	-184 a 400	126.0

Las características completas en cuanto a tensión de salida, se encuentran tabuladas de forma que se dan las tensiones correspondientes a distintas temperaturas cuando la unión de referencia está a 0°C. Esto significa que la unión a 0°C dé siempre una tensión de 0 v para cualquier termopar.

Según la aplicación, se dispone de distintos tipos de uniones. Las uniones desnudas se emplean para medidas estáticas o de flujos de gases no corrosivos, donde se requiere un tiempo de respuesta rápido. Las uniones aisladas se emplean para medir en ambientes corrosivos donde además interesa aislamiento eléctrico del termopar. Este queda entonces encerrado por una vaina y aislado de ésta por un buen conductor térmico como el aceite, mercurio, o polvo metálico. Si se desea una respuesta rápida, y no hace falta una vaina gruesa, se emplean aislantes minerales como polvo de MgO, Al<sub>2</sub>O<sub>3</sub>, o BeO.

Para finalizar este punto, se muestra a continuación una tabla comparativa, en la que aparecen desplegadas algunas características importantes de los tres tipos de sensores vistos hasta ahora, logrando con ésto una importante referencia para realizar comparaciones entre los dispositivos.

Parámetro	RTD de Platino 100 W	RTD de Níquel 1000 W	RTD de Balco 2000 W	Termisto	Termopar	Semiconductores
COSTO	Elevado	Regular	Elevado	Bajo	Bajo	Bajo
RANGO DE TEMP.	-400 a 1200 °F	-350 a 600 °F	-100 a 400 °F	-100 a 500 °F	-450 a 4200 °F	-57 a 257 °F
ESTABILIDAD	Buena	Regular	Regular	Pobre	Regular a Pobre	Buena
PRECISION	Alta	Medio	Baja	Medio	Medio	Medio
SENSITIVIDAD (OUT)	Medio	Alta	Muy Alta	Muy Alta	Bajo	Alta
LINEALIDAD	Buena	Regular	Regular	Pobre	Regular	Buena
AUTO CALENTAMIENTO	Bajo	Medio	Medio	Alto	--	Bajo
DIMENSIONES FISICAS	Medianas	Grandes	Grandes	Pequeñas	Medianas	Pequeñas

## 1.2 Medición de ILUMINACION.

Para esta parte se presenta una serie tanto de sensores como de circuitos de medición para iluminación. Inicialmente se proporciona una breve explicación de los principios de funcionamiento de los sensores, para posteriormente describir la operación de algunos circuitos diseñados a partir de éstos.

**1.2.1 Celda Fotovoltaica.** Un tipo muy común de sensor fotoeléctrico es la celda fotovoltaica, ó fotocelda, aunque también se le conoce como batería solar, especialmente cuando se unen varias unidades de éstas. El símbolo esquemático para una celda fotovoltaica es mostrado en la figura 1.6a, en la cual se puede observar que este símbolo es muy parecido al de la batería común. Esta similitud implica como es lógico que es usada para la misma función. Una fotocelda es básicamente una unión p-n expuesta ópticamente. El silicón, es el material comúnmente usado, aparte de los demás materiales fotosensitivos, debido a que brinda un nivel de voltaje relativamente más alto para un determinado nivel de iluminación. Una fotocelda es virtualmente idéntica en construcción a un diodo estándar. La mayor diferencia es que en un diodo ordinario la unión p-n está protegida, mientras que en la fotocelda la unión está intencionalmente expuesta a las fuentes de luz externas.

Para maximizar el efecto fotoeléctrico, la unión p-n en una fotocelda se extiende a lo largo de una lámina delgada. Esto proporciona la mayor área de contacto posible. Cuando esta superficie de silicón está protegida de la luz, no fluye corriente a través de la fotocelda. Sin embargo, cuando se expone a una fuente luminosa, una pequeña cantidad de voltaje es generada como resultado del efecto fotoeléctrico. De tal forma que si la fotocelda es conectada a una carga y a su vez se le expone a una fuente de luz como se ilustra en la figura 1.6b, entonces se tendrá una corriente fluyendo a través del circuito alimentando a la carga. La cantidad de corriente en el circuito dependerá de la intensidad luminosa incidiendo sobre la superficie de la fotocelda. Mientras más brillante sea la luz, mayor será la cantidad de corriente que la fotocelda puede proporcionar a la carga.

El voltaje proporcionado por la mayoría de las celdas fotovoltaicas disponibles comercialmente varía de 0.5 v a 12 v. Voltajes mayores pueden ser obtenidos conectando múltiples fotoceldas en serie para crear una verdadera batería. Si la aplicación lo que exige es más corriente, entonces se puede hacer una conexión en paralelo de varias fotoceldas para de esa manera sumar las contribuciones de corriente de cada unidad.

Fotoceldas que proporcionan gran potencia han sido desarrolladas para usar la energía solar como una fuente alterna. Esas celdas trabajan en una forma similar a las celdas mucho más pequeñas empleadas en la electrónica. Materiales especializados tales como óxidos de cobre son usados en esas fotoceldas a gran escala. Para obtener los niveles de energía mayores, la superficie expuesta por celda debe de ser mayor. Nuevamente, la potencia de salida de la batería solar está limitada por el espacio disponible para el montaje eficiente de fotoceldas adicionales, ésta es la razón por la cual, estas fuentes de energía no han sido tan utilizadas en la industria, y no será vista como una fuente primaria de energía hasta que no se mejoren las técnicas de construcción de éstos dispositivos.

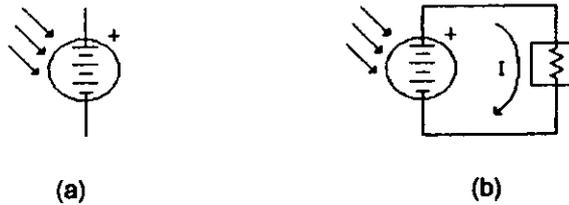


Fig. 1.6 a) símbolo de la fotocelda, b) fotocelda alimentando una carga

**1.2.2 Dispositivos Fotoconductores.** Este tipo de sensores, consisten esencialmente de una película fabricada con un material cuya resistencia es una función de la iluminación incidente. La película puede ser ya sea de material semiconductor intrínseco o extrínseco, pero para la parte visible del espectro, el calcogeniuro, el sulfuro de cadmio y el seleniuro de cadmio, son los más comunes. El tamaño y la forma de la película activa, determina tanto la resistencia en la oscuridad como la sensibilidad.

La resistencia  $R$  de una celda decrece al tiempo que la iluminación se incrementa, así que es común considerar el inverso de esta resistencia o la conductancia  $G$ , para ser el parámetro básico, por lo que a éstas celdas también se les conoce como *celdas fotoconductoras*.

En la siguiente figura se muestra el símbolo de una celda fotoconductora, así como su gráfica característica, en donde se puede observar una relación prácticamente lineal entre su resistencia y la intensidad de la luz incidente en la superficie de la celda.

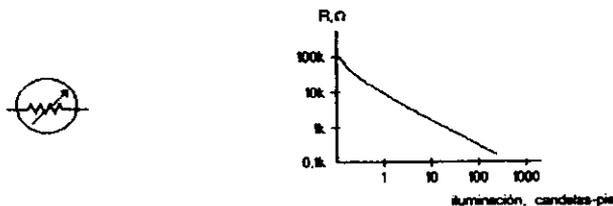


Fig. 1.7 Símbolo eléctrico y gráfica característica de una celda fotoconductora

En una celda de sulfuro de cadmio, el pico espectral se encuentra cercano a los 5100 nm, mientras que para las fabricadas con seleniuro de cadmio este punto máximo se encuentra cercano a los 6150 nm. El tiempo de respuesta de las celdas de sulfuro de cadmio es casi de 100 ms, y de 10 ms para las unidades de seleniuro. Estas celdas pueden ser fabricadas para operar en niveles de luminosidad de  $10^{-3}$  -  $10^3$  candelas-pie. La resistencia de un fotoreistor puede ir desde 10 M $\Omega$  en la oscuridad hasta 10  $\Omega$  en la luz brillante.

**1.2.3 Fotodiodo.** El fotodiodo es un dispositivo de fotounión, el cual está formado de una unión p-n elaborada usualmente a base de silicón. Si un fotón incide en la unión con suficiente fuerza como para brincar de nivel de energía, se produce un par electrón-hoyo, el cual modifica las características de la unión como se muestra en la siguiente figura.

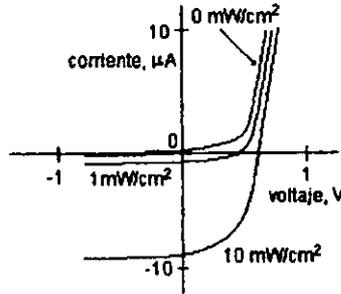


Fig. 1.8 Características voltaje-corriente de una unión p-n de silicón irradiada.

De esta figura, vemos las tres diferentes curvas. Para irradiancia cero, las características tanto de polarización directa como inversa son normales. Para una irradiancia de  $1\text{mW/cm}^2$ , el voltaje de circuito abierto es de  $500\text{ mV}$  y la corriente de corto circuito es de  $0.8\ \mu\text{A}$ . Para  $10\text{ mW/cm}^2$ , el voltaje de circuito abierto es de  $600\text{ mV}$  y la corriente de corto circuito es de  $8\ \mu\text{A}$ . Si la unión es polarizada en inversa, la fotocorriente inversa ( $\mu\text{A}$ ) es proporcional a la intensidad luminosa ( $\text{mW/cm}^2$ ). Características típicas de los fotodiodos son mostradas en la figura 1.9.

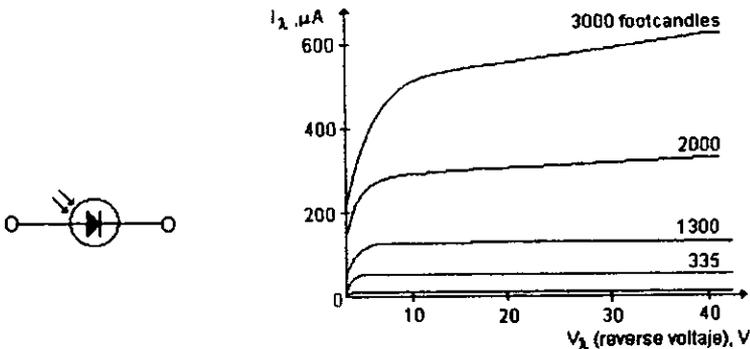


Fig. 1.9 Símbolo eléctrico del fotodiodo y su gráfica característica

A diferencia de las celdas conductoras, los fotodiodos tienen un tiempo de respuesta mucho más rápido, el cual se encuentra por debajo del rango de los microsegundos. Tanto la respuesta espectral como la velocidad, pueden ser determinadas por la geometría y el dopado de la unión. Incrementando el área de la unión, se incrementa la sensibilidad (fotocorriente por unidad de irradiación) del fotodiodo mediante la colección de más fotones, pero también se incrementa la capacitancia, la cual puede elevar el tiempo de respuesta.

Por otra parte, todos los detectores comunes de luz fabricados a base de silicón, consisten de un fotodiodo y un amplificador. El fotodiodo se fabrica sobre un simple chip de silicón empleando el mismo proceso de dopado que se utilizó en la sección del amplificador. En la mayoría de los dispositivos comerciales, la corriente del fotodiodo está comprendida en el rango de submicroamperes hasta décimas de microampere. Todas las uniones de diodos semiconductores son fotosensitivas en algún grado sobre un rango determinado de longitudes de onda de luz. La respuesta de un diodo ante una longitud de onda en particular depende del material semiconductor usado y del espesor de la unión del diodo.

**1.2.4 Fototransistor.** El transistor sensible a la luz es una de las combinaciones más simples de fotodiodo y amplificador. Dirigiendo luz hacia la unión p-n polarizada en inversa que forman la base y el colector de un transistor bipolar, se genera una corriente en la base, la cual es amplificada por el factor de ganancia ( $h_{FE}$  ó  $\beta$ ) de corriente propio del amplificador (transistor). Un alto valor de  $h_{FE}$  así como una área de unión base-colector grande, son requeridos para que el foto-transistor tenga una alta sensibilidad, la cual pueda originar flujo de corriente aún en condiciones de poca luminosidad. Un valor típico de tales corrientes está en el orden de 1 nA con voltaje de polarización de 10 V, aunque las especificaciones de fototransistores normalmente garantizan corrientes más grandes en condiciones de obscuridad, las cuales están en el orden de 50 a 100 nA.

El fototransistor es equivalente a un fotodiodo en combinación con un transistor bipolar como se muestra en la figura 1.13. Típicamente, las amplificaciones de corriente entre 100 y 1000 son empleadas dependiendo de la aplicación. El área efectiva del fototransistor es usualmente de alrededor de  $0.5 \times 0.5 \text{ mm}^2$ . Los tiempos de conmutación de los fototransistores son dependientes de las amplificaciones de corriente y de la magnitud de las cargas, siendo así estos tiempos de valores entre  $30 \mu\text{s}$  y  $1 \mu\text{s}$ . Las frecuencias de corte resultantes llegan a valores de cientos de kilohertz. Aunque es posible emplear transistores para identificar diferentes niveles de luminosidad, es más común emplearlos como interruptores o como aisladores.



Fig. 1.10 a) símbolo electrónico b) circuito equivalente

**1.2.5 Circuitos de medición para iluminación.** Las fotoresistencias son ampliamente utilizadas en la implementación de medidores, formando distintas configuraciones electrónicas dependiendo del grado de exactitud que se desee en el medidor. Aquí se muestran algunos circuitos básicos utilizando las típicas configuraciones de puente de Wheatstone, así como otro a base de amplificadores operacionales.

Quizá la configuración mas elemental es la que se muestra en la siguiente figura, la cual consta únicamente de una fotoresistencia conectada en serie con un potenciómetro de ajuste y un miliamperímetro. En éste circuito,  $M_1$  es un simple miliamperímetro de dc que puede ser del rango de 0 a 1 mA.  $R_1$  es un potenciómetro que sirve para controlar la sensibilidad del medidor.  $R_2$  es la fotoresistencia, la cual permitirá un mayor o menor flujo de corriente en el circuito dependiendo de la cantidad de luz que incida en su superficie.

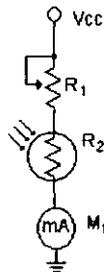


Fig. 1.11 Medidor básico de iluminación

El siguiente es un diagrama del medidor basado en puente de Wheatstone, en el cual una de las ramas la forma un transistor FET, que tiene como tarea amplificar la corriente que entrega la fotocelda en la compuerta del transistor. Las otras 3 ramas están formadas por las resistencias  $R_1$ ,  $R_2$  y el potenciómetro  $R_3$  con el cual se realiza el ajuste a cero (estabilización del puente) cuando se tienen condiciones de obscuridad.

El uso del amplificador FET en éste circuito incrementa la sensibilidad en un factor de dos a tres. El incremento exacto de la sensibilidad depende de las características específicas de la fotocelda y del FET. Con cualquier combinación funcional, se da un significativo incremento en la sensibilidad.

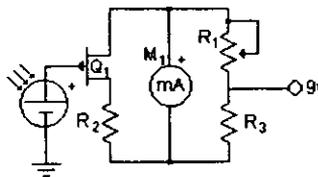


Fig. 1.12 Puente de Wheatstone con fotocelda y amplificador

Una de las características en común que presentan tanto las fotoresistencias como las fotoceldas, es que poseen un comportamiento lineal. En el caso del fotodiodo como se observó en la figura 1.9, muestra una característica exponencial. Hay algunas aplicaciones que requieren una medición en forma logarítmica, por lo que a continuación se presenta un medidor construido a base de fotodiodo.

Este circuito consta del sensor, un amplificador operacional, el miliamperímetro, un diodo, resistencias y capacitores, por lo que al igual que los anteriores circuitos resulta bastante sencillo y económico implementarlo. La sensibilidad de medidor logarítmico puede ser controlada por el potenciómetro R3, el cual por lo general es un trimpot que se sella una vez que se realiza el ajuste. En niveles de iluminación muy bajos, la salida del amplificador puede hacerse negativa lo que podría ocasionar daño en un amperímetro analógico, ya que la aguja se deflexionaría en sentido contrario, y para prevenir esto se utiliza un diodo a la salida del amplificador, el cual impide que la bobina del miliamperímetro se energice en una forma inadecuada.

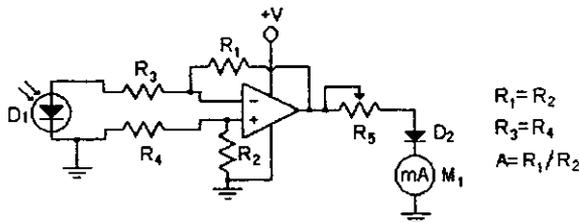


Fig. 1.13 Medidor logarítmico con fotodiodo

## 1.3. Controladores

Los principios en los que se basan los métodos de control tanto de temperatura como de iluminación no difieren en mucho de los utilizados en los dispositivos de medición, e incluso en algunos casos llegan a combinarse las dos funciones (medición y control) en la misma aplicación. Es por esto que la presente sección se apoya totalmente en lo descrito con anterioridad en la parte de medición. Los sistemas de control presentados son esquematizados en su forma más simple, con el fin de entender los conceptos involucrados en el caso.

**1.3.1 Controles Electromecánicos de temperatura.** En este tipo de controles, el sensor es mecánico, pudiendo ser de dilatación, a base de bimetales ó manométrico. Cualquiera que sea el tipo, el resultado final es un desplazamiento mecánico, el cual, se puede aprovechar para cerrar o abrir contactos eléctricos, dando como resultado el control de equipos de una amplia gama de potencias.

A continuación, se muestra un ejemplo de control muy sencillo pero que ha encontrado bastante aplicación sobre todo en los equipos de aire acondicionado.

- Control a base de Switch de Mercurio.** El elemento sensitivo en este caso, puede ser cualquiera de los mencionados, aunque por lo general, se trata de un bimetálico. El switch está formado a base de un tubo herméticamente sellado en cuyo interior se encuentran dos contactos con terminales al exterior, y una cierta cantidad de mercurio. El switch se monta en una base en posición inclinada, de tal forma que los contactos queden en la parte superior del tubo, y el mercurio en la parte inferior. La base completa se encuentra unida al mecanismo sensor, así al presentarse una deflexión producto de un cambio de temperatura, la base con todo y switch gira, permitiendo que el mercurio cierre el contacto. En la figura 1.15 se muestra un circuito básico para el encendido de un motor, en el cual se observa como las terminales del switch de mercurio se encargan de energizar la bobina del relevador, logrando así que los contactos del relevador manejen la potencia que finalmente permitirá el arranque y la operación permanente del motor.

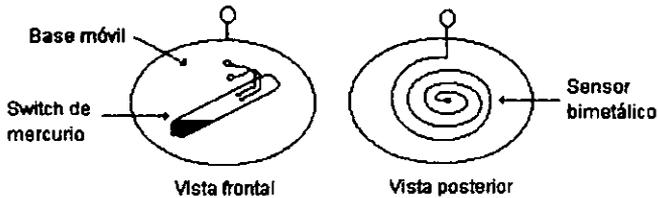


Fig. 1.14 Control con switch de mercurio

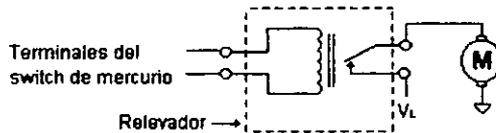


Fig. 1.15 Control básico para la activación de una carga

- Control graduado a base de Potenciómetro.** El control realizado como se describe en el punto anterior es del tipo todo o nada, encendido o apagado. Existen algunas aplicaciones en que se requiere un control gradual, que no se podría obtener mediante dos contactos. Cuando se tiene esta necesidad, es muy común la utilización de potenciómetros como elemento de graduación. Este arreglo se muestra en la siguiente figura, y su funcionamiento sería el siguiente: un potenciómetro para tal aplicación típicamente consiste de un núcleo con devanado de alambre desnudo, sobre el cual se deslizará una laminilla, ésta a su vez, se encuentra unida al elemento sensor. Con esta disposición, el potenciómetro queda partido en dos resistencias, cuyas magnitudes variarán de acuerdo a la posición de la laminilla (cursor), incrementándose la magnitud de la resistencia en un segmento y disminuyendo en el otro.

Al existir una variación en la temperatura, el sensor presentará una deflexión o un incremento en su longitud, haciendo que la laminilla se desplace sobre el potenciómetro. Como resultado, se tendrá una cantidad variable de voltaje en la salida.

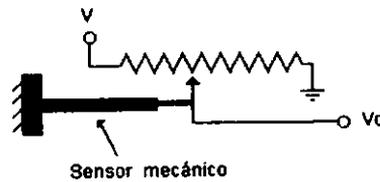


Fig. 1.16 Control electromecánico graduado

**1.3.2 Controles Electrónicos.** En la actualidad, los controles electrónicos se pueden ver en un sin número de aplicaciones, y día con día abarcan un mayor campo de aplicación. Debido a las grandes ventajas que ofrecen, han ido desplazando a los tradicionales controles basados en principios mecánicos como los que se describieron. Existen varias razones por las cuales los controles electrónicos resultan ideales en la mayoría de las aplicaciones, algunas de estas razones son las siguientes:

- Mayor velocidad de respuesta ante el cambio de la variable de control.
- Un control más preciso.
- Disminución e inclusive eliminación de las tareas de mantenimiento del controlador.
- Alta flexibilidad y eficiencia.
- Control de variables y procesos con un alto grado de complejidad.
- Capacidad de integración a redes de control.
- Capacidad de comunicación con varias etapas de un mismo proceso
- Dimensiones reducidas de los equipos.

- **Controladores tipo "ON-OFF".** Existen muchas configuraciones posibles para la implementación de éste tipo de controladores, que de hecho son los más simples. Aquí la tarea del sensor es determinar en que momento la variable de control sobrepasa cierto umbral. Este nivel de umbral es el que va a determinar cuando el actuador activará o desactivará la carga. Una de las configuraciones más ampliamente utilizadas es la que trabaja en base a un puente de Wheatstone como elemento de detección.

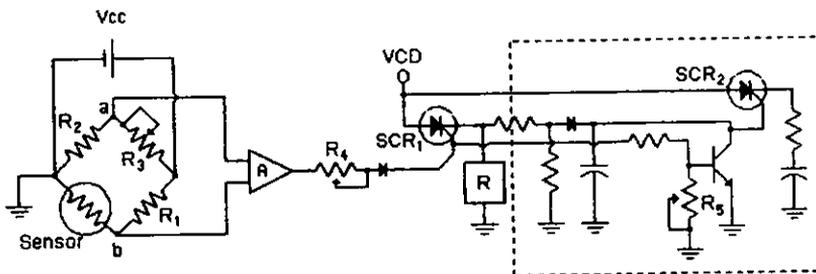


Fig. 1.17 Controlador con puente de Wheatstone y SCR's

El circuito mostrado en la figura 1.17 es un ejemplo de controlador on-off. En este caso, la carga deberá ser de C.D. Como se puede observar, en éste circuito no se especifica el tipo de sensor, pudiendo ser de temperatura u óptico. El tipo de actuador que se utiliza en este ejemplo es de estado sólido, la razón por la cual se utilizan 2 SCR's es para energizar y desenergizar la carga. El SCR<sub>1</sub> se encarga de la energización al momento que su compuerta recibe un pulso del amplificador, en tanto que el SCR<sub>2</sub> actúa para abrir al primer SCR mediante disminución de corriente. Los potenciómetros R<sub>1</sub> y R<sub>2</sub> controlar los niveles de conexión y desconexión respectivamente de la carga. Para simplificar la etapa del actuador es posible utilizar un circuito similar al presentado en la figura 1.15, en el cual el que actúa es un dispositivo electromecánico. Este tipo de actuadores son útiles cuando la velocidad de conmutación no importa y únicamente se requiere controlar la conexión y desconexión de circuitos. Existen otras configuraciones igualmente funcionales que involucran menos dispositivos y ofrecen buenos resultados en la mayoría de las aplicaciones.

En la figura 1.18 se presentan 3 controladores on-off para iluminación, aunque bien pudieran servir para controlar temperatura mediante la sustitución del sensor óptico por uno de temperatura, ya sea del tipo resistivo o generador.

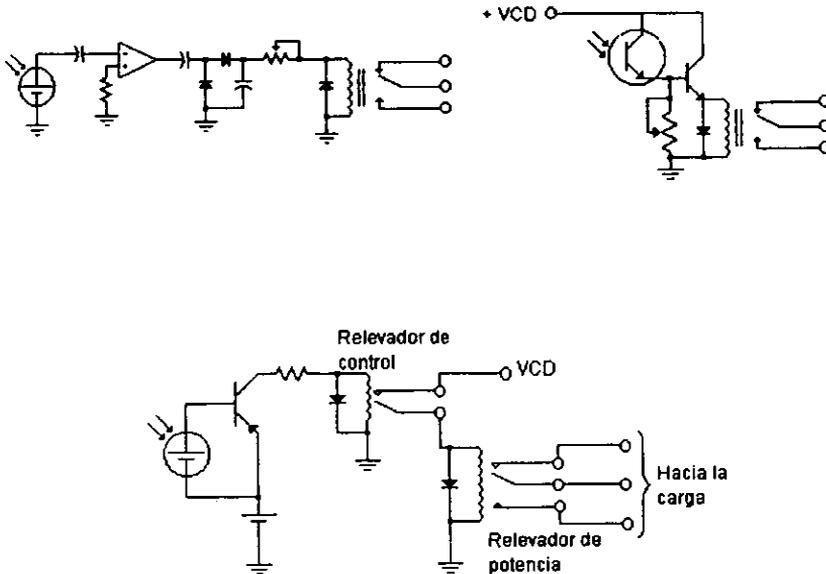


Fig. 1.18 Controladores ON-OFF de iluminación

**1.3.3 Controles graduados.** Los métodos de control de ON-OFF son por mucho los más sencillos de implementar, ya que únicamente encienden y apagan cargas sin mayor contemplación. La tarea de realizar electrónicamente control en base a los distintos niveles de señal que arrojan los sensores, se vuelve un poco más complicado, en especial cuando los niveles de potencia son considerables. Una de las técnicas más utilizadas para el control de cargas tanto grandes como regulares, es la variación del nivel de voltaje que se entrega a la carga mediante el disparo de tiristores en cierto

ángulo de fase de la señal de voltaje. Como se sabe en nuestro país las compañías generadoras entregan al usuario el voltaje en determinados niveles dependiendo la aplicación, pudiendo ser estos 220 kV, 115 kV, 85 kV, 23 kV, 13.8 kV, 440V, 220 V y 120V por mencionar algunos, pero independientemente de la magnitud del voltaje, la frecuencia siempre será de 60 Hz, tomando esto en cuenta tenemos que un periodo de cualquiera de estas señales es de 16.6 ms. La siguiente figura muestra una forma de onda de voltaje, en la cual se observan los distintos ángulos en los que se divide. El voltaje que ve la carga no es el valor instantáneo que se tiene en un tiempo 't' o en un ángulo ' $\alpha$ ', sino el área bajo la curva de cada semiciclo. De esta forma, si nosotros limitamos el voltaje, entregando a la carga energía únicamente después de la detección de un ángulo ' $\alpha$ ', entonces el nivel de voltaje que realmente alimentará a ésta será menor.

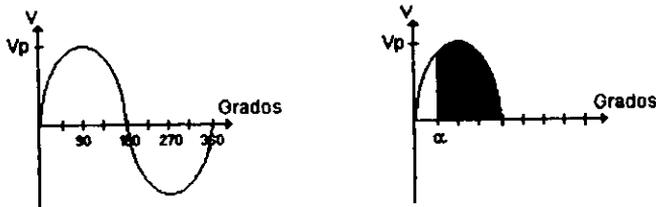


Fig. 1.19 Formas de onda de una señal de voltaje

Los circuitos variadores de voltaje no hacen sino explotar éste principio, teniendo como primordial tarea entonces detectar el ángulo a partir del cual se activarán los tiristores para liberar energía hacia la carga. A continuación se presenta en forma de bloques el funcionamiento de un variador de voltaje por disparo de ángulo de fase. El primer bloque lo forma la etapa reductora de voltaje que por un lado alimenta al circuito y por el otro proporciona una muestra de la onda de voltaje. Esta muestra de voltaje se presenta a la entrada no inversora de un amplificador operacional en configuración de comparador, teniendo éste su entrada inversora a nivel de tierra; debajo de cada bloque se observa la forma de onda perteneciente a cada etapa, con lo cual se puede observar como se genera una señal cuadrada a la salida del detector de cruce por cero. El objetivo de ésta etapa es precisamente detectar cuando la señal alterna presenta una magnitud de cero volts, con lo cual se sincronizarán las restantes etapas. Mediante un diodo y un divisor de voltaje se escala la parte positiva de la señal cuadrada, la cual posteriormente es integrada para generar una función rampa. Esta última señal debido a su linealidad durante todo el semiciclo es la que nos servirá para detectar más fácilmente el ángulo de fase que requerimos. Sin la participación de ésta etapa no sería posible hacer disparos en ángulos que rebasaran los 90°. La siguiente etapa la forma nuevamente un comparador de voltaje, el cual finalmente generará el pulso de disparo al alcanzar la señal de rampa una magnitud superior a la del voltaje de referencia. Este voltaje de referencia puede ser obviamente ajustado para producir disparos a diferentes ángulos, pudiendo implementarse entonces desde un controlador manual a base de potenciómetro, hasta un sistema más sofisticado basado en microprocesador.

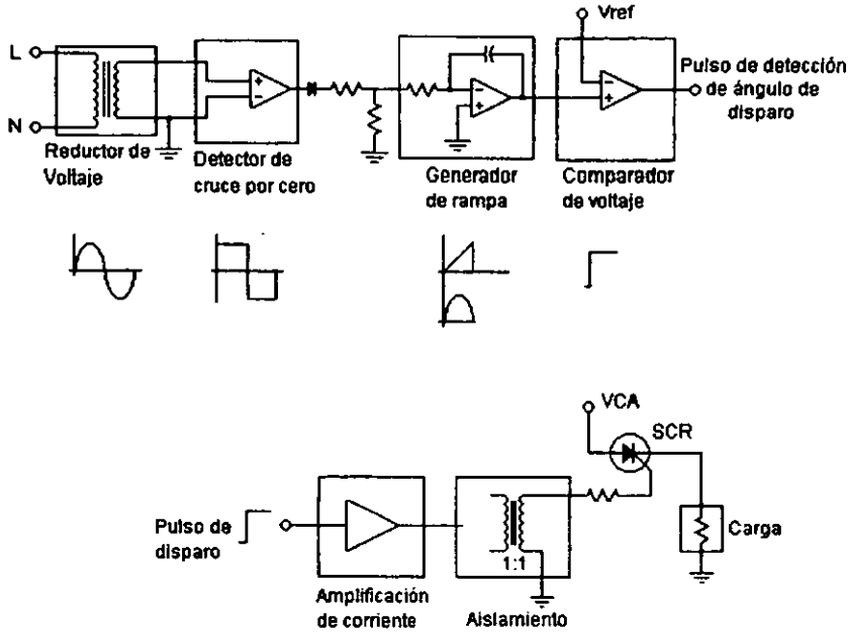


Fig. 1.20 Circuito a bloques del variador de voltaje por ángulo de disparo

Los tiristores de gran capacidad de conducción demandan igualmente un nivel de activación de compuerta mayor que los dispositivos de menor capacidad, por lo cual se hace necesario amplificar en corriente al pulso de disparo. Una vez activado el tiristor, la carga recibirá la porción del semiciclo positivo de la onda de voltaje que corresponde al ángulo en el que se presentó el pulso. La desactivación del SCR se da de forma natural al disminuir la corriente que circula a través de él. En caso que la carga a controlar requiera ser alimentada con corriente alterna, se adiciona un integrador que genere una función rampa con cada semiciclo negativo, para de esa manera activar un segundo SCR, que permita el paso de corriente de polaridad negativa hacia la carga.

## CAPITULO 2:

### Conjuntos Difusos: conceptos y operaciones básicas

---

#### 2.1 Conjuntos exactos y conjuntos difusos.

En términos generales, un conjunto es una colección de elementos que comparten ciertas características. En el caso de los conjuntos exactos (tradicionales), los elementos que los forman se encuentran exactamente definidos en cuanto a su grado de pertenencia al conjunto. De esta forma, algunos conjuntos exactos podrían ser: el conjunto de números mayores a 10 y menores a 100, el conjunto de automóviles de 2 puertas en un lote de autos, o el conjunto de profesores de matemáticas en una escuela. Cada uno de estos conjuntos cuenta con un universo bien definido, es decir, no existe ninguna duda de la pertenencia de un elemento a su conjunto, o pertenece o no pertenece. Formalmente, la teoría de conjuntos exactos define el grado de pertenencia de cada elemento de un universo a un conjunto mediante una función de asignación o de membresía que se define de la siguiente manera:

**Def. 2.1.1** Sea "x" un elemento definido en un conjunto universal X y A un subconjunto de X, entonces el grado de membresía de "x" al subconjunto A estará dado por la siguiente función característica

$$\mu_A(x) = \begin{cases} 1 & \text{si y sólo si } x \in X \\ 0 & \text{si y sólo si } x \notin X \end{cases}$$

Esta definición no hace sino dejar al descubierto el carácter dicotómico de la teoría de conjuntos exactos, en ella no existe la posibilidad de representar condiciones de ambigüedad o inexactitud, en pocas palabras, no es posible asignar grados de membresía intermedios que representen conceptos que no estén bien definidos. Basados en esto y a manera de ejemplo, no tenemos problema para representar el conjunto "calles". Pero qué pasa si queremos representar matemáticamente el conjunto "calles largas", cómo se representaría el concepto "largas", o mas bien que tan largo es "largas", 20 m, 50m, o quizá 100m. Evidentemente existe un alto grado de ambigüedad en este conjunto, y con las herramientas matemáticas tradicionales no es posible hacer una representación y mucho menos un modelo matemático de fenómenos o situaciones con niveles de incertidumbre.

La teoría de Conjuntos Difusos por su parte, permite una transición más suave entre el nivel 0 de pertenencia y el 1 de pertenencia absoluta, es decir, en los conjuntos difusos es posible encontrar elementos con grados de pertenencia como 0.1, 0.2, 0.33, 0.57, 0.999, etc. Como se puede ver,

la teoría de conjuntos difusos no es una herramienta bi-valuada, o tri-valuada, sino mas bien n-valuada, ya que existen tantos niveles de certeza como sea posible. Ahora, en lugar de sentenciar que un elemento pertenece o no pertenece a un conjunto, es posible decir que 'mas o menos pertenece' o que 'pertenece casi mucho' o 'muy poco'. Al igual que la teoría de conjuntos exactos, la teoría de conjuntos difusos asigna formalmente grados de membresía mediante una función característica cuya definición se presenta como:

**Def. 2.1.2** Sea "x" un elemento definido en un conjunto universal X y A un subconjunto difuso de X, entonces el grado de membresía de "x" al subconjunto A estará dado por la función característica  $\mu_A(x)$ .

**Ejemplo 2.1.1**

Sea A el conjunto de los "números reales cercanos a 10", cuya función de membresía dada por la expresión  $\mu_A(x) = (1 + (x-10)^2)^{-1}$  muestra la siguiente gráfica.

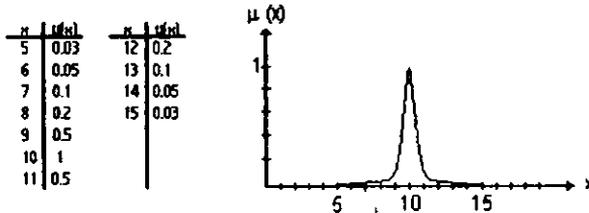


Fig. 2.1 Gráfica del ejemplo 2.1

Como se ve en el ejemplo, la función característica asigna a cada elemento en el dominio de x un grado de membresía, el cual de acuerdo con el argumento de la función estará comprendido en el rango de [0,1], lo cual es lo más común pero no indispensable, de hecho es posible utilizar cualquier otro conjunto de valores siempre y cuando presenten un orden lógico que permitan representar distintos grados de pertenencia.

La forma de representar a los elemento de un conjunto difuso es mediante duplas formadas por el elemento mismo y su grado de membresía, de ésta forma  $(\mu_A(x), x)$  es un elemento del conjunto difuso A, por lo cual se dice que A es un conjunto de pares ordenados. Para simplificar la representación de los conjuntos difusos, es conveniente usar la siguiente notación:

**Def. 2.1.3** Sea A un conjunto difuso en el conjunto universal X, y  $\mu_A(x_1)/x_1, \mu_A(x_2)/x_2, \dots, \mu_A(x_n)/x_n$  los elementos que lo integran, entonces la notación para el conjunto difuso A es como sigue:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n = \sum \mu_A(x_i)/x_i$$

donde el símbolo "/" es de relación mas no de división aritmética.

En el caso del conjunto difuso del ejemplo 2.1, éste quedaría representado como:

$$A = \dots + 0.03/5 + 0.05/6 + 0.1/7 + 0.2/8 + 0.5/9 + 1/10 + 0.5/11 + 0.2/12 + 0.1/13 + 0.05/14 + 0.03/15 + \dots$$

## 2.2 Conceptos Básicos.

Sea A un conjunto difuso definido en el conjunto universal X, de tal forma que :

$$A = 0.75/100 + 1/80 + 0.69/75 + 0.44/30 + 1/120 + 0.1/110 + 0.3/50 + 0.6/75 + 0.15/105 + 0.0/10$$

$$B = 1/10 + 0.25/5 + 0.78/6 + 0.95/11 + 0.21/19 + 0.3/8 + 1/13$$

entonces se definen los siguientes conceptos de conjuntos difusos:

- **Soporte "Supp A"**. El soporte de un conjunto difuso se define como el conjunto exacto formado por todas las  $x \in X$  tal que sus grados de membresía sean mayores o iguales a cero, formalmente:

$$\text{Supp } A = \{ x \in X \mid \mu_A(x) \geq 0 \}$$

### Ejemplo 2.2.1

$$\text{Supp } A = \{ 100, 80, 75, 30, 120, 110, 50, 75, 105 \}$$

- **Altura**. La altura de un conjunto difuso está definida por el máximo grado de membresía de cualquiera de sus elementos. Así, la altura del subconjunto A propuesto sería 1.
- **Corte " $\alpha$ "**. Sea A un conjunto difuso en el conjunto universal X, entonces un corte- $\alpha$  de A que se denota como  $A_\alpha$ , es un conjunto exacto formado por todas las  $x \in X$  tal que sus grados de membresía sean mayores o iguales a  $\alpha$ , formalmente:

$$A_\alpha = \{ x \in X \mid \mu_A(x) \geq \alpha \}$$

### Ejemplo 2.2.2

$$A_{0.5} = \{ 100, 80, 75, 120, 75 \}$$

- **Conjunto Nivel**. Se define así al conjunto de todos los cortes- $\alpha$  de un conjunto difuso, de tal forma que :

$$\Lambda_A = \{ \alpha \mid \mu_A(x) = \alpha \text{ para algún } x \in X \}$$

- **Conjunto Difuso Normal.** Un conjunto difuso se dice que es normal si por lo menos uno de sus elementos presenta un grado de asociación (membresía) de "uno".
- **Cardinalidad Escalar** " $|A|$ ". La cardinalidad escalar de un conjunto difuso, es un número escalar definido por la suma total de los grados de membresía de cada uno de sus elementos.

$$|A| = \sum \mu_A(x) \quad \text{para toda } x \in X$$

**Ejemplo 2.2.3**

$$|A| = 0.75 + 1 + 0.69 + 0.44 + 1 + 0.1 + 0.3 + 0.6 + 0.15 + 0.0 = 5.03$$

- **Cardinalidad Relativa** " $||A||$ ". La cardinalidad relativa de un conjunto difuso A, es un número escalar definido por el cociente de la cardinalidad escalar de A entre la de X, es decir:

$$||A|| = |A| / |X|$$

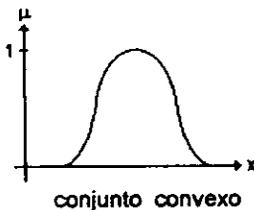
- **Cardinalidad Difusa.** Este tipo de cardinalidad es definida como un número difuso en lugar de un número real, como es el caso para la cardinalidad escalar. Cuando un conjunto difuso A tiene un soporte finito, su cardinalidad difusa  $| \sim A |$  es un conjunto difuso definido en N cuya función de asociación está definida por

$$\mu_{| \sim A |}(\alpha) = \mu_A(\alpha)$$

para toda  $\alpha$  en el conjunto nivel de A.

- **Conjunto Convexo.** Un conjunto difuso A es *convexo*, si todos sus cortes- $\alpha$  son convexos, ó si se cumple la siguiente función:

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(\lambda x_2)), \quad x_1, x_2 \in X, \quad \lambda \in [0,1]$$



- **Subconjunto.** Sean A y B dos conjuntos difusos definidos en X, entonces decimos que A es subconjunto de B si y sólo si el grado de membresía de A es menor o igual que el de B.

$$\mu_A(x) \leq \mu_B(x)$$

- **Subconjunto Propio.** Sean A y B dos conjuntos difusos definidos en X, entonces decimos que A es un subconjunto de B si y solo si el grado de membresía de A es menor que el de B.

$$\mu_A(x) < \mu_B(x)$$

- **Conjuntos Difusos Iguales.** Dos conjuntos difusos son iguales cuando sus grados de membresía son iguales.

$$\mu_A(x) = \mu_B(x)$$

- **Conjuntos Difusos Diferentes.** Dos conjuntos difusos son diferentes cuando sus grados de membresía son diferentes.

$$\mu_A(x) \neq \mu_B(x)$$

- **Conjuno Difuso Vacío.** Un conjunto difuso es vacío si tiene un soporte vacío, esto es, la función de membresía asigna cero a todos los elementos del conjunto universal.

- **Principio de Extensión.** Una de las ideas básicas de la teoría de conjuntos difusos, es la que proporciona una extensión general de los conceptos matemáticos exactos al ambiente difuso, esta idea es precisamente el *principio de extensión*. Esta es una identidad básica que permite al dominio de definición de una relación ser extendido desde puntos en X hacia subconjuntos difusos de X. Más específicamente, supongamos que f es una relación de X a Y y A es subconjunto difuso de X expresado como:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n$$

entonces el principio de extensión establece que:

$$f(A) = f(\mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n) = \mu_A(x_1)/f(x_1) + \mu_A(x_2)/f(x_2) + \dots + \mu_A(x_n)/f(x_n)$$

así la imagen de A bajo f puede ser deducida del conocimiento de la imagen de  $x_1, \dots, x_n$  bajo f.

### Ejemplo 2.2.4

Sea  $X = 1, 2, 3, \dots, 7$  y "pequeño" un subconjunto difuso en X definido por  $\text{pequeño} = 1/1 + 1/2 + 0.8/3 + 0.5/4$ . Si tomamos a  $f(x) = x^2$ , entonces:

$$f(\text{pequeño}) = \text{pequeño}^2 = 1/1 + 1/4 + 0.8/9 + 0.5/16.$$

- **Tipos de Conjuntos Difusos.** Un conjunto difuso de tipo 1 es un conjunto difuso ordinario en  $X$ ; un conjunto difuso tipo  $k$  (con  $k > 1$ ) en  $X$  es un conjunto  $L$ -difuso cuyos valores de membresía son conjuntos difusos del tipo  $(k-1)$  en  $[0, 1]$ .

De esta manera un conjunto difuso  $A$  del tipo 2 definido en un universo  $X$  es caracterizado por una función de membresía  $\mu_A(x)$  como:

$$\mu_A(x): X \rightarrow [0, 1]$$

donde el valor  $\mu_A(x)$  es el grado difuso de membresía y es un conjunto difuso en el intervalo unitario  $[0, 1]$  representado por

$$\mu_A(x) = \sum f(u)/u, \quad u \in [0, 1]$$

donde  $f$  es una función de membresía  $\mu_A(x)$  para el grado difuso y está definida como:

$$f: [0, 1] \rightarrow [0, 1]$$

## 2.3 Operaciones con Conjuntos Difusos.

Las funciones de membresía son obviamente el componente crucial de un conjunto difuso. Por eso no es sorprendente que las operaciones con conjuntos difusos sean definidas via sus funciones de membresía. Lo que a continuación se presenta son los conceptos sugeridos por Zadeh en 1965. Ellos constituyen un marco de trabajo consistente para la teoría de los conjuntos difusos. Pero sin embargo, estas operaciones no son la única forma posible de extender la teoría clásica de conjuntos en una forma consistente. Zadeh y otros autores han sugerido definiciones alternativas o adicionales para éstas operaciones.

La teoría de conjuntos difusos original fue formulada en términos de los siguientes operadores específicos de complemento, unión e intersección de conjuntos:

$$\mu_{A^c}(x) = 1 - \mu_A(x) \quad (1)$$

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (2)$$

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad (3)$$

Nótese que cuando el intervalo de los grados de membresía es restringido al intervalo abierto  $(0, 1)$ , estas funciones se comportan precisamente como los operadores correspondientes a los conjuntos exactos, estableciéndolas así como generalizaciones claras de éstos últimos. Sin embargo, es entendido que estas funciones no son las únicas generalizaciones posibles de los operadores de los conjuntos exactos. Para cada una de las tres operaciones de conjuntos, varias clases diferentes de funciones, las cuales poseen propiedades axiomáticas apropiadas, han sido subsecuentemente propuestas. La teoría de conjuntos difusos que está basada en los operadores dados por las ecuaciones (1) a la (3) son llamadas operaciones estándar de la teoría de conjuntos difusos.

**2.3.1 Complemento Difuso.** Un complemento difuso de un conjunto difuso  $A$  está especificado por una función

$$c: [0,1] \rightarrow [0,1],$$

la cual asigna un valor  $c(\mu_A(x))$  a cada uno de los grados de asociación de  $\mu_A(x)$ . Este valor asignado es interpretado como el grado de asociación del elemento  $x$  en el conjunto difuso que representa la negación del concepto representado por  $A$ . Así, si  $A$  es el conjunto de hombres altos, su complemento es el conjunto difuso de hombres que no son altos. Obviamente, hay muchos elementos que pueden tener algún grado de asociación no cero en el conjunto difuso y en su complemento. Así que para que cualquier función sea considerada un complemento difuso, debe satisfacer al menos los dos siguientes requerimientos axiomáticos.

**Axioma c1.**  $c(0)=1$  y  $c(1)=0$ , esto es,  $c$  se comporta como el complemento ordinario de los conjuntos exactos (**condiciones de frontera**).

**Axioma c2.** Para toda  $a, b \in [0,1]$ , si  $a < b$  entonces  $c(a) < c(b)$ , esto es,  $c$  es monótona no creciente.

Los símbolos  $a$  y  $b$ , usados en este axioma como argumentos de la función  $c$ , representan los grados de asociación de algunos elementos arbitrarios del conjunto universal en un conjunto difuso dado. Por ejemplo,  $a = \mu_A(x)$  y  $b = \mu_A(y)$  para algún  $x, y \in X$  y algún conjunto difuso  $A$ .

Hay muchas funciones que satisfacen los axiomas  $c1$  y  $c2$ . Para cualquier conjunto difuso particular  $A$ , existen diferentes conjuntos difusos que constituyen su complemento, siendo cada uno producido por una función de complemento difuso diferente. Así, para distinguir el complemento que resulta de la aplicación del complemento difuso clásico de la ecuación 1 y estos otros, el complemento es denotado por  $(A)^c$ , lo último, expresado por la función  $c$ , es denotado por  $C(A)$ , en donde:

$$C: \neg P(X) \rightarrow \neg P(X)$$

es una función tal que  $c(\mu_A(x)) = \mu_{C(A)}(x)$  para toda  $x \in X$ .

Todas las funciones que satisfacen los axiomas  $c1$  y  $c2$  forman la clase más general de complementos difusos. Es bastante obvio que la exclusión de uno u otro axioma añadiría a esta clase algunas funciones totalmente inaceptables como complementos. Una violación del axioma  $c1$  incluiría funciones que no conforman el complemento ordinario de los conjuntos exactos. El axioma  $c2$  es esencial, puesto que intuitivamente se espera que un incremento en el grado de asociación en un conjunto difuso, debe resultar ya sea en un decremento o en el caso extremo en no cambiar en el grado de asociación en su complemento. A estos dos primeros axiomas se les conoce como el esqueleto axiomático para los complementos difusos.

En la mayoría de los casos de significado práctico, es deseable considerar varios requerimientos adicionales para los complementos difusos. Cada uno de ellos reduce la clase general de complementos difusos a una subclase especial. Dos de los requerimientos más deseables, los cuales son usualmente listados en la literatura entre los axiomas de complementos difusos, son los siguientes:

**Axioma c3.**  $c$  es una función continua.

**Axioma c4.**  $c$  es involutiva, lo cual significa que  $c(c(a))=a$  para toda  $a \in [0,1]$ .

Las funciones que satisfacen el axioma c3 forman una subclase especial de la clase general de los complementos difusos; aquellos que satisfacen el axioma c4 son necesariamente continuos y, por lo tanto, forman una subclase anidada adicional. El complemento difuso clásico dado por la ecuación (1) está contenido dentro de la clase de complementos involutivos.

Un ejemplo de un complemento difuso que es continuo (axioma c3) pero no involutivo (axioma c4) es la función:

$$c(a) = \frac{1}{2} (1 + \cos \pi a)$$

ya que si  $a=0.33$  entonces  $c(0.33)=0.75$  pero  $c(0.75)=0.15 \neq 0.33$

Una clase de complemento difuso involutivo es la clase Sugeno definida por

$$c_\lambda(a) = (1-a)/(1+\lambda a)$$

en donde  $\lambda \in (-1, \infty)$ . Para cada valor del parámetro  $\lambda$ , se obtiene un complemento difuso involutivo particular.

Otro ejemplo de una clase de complemento difuso involutivo está definido por

$$c_\omega(a) = (1 - a^\omega)/\omega$$

en donde  $\omega \in (0, \infty)$ ; ésta es conocida como la clase Yager de complementos difusos. Cuando  $\omega=1$ , ésta función se convierte en el complemento difuso clásico de  $c(a)=1-a$ .

**2.3.2 Unión Difusa.** La unión difusa de dos conjuntos A y B es especificada en general por una función de la forma

$$u: [0, 1] \times [0, 1] \rightarrow [0, 1]$$

Para cada elemento X en el conjunto universal, esta función toma como su argumento el par consistente de los grados de asociación del elemento en los conjuntos A y B. Así,

$$\mu_{u \cup b}(x) = u[\mu_A(x), \mu_B(x)] = \max[\mu_A(x), \mu_B(x)]$$

donde el término  $\max[\mu_A(x), \mu_B(x)]$  representa el máximo grado de membresía que tiene x en los dos conjuntos A y B.

Cualquier función de esta forma que se califica como una unión difusa, debe satisfacer al menos los siguientes axiomas:

**Axioma u1.**  $u(0,0) = 0$ ;  $u(0,1) = u(1,0) = u(1,1) = 1$ ; esto es, u se comporta como la unión clásica con conjuntos exactos (condiciones de frontera).

**Axioma u2.**  $u(a,b) = u(b,a)$ ; esto es, u es conmutativa.

**Axioma u3.** Si  $a \leq a'$  y  $b \leq b'$ , entonces  $u(a,b) \leq u(a',b')$ ; esto es,  $u$  es *monótona*.

**Axioma u4.**  $u(u(a,b),c) = u(a,u(b,c))$ ; esto es,  $u$  es *asociativa*.

El primer axioma asegura que la función definirá una operación que generalice la unión clásica de conjuntos exactos. El segundo axioma de conmutatividad (o simetría) indica indiferencia al orden en el cual los conjuntos, que serán ordenados, son considerados. El tercer axioma es el requerimiento natural de que un decremento en el grado de asociación en el conjunto A o en el conjunto B no puede producir un incremento en el grado de asociación AUB. Finalmente, el cuarto axioma de asociatividad asegura que se puede tener la unión de cualquier número de conjuntos en cualquier orden deseado de agrupación por pares; este axioma permite entender la operación de unión de conjuntos difusos o más de dos conjuntos.

Es frecuentemente deseable restringir la clase de uniones difusas considerando varios requerimientos adicionales. Dos de los más importantes son expresados por los siguientes axiomas:

**Axioma u5.**  $u$  es una función continua.

**Axioma u6.**  $u(a,a) = a$ ; esto es,  $u$  es *idempotente*.

El axioma de continuidad impide una situación en la cual un muy pequeño incremento en el grado de asociación del conjunto A o en el conjunto B produce un cambio grande en el grado de asociación en A U B. El axioma asegura que la unión de cualquier conjunto consigo mismo produce precisamente el mismo conjunto.

Diferentes clases de funciones han sido propuestas cuyos miembros individuales satisfacen todos los requerimientos axiomáticos para la unión difusa y ninguno, uno o ambos de los axiomas opcionales. Una de esta clase de uniones difusas es conocida como la clase Yager, y está definida por la función:

$$u_{\omega}(a,b) = \min [1, (a_{\omega} + b_{\omega})^{1-\omega}],$$

en donde los valores del parámetro  $\omega$  corresponden al intervalo abierto  $(0, \infty)$ . Esta clase de funciones satisfacen los axiomas de u1 a u5, pero estas funciones no son, en general, idempotentes. Las funciones especiales de esta clase son formadas cuando ciertos valores son seleccionados para el parámetro  $\omega$ .

**2.3.3 Intersección Difusa.** Igual que la unión difusa general de dos conjuntos difusos A y B, la intersección difusa es especificada por la función:

$$i: [0,1] \times [0,1] \rightarrow [0,1]$$

El argumento de esta función es el par consistente del grado de asociación de algún elemento  $x$  en el conjunto difuso  $B$ . La función regresa el grado de membresía del elemento en el conjunto  $A \cap B$ . Así,

$$\mu_{A \cap B}(x) = i[\mu_A(x), \mu_B(x)] = \min[\mu_A(x), \mu_B(x)]$$

donde el término  $\min[\mu_A(x), \mu_B(x)]$  representa el mínimo grado de membresía que tiene  $x$  en los dos conjuntos  $A$  y  $B$ .

Una función de ésta forma debe satisfacer los siguientes axiomas para ser considerada una intersección difusa:

**Axioma i1.**  $i(1,1)=1; i(0,1) = i(1,0) = i(0,0) = 0$ ; esto es,  $i$  se comporta como la intersección clásica con conjuntos exactos (**condiciones de frontera**).

**Axioma i2.**  $i(a,b) = i(b,a)$ ; esto es,  $i$  es **conmutativa**.

**Axioma i3.** Si  $a \geq a'$  y  $b \geq b'$ , entonces  $i(a,b) \geq i(a',b')$ ; esto es,  $i$  es **monótona**.

**Axioma i4.**  $i(i(a,b),c) = i(a, i(b,c))$ ; esto es,  $i$  es **asociativa**.

La justificación para estos axiomas esenciales (el esqueleto axiomático para la intersección de conjuntos difusos) es similar a la dada en la sección anterior para los axiomas requeridos en la unión difusa. Los requerimientos adicionales más importantes para las intersecciones de conjuntos difusos, las cuales son deseables en ciertas aplicaciones, son expresados por los dos siguientes axiomas:

**Axioma i5.**  $i$  es una **función continua**.

**Axioma i6.**  $i(a,a) = a$ ; esto es,  $i$  es **idempotente**.

Una clase de funciones que satisfacen los axiomas del i1 al i4 son las de la clase **Yager**, la cual está definida por la función

$$i_{\omega}(a,b) = 1 - \min[1, ((1-a)^{\omega} + (1-b)^{\omega})^{1/\omega}],$$

en donde los valores del parámetro  $\omega$  corresponden al intervalo abierto  $(0, \infty)$ .

**2.3.4 Suma Inclusiva.** La suma inclusiva  $C=A \oplus B$  es definida como :

$$\mu_{A \oplus B}(x) = \min[1, \mu_A(x) + \mu_B(x)]$$

**2.3.5 Resta inclusiva.** La resta inclusiva  $C = A \ominus B$ , se define de la siguiente manera.

$$\mu_{A \ominus B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}$$

**2.3.6 Suma Algebraica.** La suma algebraica o probabilística de  $C = A + B$ , se define como:

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

**2.3.7 Producto Algebraico.** El producto algebraico de dos conjuntos difusos  $C = A \cdot B$ , se define

como:

$$C = \{(x, \mu_A(x) \cdot \mu_B(x)) \mid x \in X\}$$

### Ejemplo 2.3.1

Sean los siguientes conjuntos difusos  $A = \{(3,0.5), (5,1), (7,0.6)\}$  y  $B = \{(3,1), (5,0.6)\}$  entonces:

$$\begin{aligned} A^c &= 0.5/3 + 0/5 + 0.4/7 \\ A \cup B &= 1/3 + 1/5 + 0.6/7 \\ A \cap B &= 0.5/3 + 0.6/5 + 0.6/7 \\ A \oplus B &= 1/3 + 1/5 + 0.6/7 \\ A \ominus B &= 0.5/3 + 0.6/5 \\ A \cdot B &= 0.5/3 + 0.6/5 \end{aligned}$$

## 2.4 Lógica bivaluada y n-valuada.

La lógica clásica trata con proposiciones que son verdaderas o falsas, las cuales en la lógica proposicional son representadas por variables, conocidas generalmente como variables lógicas. Como cada variable representa una proposición hipotética, puede asumir uno de los varios valores de verdad. En la lógica proposicional es posible generar nuevas variables a partir de un conjunto de variables lógicas existentes. Así, si tomamos en cuenta  $n$  variables lógicas  $v_1, v_2, \dots, v_n$ , entonces se puede definir una nueva variable mediante una función que asigne un valor de verdad particular a la nueva variable para cada combinación de los valores de verdad de las variables dadas. Esta función es generalmente llamada función lógica. Puesto que  $n$  variables lógicas pueden asumir  $2^n$  valores de verdad probables, hay  $(2^2)^n$  funciones lógicas posibles definiendo estas variables. Como puede observarse el número de funciones lógicas crece en una forma considerable por cada nueva variable que se contempla, con la ayuda claro de un pequeño número de funciones simples que

se conocen como primitivas lógicas. Tomando en cuenta únicamente 2 variables lógicas, los siguientes serían ejemplos de funciones lógicas:

$v_2$	1100			
$v_1$	1010	Función	Símbolo	Otro símbolo
$w_1$	0000	CERO	0	F
$w_2$	0001	NOR		
$w_3$	0010	Inhibición		
$w_4$	0011	Negación	$v_2'$	$\neg v_2$
$w_5$	0100	Inhibición		
$w_6$	0101	Negación	$v_1'$	$\neg v_1$
$w_7$	0110	OR-exclusiva		
$w_8$	0111	NAND		
$w_9$	1000	Conjunción	$v_1 \wedge v_2$	$v_1 \& v_2$
$w_{10}$	1001	Bicondicional	$v_1 \Leftrightarrow v_2$	$v_1 \equiv v_2$
$w_{11}$	1010	Identidad	$v_1$	$v_1^1$
$w_{12}$	1011	Implicación	$v_1 \Leftarrow v_2$	$v_1 \subset v_2$
$w_{13}$	1100	Identidad	$v_2$	$v_2^1$
$w_{14}$	1101	Implicación	$v_1 \Rightarrow v_2$	$v_1 \supset v_2$
$w_{15}$	1110	Disyunción	$v_1 \vee v_2$	$v_1 + v_2$
$w_{16}$	1111	UNO	1	T

Dos de los muchos conjuntos de primitivas han predominado en la lógica proposicional: (1) negación, conjunción y disyunción, y (2) negación e implicación. Combinando, por ejemplo, conjunciones y disyunciones (empleadas como primitivas) en cualquier expresión algebraica, conocidas como fórmulas lógicas, se puede formar cualquier otra función lógica.

Por otra parte, cuando la variable representada por una fórmula lógica es siempre verdadera sin importar los valores de verdad asignados a las variables participantes en la fórmula, es llamada tautología; cuando es siempre falsa es llamada contradicción. Las tautologías son importantes por el razonamiento deductivo, puesto que ellas representan las fórmulas lógicas que debido a sus formas, son verdaderas sin importar el fundamento lógico. Varias formas de tautologías pueden ser usadas para hacer inferencias deductivas, y estas son llamadas reglas de inferencia. Ejemplos de algunas tautologías frecuentemente usadas como reglas de inferencia son:

$$(a \wedge (a \Rightarrow b)) \Rightarrow b \quad (\text{modus ponens})$$

$$('b \wedge (a \Rightarrow b)) \Rightarrow a' \quad (\text{modus tollens})$$

$$((a \Rightarrow b) \wedge (b \Rightarrow c)) \Rightarrow (a \Rightarrow c) \quad (\text{silogismo hipotético})$$

Las proposiciones son sentencias expresadas en algún lenguaje. Cada sentencia representa una proposición que puede ser fundamentalmente dividida en un sujeto y un predicado. En otras palabras, una simple proposición puede ser expresada, en general, en la forma canónica:

x está en P

en donde  $x$  es la representación de un sujeto y  $P$  designa un predicado, lo cual caracteriza una propiedad. Por ejemplo, "Austria es un país de habla alemana" es una proposición en la cual "Austria" es un sujeto, y "un país de habla alemana" es un predicado que caracteriza una propiedad específica, la propiedad de que siendo un país cuyos habitantes no son alemanes hablan alemán, y esta proposición es verdadera. Al igual que en la lógica proposicional, es posible producir predicados nuevos a partir de predicados dados por fórmulas lógicas. Estas fórmulas, las cuales son llamadas fórmulas de predicado, son la esencia de la lógica de predicados.

La lógica clásica bivaluada puede ser extendida a lógica trivaluada en varias formas. Varias lógicas trivaluadas cada una con su propia razón, están ahora bien establecidas. Es común en estas lógicas indicar la verdad, falsedad, e indeterminación por 1, 0, y  $\frac{1}{2}$ , respectivamente. Cinco de las lógicas trivaluadas más conocidas, etiquetadas con los nombres de sus originadores, son definidas en términos de estas cuatro primitivas en la siguiente tabla:

a	b	Lukasiewicz				Bochvar				Kleene				Heyting				Reichenbach							
		0	1	$\frac{1}{2}$	1																				
0	0	0	0	$\frac{1}{2}$	1																				
0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$																		
0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$																		
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1												
$\frac{1}{2}$	1	$\frac{1}{2}$	1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1	1	$\frac{1}{2}$												
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$																		
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Una vez que las diferentes lógicas tri-valuadas fueron aceptadas como significativas, surge el reto de explorar las generalidades dentro de las lógicas n-valuadas para un número arbitrario de valores de verdad ( $n \geq 2$ ). Varias lógicas n-valuadas fueron de hecho desarrolladas en 1930. Para cualquier  $n$  dada, los valores de verdad en estas lógicas generalizadas son usualmente etiquetadas por números racionales en el intervalo unitario  $[0,1]$ . Las primeras series de lógicas n-valuadas para las cuales  $n \geq 2$  fueron propuestas por Lukasiewicz cerca del año 1930 como una generalización de su lógica tri-valuada, definiendo las primitivas mediante las siguientes ecuaciones:

$$\begin{aligned}
 a' &= 1 - a, \\
 a \wedge b &= \min(a,b), \\
 a \vee b &= \max(a,b), \\
 a \Rightarrow b &= \min(1, 1+b-a), \\
 a \Leftrightarrow b &= 1 - |a - b|.
 \end{aligned}$$

La lógica estándar de Lukasiewicz es solamente una de una variedad de lógicas infinito-valuadas en el mismo sentido como la teoría de conjuntos difusos estándar es solamente una de una variedad de teorías de conjuntos difusos, las cuales difieren una de otra por el conjunto de operaciones que emplean. Puesto que las distintas lógicas multi-valuadas tienen sus contrapartes en la teoría de conjuntos difusos, ellas forman el muelle de la lógica difusa, esto es, una lógica basada sobre la teoría de los conjuntos difusos. En su escala completa, sin embargo, la lógica difusa es actualmente una extensión de la lógica multi-valuada. De hecho su meta fundamental es proveer fundamentos para aproximar el razonamiento con proposiciones imprecisas usando la teoría de conjuntos difusos como la principal herramienta. El enfoque primario de la lógica difusa es

sobre el lenguaje natural, en donde la aproximación al razonamiento con proposiciones imprecisas es bastante típico. El siguiente silogismo es un ejemplo de la aproximación al razonamiento en términos de lingüística que no puede ser tratado por el predicado de lógica clásica:

**Las monedas antiguas son usualmente poco cobrables**

**Las poco cobrables son usualmente caras**

**Las monedas antiguas son usualmente caras**

**Esta es una inferencia deductiva significativa. Así, al tratar con inferencias como ésta, la lógica difusa permite el uso de predicados difusos, cuantificadores difusos, valores de verdad difusos, etc.**

## CAPITULO 3:

### *Técnicas de Control Difuso y Aplicaciones*

Un sistema de control es un sistema físico que puede comandar o regular a otro sistema o a sí mismo. Siendo así, en nuestro alrededor existen muchos ejemplos que van desde un simple switch eléctrico que controla el flujo de corriente, el conductor que mediante el volante y los pedales puede controlar la velocidad y la dirección de un automóvil, o el caso de un sistema de calefacción que hace uso de un termostato para regular la temperatura de un cuarto, etc. La figura 3.1 muestra un sistema de control típico, el cual puede tener una o más entradas e igual número de salidas. El bloque marcado como "Sistema de Control", representa todo un sistema en el que el proceso de control se lleva a cabo. El tipo de proceso que ocurre puede ser clasificado como de lazo abierto o de lazo cerrado. Un sistema de control de lazo abierto (como el que se muestra en la figura 3.1a), es aquel en el cual la acción de control es independiente de la salida, o en otras palabras, la salida está relacionada únicamente con lo que se obtiene a través de la entrada del sistema. En cambio, un controlador de lazo cerrado (figura 3.1b) difiere del de lazo abierto en que su salida es dependiente tanto de la entrada como de la salida. De esta forma, el resultado que se obtiene del controlador estará en función de las dos magnitudes a través del lazo de realimentación, el cual en algunos casos puede modificar o adaptar la muestra que toma de la salida antes de combinarla con la entrada.



Fig. 3.1 Sistemas de control

Una computadora puede ser claro ejemplo de un sistema de lazo abierto, ya que los números proporcionados mediante el teclado son procesados internamente para producir un resultado. La conducción de un automóvil representa un ejemplo para el caso del sistema de lazo cerrado, el conductor ajusta la velocidad presionando el acelerador o el freno; el lazo de realimentación en este caso es el mismo conductor de acuerdo a la información que recibe del velocímetro.

El método de control más comúnmente usado a la fecha, está basado en el método Proporcional-Integral-Derivativo (PID). Este método realiza control sobre un sistema mediante la lectura de las señales que recibe de los sensores, para después aplicar ecuaciones matemáticas que finalmente producirán la salida del controlador. Típicamente la salida de este tipo de sistemas es usada para controlar actuadores que regulen flujos, como podría ser el caso del vapor para una turbina. Las ecuaciones matemáticas en las que está basado el PID son usualmente complicadas, y pueden consumir bastante tiempo para su implementación y pruebas. De acuerdo a la experiencia, el PID resulta adecuado para procesos donde la dinámica es esencialmente de segundo orden, como también podría ser el caso del movimiento con fricción. Para sistemas más complicados, un controlador PID puede no ser suficiente.

Un método paralelo o alternativo para implementar hoy en día sistemas de control es la lógica difusa, la misma que en los últimos años ha ido ganando terreno sobre los controladores basados en las técnicas tradicionales. Después de haber sido aplicado en dispositivos electrónicos de consumo y del hogar, el control difuso ha tenido una incursión exitosa en el área industrial, lográndose controlar procesos que vistos desde la posición de los controladores proporcionales, resultaban sumamente complejos. A diferencia de los controladores basados en modelos matemáticos como los mencionados anteriormente, los basados en lógica difusa operan en forma muy similar a como lo haría un operador experto o un ingeniero.

Mientras un sistema PID requiere de un modelo matemático que describa la dinámica del proceso para en base a éste producir resultados calculados, el controlador difuso emplea los conocimientos y experiencias que se obtienen a través de una persona que conozca a fondo el proceso. La descripción lingüística del sistema que se obtiene por parte de este experto, va formando una base de conocimiento, la cual sustituye al modelo matemático. El método que utiliza la lógica difusa para implementar esta base de datos, es la regla del tipo Si ..... Entonces ( ó IF ..... THEN), con la cual están bastante familiarizadas aquellas personas que hayan manejado algún lenguaje de programación.

Una vez que se tiene construida la base de conocimientos, los datos que entran al controlador son comparados con los antecedentes de las reglas, es decir, con la información contenida en la parte condicional (la parte SI). Cuando un dato de entrada coincide con algún antecedente, se dice que se ha disparado una regla, y se toma la conclusión ( información de la parte Entonces) para construir lo que será la salida del controlador. En la mayoría de los casos, más de una regla llega a ser disparada debido a la multiplicidad de conclusiones que se pueden generar a partir de un suceso. La abundancia de información acerca de las distintas etapas del sistema obviamente incrementa el tamaño de la base de conocimientos, pero eso repercute en una mejora global de los resultados que arroja el controlador difuso, ya que al igual que en el caso de un operador, mientras más experiencias y conocimientos tenga sobre su tarea, mejor será su rendimiento. Este tipo de control basado en el conocimiento, por lo general se utiliza en sistemas en los que una buena aproximación es más suficiente, pero existen algunas otras aplicaciones en las que se requiere un tipo de control más preciso, y estos casos pueden llegar a representar la excepción para la aplicabilidad del método difuso.

Por otra parte y en principio, los controladores difusos son similares en su estructura a los sistemas expertos, esto se debe principalmente a que al igual que un sistema experto, el controlador basado en lógica difusa intenta modelar las formas de razonamiento del ser humano, para en este caso realizar operaciones de control. Básicamente son seis los módulos que conforman al controlador difuso, tal y como se observa en la figura 3.2. A continuación se describirán brevemente cada una de estas seis etapas, y se proporcionan algunas técnicas para su diseño.

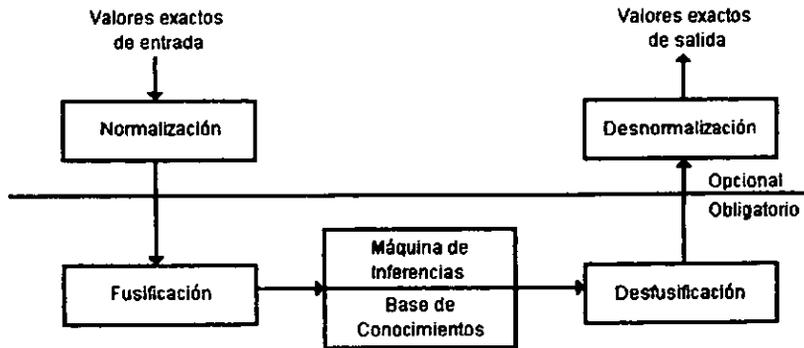


Fig. 3.2 Estructura básica de un controlador difuso

### 3.1 Proceso de Escalamiento o Normalización.

Las señales provenientes de los sensores son digitalizadas e introducidas directamente a este módulo; el rango de valores digitales que se presenta, puede variar en cada sensor. Por lo tanto resulta conveniente sustituir este rango original por uno normalizado, dentro del cual se encuentren definidas las funciones de membresía. Los factores de escalamiento que describen la normalización de las entradas, así como la desnormalización de las salidas, juegan un rol similar al de los coeficientes de ganancia en un controlador convencional. En otras palabras, tales factores se encuentran involucrados con la eficiencia y la estabilidad del controlador, y pueden llegar a ser fuente de posibles inestabilidades, oscilaciones y efectos de deterioro de amortiguamiento.

Existen básicamente dos formas de determinar los factores de escala: (1) heurística y (2) analíticamente. La primera forma sigue el método de prueba y error, para lo cual se toman valores supuestos de *overshoot* (*OV*), *rise-time* (*RT*), y *amplitud de oscilaciones* (*OSC*); observando el comportamiento de estos parámetros se realizan los ajustes correspondientes en los factores de escalamiento, los cuales pueden ser realizados en forma automática mediante reglas. Durante el proceso, puede llegar a ser necesario un reajuste de factores de escala, por lo que estos pueden ser actualizados en cada iteración del proceso. Estas iteraciones continúan hasta que se obtienen los valores deseados.

Las reglas heurísticas pueden ser explicadas como sigue. Al cambiar los factores de escala tanto de la señal de error  $e$ , como del incremento de error  $\Delta e$ , estamos en efecto cambiando los pesos dados a estas variables de proceso (variables de entrada). Por ejemplo, si la respuesta del proceso es más lenta que la deseada (si  $\Delta RT$  es positivo), entonces necesitamos incrementar el efecto del error en el proceso, así el factor de escalamiento  $N_e$  es incrementado. De igual manera, si el *overshoot* o la oscilación son de valores más altos de lo deseado, entonces necesitamos incrementar el efecto del error, así  $N_{\Delta e}$  aumenta su valor. La segunda forma para la derivación de los factores de escala apunta a establecer una relación analítica entre los valores de los factores de escala y la conducta de lazo cerrado del proceso controlado. En este caso, se asume que existe un modelo convencional del proceso bajo control.

Una vez determinados los valores de los factores de escala tanto de las entradas como de las salidas, se pueden almacenar en una matriz de escalamiento, lo cual facilita su manejo en la implementación del software.

$$\begin{bmatrix} N_{I1} \\ N_{I2} \\ N_{I3} \\ \vdots \\ N_{In} \end{bmatrix} \quad \begin{bmatrix} N_{O1} \\ N_{O2} \\ N_{O3} \\ \vdots \\ N_{Om} \end{bmatrix}$$

$N_{In}$  = Factor de normalización de entradas

$N_{Om}$  = Factor de normalización para las salidas

### 3.2 Difusificación.

La información digitalizada de los transductores, antes y después del proceso de normalización, se encuentra expresada en forma exacta, por lo que en esa condición no puede ser interpretada ni manejada difusamente. La principal tarea de la etapa de difusificación es entonces convertir estos valores exactos en valores difusos, con el fin de hacerlos compatibles con la representación de conjuntos difusos. Bajo la difusificación, las funciones de membresía definidas para las variables de entrada, son aplicadas sobre los valores exactos ya normalizados para determinar su grado de membresía al conjunto. Este proceso se realiza tal como se muestra en la figura 3.3 en la que el valor " $x_i$ " ya normalizado es difusificado al mapearlo con la función de membresía. Con lo que respecta al universo de discurso (conjunto de valores en el eje de las abscisas) en base al cual se definen los conjuntos difusos, éste se determina en base al conocimiento experto que se tiene sobre la variable que se maneja. Si tomamos como ejemplo la variable lingüística *Temperatura*, la cual tiene como valores los conjuntos difusos "Alta", "Media", y "Baja". Entonces en base al conocimiento que se tiene sobre el proceso, se podría definir a la temperatura Alta en un rango de {50 - 80°C}, mismo que posteriormente puede ser normalizado. De la misma forma pueden definirse los universos de discurso restantes.

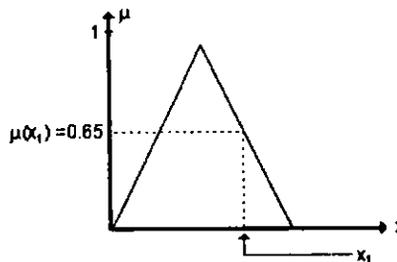


Fig. 3.3 Conjunto difuso

Existen algunos aspectos importantes que deben mencionarse acerca de las funciones de membresía. A continuación se detallan algunos de estos aspectos, buscando sobre todo presentar información útil para su diseño.

**3.2.1 Forma de las funciones de membresía.** Para representar un conjunto difuso electrónicamente, necesitamos definir su función de membresía. Concretamente la selección de la forma que tendrá esta función es un punto importante a considerar. Con fines de simplificar las rutinas de programación y hacer más rápida su ejecución, se consideran generalmente 3 tipos de funciones: la triangular, la trapezoidal o la acampanada. Cualquiera de estas tres funciones puede ser almacenada con un mínimo uso de memoria, y manipulada eficientemente en términos de requerimiento de tiempo real por la máquina de inferencias. Las siguientes figuras describen las funciones mencionadas.

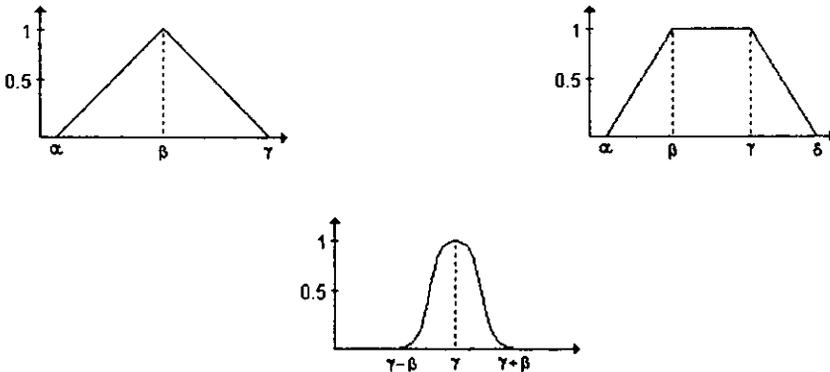


Fig 3.4 Funciones de membresía más comunes

Como se puede ver, la más fácil de representar y por lo tanto de trabajar, es la triangular, lo cual explica el predominante uso que se le da. Después de seleccionar el tipo de función, se mapean los valores de la variable lingüística en el eje correspondiente. Por ejemplo, si tratáramos de representar distintos conjuntos difusos relacionados con una cantidad de error en el rango  $\varepsilon = [-6, +6]$ , entonces sería como se muestra en la siguiente figura:

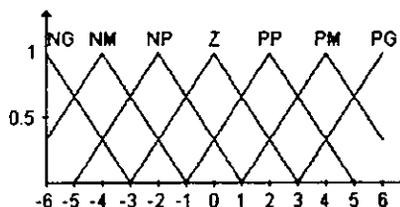


Fig. 3.5 Representación en conjuntos difusos de la variable lingüística error

**3.2.2 Parámetros importantes de las funciones de membresía.** Existen algunos parámetros que caracterizan a una función de membresía. Para mostrar estos parámetros nos basaremos únicamente en las funciones de forma triangular.

**Valor Pico.** Sea  $M$  una variable lingüística y sea  $LM$  un valor específico de  $M$  representado por la función  $\mu_{LM} : M \rightarrow [0,1]$  donde  $M$  es el dominio de  $M$ . El valor pico de  $\mu_{LM}$  es entonces un valor del dominio para el cual la función de membresía alcanza un valor de 1. En el caso de que la función de membresía sea de forma trapezoidal, estaremos hablando entonces de un intervalo.

**Ancho Izquierdo y Derecho.** El ancho izquierdo de  $\mu_{LM}$  es la longitud del intervalo izquierdo que forman el valor pico y el valor del dominio para el cual la función de membresía toma un valor de cero. De forma similar, el ancho derecho es la longitud del intervalo derecho que forman el valor pico y el valor del dominio para el cual la función de membresía toma un valor de cero. La suma del ancho derecho y del ancho izquierdo forman el intervalo para el cual está definido el soporte de  $\mu_{LM}$ . Si el ancho izquierdo es igual al ancho derecho, entonces la función de membresía es simétrica.

**Puntos de cruce.** Sea  $\mu_{LM1}$  y  $\mu_{LM2}$  dos funciones de membresía representando dos diferentes valores lingüísticos de  $M$ , los cuales son elementos del conjunto  $LM$ . Un punto de cruce entre  $\mu_{LM1}$  y  $\mu_{LM2}$  es aquel valor  $x_{CRUCE}$  definido en  $M$  tal que  $\mu_{LM1}(x_{CRUCE}) = \mu_{LM2}(x_{CRUCE}) > 0$ . Además la etiqueta del punto de cruce es el grado de membresía  $\mu_{LM}(x_{CRUCE})$ . Esto significa que cada valor exacto 'x' pertenezca por lo menos a una función de membresía con un grado estrictamente mayor que cero. Si éste no es el caso, entonces este valor de entrada no podrá ser fusificado. Así, ninguna de las reglas será disparada en consecuencia, y no será calculado ningún valor para las variables de salida del controlador, lo que obviamente produce una discontinuidad en el sistema.

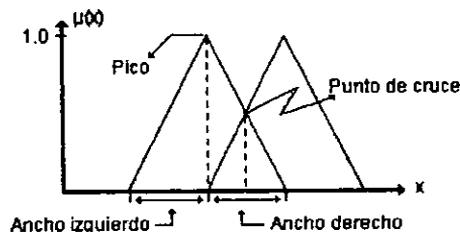


Fig. 3.6

Por otro lado, un excesivo traslapamiento puede también ser perjudicial para el sistema, ya que cuando más de dos funciones de membresía se traslapan, se produce una especie de amortiguamiento en la superficie de control. En la figura 3.7 se muestra el efecto de amortiguamiento de la entrada 'x' sobre la salida 'u', el cual puede ocurrir cuando existen más de dos conjuntos difusos traslapados sobre un universo de discurso. Los círculos vacíos representan los resultados numéricos de las reglas individuales en el caso del traslapamiento sencillo. Los círculos rellenos representan la salida numérica del controlador

difuso cuando el valor del soporte de por lo menos una regla es 1 y existe traslapamiento de más de dos funciones de membresía.

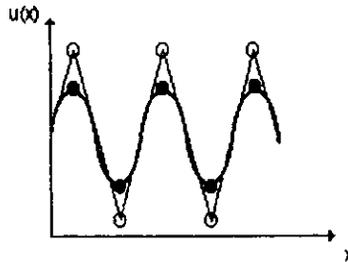


Fig. 3.7

**3.2.3 Número y posición de las Funciones de Membresía.** Cuando se diseña un controlador difuso una de las primeras dudas que surgen, después de elegir las entradas, es cuántas funciones son necesarias y cómo deberían ser divididas éstas en el universo de discurso de las entradas. El uso de un tipo de función lineal (fig. 3.8 a) o de una no lineal (fig. 3.8 b), depende del problema con el que se trate.

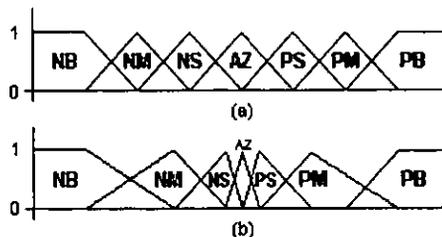


Fig. 3.8 (a) función de membresía lineal; (b) función de membresía no lineal.

Mientras más alta sea la densidad de una función de membresía en una cierta región del universo de discurso, mejor se pueden obtener controladores con grados de complejidad elevados. Actualmente, la elección del número de funciones de membresía y la forma como son divididas a lo largo del universo de discurso, requiere de un conocimiento acerca de cómo la salida del controlador estaría relacionada con la(s) entrada(s). Por ejemplo, el diseño de un controlador difuso para un proceso no lineal, requiere conocimiento acerca de la no linealidad del proceso. Por otra parte, no existen recetas de cocina con las cuales se pueda determinar tanto el número como la posición de las funciones de membresía, en todo caso, se comienza el diseño y se afina en las partes finales del mismo, eliminando tanto reglas como funciones innecesarias.

**3.2.4 Modificadores (Hedges).** En las anteriores secciones se describió como capturar y representar términos lingüísticos cualitativamente a través del uso de conjuntos difusos. En conversaciones normales, los humanos pueden incorporar incertidumbre adicional a una

expresión mediante el uso de adverbios tales como *muy*, *ligeramente*, *algo*, *mucho* *muy*, etc. Un adverbio es una palabra que modifica a un verbo, a un adjetivo, a otro adverbio o a una sentencia completa. Consideremos por ejemplo un adverbio que modifica a un verbo, "La persona es muy alta". Si necesitamos representar este nuevo conjunto difuso, podríamos partir de la información existente de un conjunto como "La persona es alta", ya que existen técnicas disponibles para trabajar con un conjunto difuso existente para capturar el impacto de un adverbio adicionado, y ésta es precisamente la tarea de los Hedges o Modificadores.

Un Hedge modifica matemáticamente un conjunto difuso existente para de esa forma representar el efecto del adverbio. Por ejemplo, la figura 3.9 muestra conjuntos difusos que representan valores de la variable lingüística "Altura", junto con los conjuntos ajustados mediante la incorporación del adverbio "muy". A continuación se describen algunos hedges que son bastante empleados en la representación de adverbios.

- **Concentración (*muy*).** La operación de concentración tiene el efecto de reducir más los valores de membresía de aquellos elementos que tienen valores de membresía más pequeños. Esta operación está dada como:

$$\mu_{\text{CON}(A)}(x) = (\mu_A(x))^2$$

Dado un conjunto difuso "*personas altas*", podríamos usar esta operación para crear el conjunto "*personas muy altas*".

- **Dilatación (*algo*).** La operación de dilatación expande a los elementos del conjunto difuso mediante el incremento del valor de membresía de aquellos elementos con valores pequeños. Esta operación está dada como:

$$\mu_{\text{DIL}(A)}(x) = (\mu_A(x))^{0.5}$$

Dado un conjunto difuso "*personas medianas*", podríamos usar esta operación para crear el conjunto de "*personas más o menos medianas*".

- **Intensificación (*verdaderamente*).** La operación de intensificación tiene el efecto de intensificar el significado de la frase mediante el incremento de los valores de membresía que están por encima de 0.5, y decrementando aquellos que están por debajo de 0.5. Esta operación se da como:

$$\begin{aligned} \mu_{\text{INT}(A)}(x) &= 2(\mu_A(x))^2 && \text{para } 0 \leq \mu_A(x) \leq 0.5 \\ &= 1-2(1-\mu_A(x))^2 && \text{para } 0.5 < \mu_A(x) \leq 1 \end{aligned}$$

Dado un conjunto difuso de "*personas medianas*", podríamos usar ésta operación para crear el conjunto de "*personas verdaderamente medianas*".

- **Potencia (*muy muy*).** La operación potencia es una extensión de la operación de concentración.

$$\mu_{\text{POW}(A)}(x) = (\mu_A(x))^n$$

Dado un conjunto difuso de "*personas altas*", podríamos usar esta operación con  $n=3$  para generar el conjunto "*personas muy muy muy altas*".

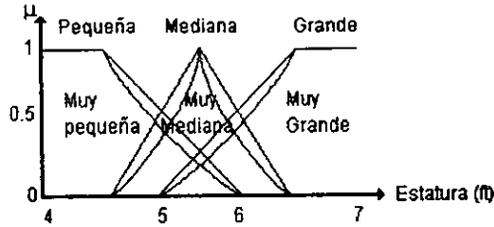


Fig. 3.9 Conjuntos difusos de Altura con el modificador "Muy"

Como ya se mencionó, mediante el uso de modificadores, es posible generar una variedad adicional de conjuntos difusos aparte de los existentes. Asumiendo por ejemplo que tenemos un conjunto difuso  $A$  de las *personas altas*, podríamos derivar un conjunto  $B$  de las *personas no muy altas*, a partir de la siguiente operación:

$$\mu_B(x) = 1 - \mu_A(x)^2$$

Para extender esta idea, asumamos que se tiene los conjuntos difusos  $A$  de *personas altas* y  $B$  de *personas pequeñas*. Podríamos derivar un conjunto  $C$  de *personas no muy altas y personas no muy pequeñas* a partir de la siguiente operación:

$$\mu_C(x) = [1 - (\mu_A(x))^2] \wedge [1 - (\mu_B(x))^2]$$

En general podemos utilizar los operadores difusos así como los hedges para derivar conjuntos difusos que representan varias descripciones lingüísticas y combinaciones de sentencias encontradas en nuestro lenguaje natural.

### 3.3 Base de reglas.

La función básica de la base de reglas es representar en una forma estructurada el sistema de control de un operador experimentado en forma de un conjunto de reglas de producción tales como:

*Si <condición del proceso> Entonces <salida de control>*

La parte *Si* de tal regla se conoce como 'antecedente' y es una descripción de las condiciones del proceso en términos de una combinación lógica de proposiciones difusas. La parte *Entonces* es llamada 'consecuencia' e igualmente es una descripción de la salida de control en términos de una combinación lógica de proposiciones difusas. Los parámetros de diseño involucrados en la construcción de la base de reglas incluye:

- ◆ Selección de las variables de proceso (variables de entrada) y las variables de control (variables de salida).
- ◆ Selección de los contenidos de los antecedentes y consecuencias.
- ◆ Selección del 'term-set' (dominio de las variables lingüísticas).

El term-set de una variable lingüística consiste de un número finito de valores expresados verbalmente. Estos valores lingüísticos, son generalmente expresados en duplas de la forma <signo, magnitud>, por ejemplo, <positivo grande>, <negativo medio>, etc.

Por lo que respecta a la implementación de las reglas, existen tres formas principales para tal tarea, las cuales se complementan una con otra, pareciendo que una combinación de ellas sería necesaria para construir eficientemente el conjunto de reglas.

- 1) Esta primer forma es una de las más utilizadas, y está basada en el conocimiento adquirido mediante la experiencia del operador y/o el ingeniero de control. Esta aproximación es realizada usando dos tipos de técnicas.

- A. Una verbalización introspectiva del conocimiento.
- B. Utilización de un cuestionario cuidadosamente diseñado.

Ambas técnicas ayudan en la obtención de una versión prototipo de la base de reglas, dando por consecuencia la necesidad de realizar posteriormente un ajuste de las funciones de membresía.

- 2) La segunda forma utiliza una descripción lingüística, vista como un modelo difuso del proceso bajo control. Basados en este modelo, se derivan ya sea el conjunto de reglas o una relación difusa explícita que describa al controlador difuso.
- 3) La tercera forma se basa en la existencia de un modelo convencional del proceso, usualmente uno no lineal.

Existen varias aplicaciones en las que se presentan reglas con antecedentes y conclusiones simples, pero en la mayoría de los casos se tiene la necesidad de trabajar con reglas que cuenten con más de un antecedente o más de una consecuencia. Para este caso, se hace necesaria la utilización de conectores lógicos tales como "AND" u "OR" para estructurar las reglas.

**3.3.1 Reglas difusas tipo Mamdani.** Este tipo de reglas fue usado en la primera aplicación de control difuso y tienen la siguiente forma general:

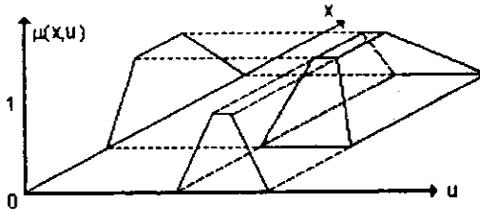
$$r_j: \text{Si } x_1 \text{ es } X_{i_1} \text{ Y } \dots \text{ Y } x_{N_x} \text{ es } X_{i_{N_x}} \text{ Entonces } u \text{ es } U_j$$

- con :
- $N_x$  : número de entradas de control
  - $x_i$  :  $i$ -ésima entrada del controlador
  - $X_{i_j}$  : etiqueta difusa para la  $i$ -ésima entrada en la  $j$ -ésima regla
  - $u$  : salida del controlador
  - $U_j$  : etiqueta difusa para la salida del controlador en la  $j$ -ésima regla

La consecuencia de este tipo de reglas es algo simbólico, por ejemplo, "la salida de control es grande". Teóricamente una regla difusa puede ser representada por una relación  $R_j$ , la cual es el producto cartesiano de las condiciones en las premisas y las conclusiones de la regla difusa:

$$R_j = \{X_{i_1} \times \dots \times X_{i_{N_k}}\} \times U_i$$

La siguiente figura muestra un ejemplo de un conjunto difuso tridimensional que representa una regla difusa, usando el operador Min para la conjunción. Usando las reglas tipo Mamdani en un controlador difuso, tendremos como resultado una salida difusa como resultado de la inferencia difusa, la cual debe ser desdifusificada para obtener una salida numérica del controlador.



3.10 Representación de una regla difusa por medio de una relación difusa

**3.3.2 Reglas difusas tipo Sugeno-Takagi.** La forma general de otro tipo de reglas, conocidas como tipo Sugeno, debido a la introducción de éstas por Takagi y Sugeno (1983) es como sigue :

$$r_j: \text{Si } x_1 \text{ es } X_{i_1} \text{ Y } \dots \text{ Y } x_{N_k} \text{ es } X_{i_{N_k}} \text{ Entonces } u \text{ es } f_i(x_1, \dots, x_{N_k})$$

la cual muestra que la consecuencia de una regla difusa es una función ( $f_i$ ) de las entradas del controlador. Una función lineal en la consecuencia simplificaría este tipo de reglas como:

$$r_j: \text{Si } x_1 \text{ es } X_{i_1} \text{ Y } \dots \text{ Y } x_{N_k} \text{ es } X_{i_{N_k}} \text{ Entonces } u \text{ es } a_{i_1}x_1 + \dots + a_{i_{N_k}}x_{N_k} + b_i$$

con parámetros constantes  $a_{i_j}$  y  $b_i$ , los cuales pueden ser más simplificados cuando los parámetros  $a_{i_j}$  son 0, se tiene una consecuencia numéricamente constante  $b_i$ . Este tipo de reglas han sido utilizadas con éxito en el control de un carro modelo, los resultados fueron reportados por Nishida (1985) y Sugeno y Murakami (1985). La salida de un controlador difuso que utiliza este tipo de reglas es en efecto un conjunto de salidas numéricamente valoradas, por lo que no es necesario utilizar ningún método de desdifusificación.

### 3.4 Inferencia Difusa.

La lógica difusa trata a un conjunto difuso como una proposición. Una proposición difusa es un enunciado que afirma un valor para alguna variable lingüística dada tal como "Altura Media". En general, podemos representar una proposición difusa como:

$$\text{Proposición: } X \text{ es } A$$

donde A es un conjunto difuso definido en el universo de discurso X. Una relación difusa relaciona dos proposiciones de la siguiente forma:

## SI X es A ENTONCES Y es B

Esta regla establece una relación o asociación entre las dos proposiciones. Los sistemas de control difuso almacenan reglas como asociaciones difusas. Esto es, para la regla anterior, donde A y B son conjuntos difusos, el sistema almacena la asociación (A,B) en una matriz M conocida como *matriz asociativa difusa* (ó FAM: Fuzzy Associative Memory). Esta matriz mapea el conjunto difuso A al conjunto difuso B.

Como técnica de razonamiento inexacto, la inferencia difusa intenta establecer un grado de confianza en la parte ENTONCES de la regla, dada una evidencia válida en su premisa. Sin embargo, ya que las proposiciones contenidas en una regla difusa no son otra cosa que conjuntos difusos, la lógica difusa debe mapear la información contenida en los conjuntos de la premisa con la información de los conjuntos pertenecientes a las conclusiones. Para realizar ésto, la inferencia difusa establece un conjunto inducido a partir de la información de un conjunto difuso relacionado. Consideremos por ejemplo los conjuntos A = "Estatura grande" y otro B= "Peso grande", relacionados mediante una regla del tipo SI A ENTONCES B.

SI Estatura es grande ENTONCES Peso es grande

Nosotros podemos representar tanto a A como a B en la forma de vectores y capturar su relación en la memoria asociativa difusa M. En la práctica, nosotros habríamos podido usar esta regla para crear un grado de confianza de que algunas personas de una altura grande son pesadas. La inferencia difusa hace esto tomando la información disponible de altura codificada en A' (un subconjunto de A) e infiere un conjunto difuso B' en B que cualitativamente capture este grado de confianza. Para derivar el conjunto difuso inducido, la inferencia difusa se apoya en la multiplicación difusa de un *vector-matriz*. En la práctica se utilizan dos técnicas de inferencia difusa, conocidas como *inferencia max-min* e *inferencia max-producto*. Antes de presentarlas, es recomendable entender primero la multiplicación difusa vector-matriz.

**Multiplicación Difusa Vector-matriz.** En la multiplicación clásica vector-matriz, podemos generar un vector 'y' dado un vector 'x' y una matriz A mediante:

$$x \cdot A = y$$

$\begin{matrix} \text{bn} & \text{mp} & \text{bp} \end{matrix}$

$$y_j = \sum_{i=1}^n x_i a_{ij}$$

La multiplicación difusa vector-matriz usa una técnica conocida como *composición Max-Min*, definida por el operador composición "o". Este operador realiza una operación max-min sobre una matriz y un vector dados. La operación es similar a la clásica multiplicación vector-matriz, sin embargo, reemplazamos la operación de multiplicación por la operación difusa *min* y la operación suma por la operación difusa *max*.

Ahora considerese la siguiente operación en donde A es un conjunto difuso definido en X y B un conjunto difuso definido en Y. Para los vectores renglón A y B representados como:

$$A = (a_1, a_2, \dots, a_n); \quad a_i = \mu_A(x_i)$$

$$B = (b_1, b_2, \dots, b_p); \quad b_j = \mu_B(x_j)$$

podemos definir un matriz difusa de orden  $n \times p$  tal que

$$A \circ M = B$$

y calcular el componente  $b_j$  mediante la siguiente expresión:

$$b_j = \max [\min(a_i, m_{ij})] \quad (3.1)$$

A manera de ejemplo asumamos  $A = (.2 \ .4 \ .6 \ 1)$  y la matriz difusa  $M$

$$\begin{bmatrix} .1 & .6 & .8 \\ .6 & .8 & .6 \\ .8 & .6 & .5 \\ 0 & .5 & .5 \end{bmatrix}$$

de la ecuación (3.1), podemos calcular  $B$  como

$$\begin{aligned} b_1 &= \max[\min(.2, .1), \min(.4, .6), \min(.6, .8), \min(1, 0)] \\ &= \max(.1, .4, .6, 0) \\ &= 0.6 \end{aligned}$$

$$\begin{aligned} b_2 &= \max(.2, .4, .6, .5) \\ &= 0.6 \end{aligned}$$

$$\begin{aligned} b_3 &= \max(.2, .4, .5, .5) \\ &= 0.5 \end{aligned}$$

**3.4.1 Inferencia Max-Min.** En la inferencia Max-Min el operador de implicación usado es el operador "min". Esto es,

$$m_{ij} = \min(a_i, b_j) \quad (3.2)$$

Dados dos conjuntos difusos  $A$  y  $B$ , podemos usar la ecuación (3.2) para formar la matriz  $M$ . Posteriormente podemos utilizar la ecuación (3.1) para determinar el vector  $B'$  inducido de un subconjunto  $A'$  de  $A$ . A manera de ilustración, asumamos que tenemos un universo de discurso definido en  $X$ , el cual representa ciertos valores de "temperatura". También asumamos que tenemos otro universo de discurso definido en  $Y$  que representa "velocidad", y un conjunto difuso  $B$  definido en  $Y$  que representa "velocidad media". Finalmente asumamos que tenemos la siguiente regla:

SI Temperatura es normal ENTONCES Velocidad es media

o

SI A ENTONCES B

Además asumamos que los conjuntos difusos son representados por los siguientes vectores, donde por sencillez, los elementos del vector son mostrados con sus correspondientes valores de dominio:

$$\begin{aligned} \text{"temperatura normal"} &= (0/100, .5/125, 1/150, .5/175, 0/200) \\ \text{"velocidad media"} &= (0/10, .6/20, 1/30, .6/40, 0/50) \end{aligned}$$

Comenzamos por formar la matriz M de acuerdo a la función (3.2):

$$M = m_{ij} = \min(a_i, b_j)$$

$$M = \begin{bmatrix} \min(0,0) & \min(0,.6) & \min(0,1) & \min(0,.6) & \min(0,0) \\ \min(.5,0) & \min(.5,.6) & \min(.5,1) & \min(.5,.6) & \min(.5,0) \\ \min(1,0) & \min(1,.6) & \min(1,1) & \min(1,.6) & \min(1,0) \\ \min(.5,0) & \min(.5,.6) & \min(.5,1) & \min(.5,.6) & \min(.5,0) \\ \min(0,0) & \min(0,.6) & \min(0,1) & \min(0,.6) & \min(0,0) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & .5 & .5 & .5 & 0 \\ 0 & .6 & 1 & .6 & 0 \\ 0 & .5 & .5 & .5 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Después, asumamos que el subconjunto A' es el siguiente:

$$A' = (0, 100, .5/125, 0/150, 0/175, 0/200)$$

Este subconjunto representa una lectura exacta de temperatura de 125 grados. Esta medición corresponde a un valor de membresía de 0.5 para el conjunto difuso "temperatura normal". Esto induce un conjunto difuso B' que nosotros posteriormente desearemos determinar. Con A' = (0/100, 0.5/125, 0/150, 0/175, 0/200), entonces a través de la composición max-min tenemos

$$b'_j = \max(\min(a'_i, m_{ij}))$$

$$\begin{aligned} b'_1 &= \max\{\min(0,0), \min(.5,0), \min(0,0), \min(0,0), \min(0,0)\} \\ b'_2 &= \max\{\min(0,0), \min(.5,.5), \min(0,.6), \min(0,.5), \min(0,0)\} \\ b'_3 &= \max\{\min(0,0), \min(.5,.5), \min(0,1), \min(0,.5), \min(0,0)\} \\ b'_4 &= \max\{\min(0,0), \min(.5,.5), \min(0,.6), \min(0,.5), \min(0,0)\} \\ b'_5 &= \max\{\min(0,0), \min(.5,0), \min(0,0), \min(0,0), \min(0,0)\} \end{aligned}$$

$$B' = (0/10, .5/20, .5/30, .5/40, 0/50)$$

En efecto, este conjunto difuso inducido es una versión recortada de B, cuya altura es definida por A'. Este, es el efecto general de la inferencia max-min, como se muestra en la siguiente figura para conjuntos difusos de forma triangular. Un punto clave a notar de este ejemplo es el resultado que obtuvimos limitando a A' a un simple valor, es decir, establecimos que nuestra lectura de temperatura fue 125 grados, lo cual nos dio un vector A'=(0.5 0 0 0) que resultó en un B'=(0.5 .5 .5 0).

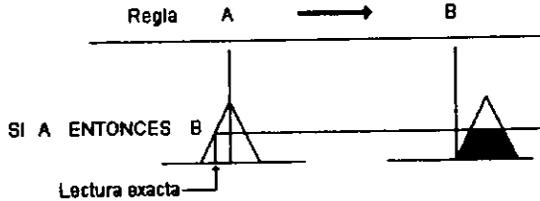


Fig. 3.11 Inferencia Max-min

En la mayoría de las aplicaciones reales de los sistemas de lógica difusa, tenemos un valor exacto sobre alguna medida (por ejemplo  $x=125$  grados). Con un simple valor medido  $x$ , podemos usar  $\mu_A(x)$  directamente con la representación difusa de B, nombrándola  $\mu_B(y)$ , para obtener el conjunto difuso inducido sobre B':

$$B' = \mu_A(x) \wedge \mu_B(y)$$

Por ejemplo, si nuevamente tomamos la temperatura 125 °C,  $\mu_A = 0.5$  y además,

$$B' = [\min(.5, 0), \min(.5, .6), \min(.5, 1), \min(.5, .6), \min(.5, 0)] \\ = (0.5, .5, .5, 0)$$

Como se puede ver, este es el mismo resultado obtenido anteriormente trabajando con la matriz asociativa difusa. Por lo tanto, cuando la información de entrada es exacta (no difusa), no tenemos que calcular y manejar matrices difusas, sino que podemos simplemente trabajar con la menor cantidad posible de información difusa.

En el caso de que la información de entrada a una regla represente una lectura difusa, podemos continuar tomando una simple aproximación. Considerando la regla IF A THEN B, y una lectura difusa de A designada como A'. Podemos simplemente tomar la intersección de los dos conjuntos como nuestra entrada,  $\min(a_i, a'_i)$ , para inducir el conjunto difuso B'. Esto se ilustra en la siguiente figura.

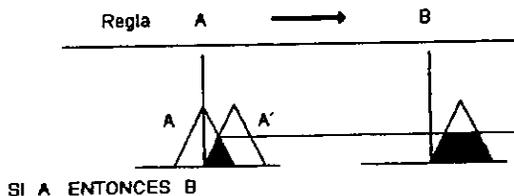


Fig. 3.12 Inferencia Max-min para una entrada difusa.

**3.4.2 Inferencia Max-Producto.** La inferencia max-producto, usa el producto estándar como el operador implicación cuando se forman los componentes de M:

$$m_{ij} = a_i b_j$$

Siguiendo los cálculos de esta matriz, la composición max-min es usada para determinar la matriz inducida B' a partir de algún subconjunto vector A'. Para ilustrar esta técnica de inferencia consideramos el mismo vector usado en la sección anterior:

$$A = (0, .5, 1, .5, 0) \quad B = (0, .6, 1, .6, 0)$$

$$M = \begin{bmatrix} (0,0) & (0,6) & (0,1) & (0,6) & (0,0) \\ (.5,0) & (.5,.6) & (.5,1) & (.5,.6) & (.5,0) \\ (1,0) & (1,.6) & (1,1) & (1,.6) & (1,1) \\ (.5,0) & (.5,.6) & (.5,1) & (.5,.6) & (.5,0) \\ (0,0) & (0,6) & (0,1) & (0,6) & (0,0) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & .3 & .5 & .3 & 0 \\ 0 & .6 & 1 & .6 & 0 \\ 0 & .3 & .5 & .3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Asumiendo nuevamente A' = (0, .5, 0, 0, 0), entonces a través la composición max-min tenemos:

$$b'_j = \max(\min(a'_i, m_{ij}))$$

$$b'_1 = \max\{\min(0,0), \min(.5,0), \min(0,0), \min(0,0), \min(0,0)\}$$

$$b'_2 = \max\{\min(0,0), \min(.5,.3), \min(0,.6), \min(0,.3), \min(0,0)\}$$

$$b'_3 = \max\{\min(0,0), \min(.5,.5), \min(0,1), \min(0,.5), \min(0,0)\}$$

$$b'_4 = \max\{\min(0,0), \min(.5,.3), \min(0,.6), \min(0,.5), \min(0,0)\}$$

$$b'_5 = \max\{\min(0,0), \min(.5,0), \min(0,0), \min(0,0), \min(0,0)\}$$

$$B' = (0, .3, .5, .3, 0)$$

La técnica de inferencia max-producto produce una versión escalada de B. La siguiente figura ilustra el resultado general del uso de esta técnica por conjuntos difusos triangulares.

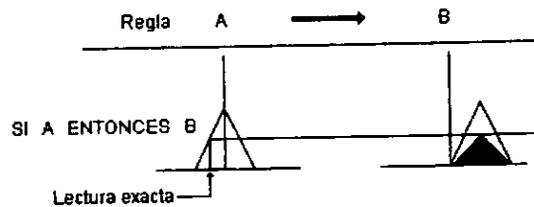


Fig. 3.13 Inferencia Max-producto

Al igual que en el ejemplo usado en la sección anterior, el vector  $A'$  contenía un simple valor exacto. Debido a esta limitada pero típica situación encontrada en la práctica, podemos nuevamente hacer una observación que facilite el cálculo de  $B'$ . Dado que la medición de A es  $x$ , podemos obtener  $B'$  a partir de

$$B' = \mu_A(x) \cdot \mu_B(x)$$

Para nuestro ejemplo

$$\begin{aligned} B' &= 0.5 \cdot (0, .06, 1, .6, 0) \\ &= (0, .3, .5, .3, 0) \end{aligned}$$

Como se mencionó anteriormente, la inferencia max-min produce una versión recortada de B. En cambio la técnica de inferencia max-producto produce una versión escalada de B. En este sentido, la inferencia max-producto conserva más información que la max-min.

**3.4.3 Evaluando múltiples reglas difusas.** Considerando el caso donde se tienen  $n$  reglas difusas o asociaciones  $(A_1, B_1), \dots, (A_n, B_n)$ , nuestro interés es el grado total de confianza en B dada una medición simple de  $A'$ . Procedemos aplicando  $A'$  en paralelo al banco de reglas, produciendo un conjunto difuso  $B'_i$  para cada regla. Se suma entonces todos los conjuntos  $B'_i$  para formar el conjunto compuesto resultante  $B'$ , usando la siguiente operación estándar para la unión, donde B está definida en el dominio de X.

$$\begin{aligned} B' &= B'_1 \cup B'_2 \cup \dots \cup B'_{n-1} \cup B'_n \\ &= \max(B'_1(x), B'_2(x), \dots, B'_{n-1}(x), B'_n(x)) \quad \text{para todas las } x \in X \end{aligned}$$

Después de realizar esta operación se puede desdifusificar el conjunto  $B'$  resultante mediante el método del centroide o cualquier otro. Este proceso produce un valor de salida exacto  $y_j$ .

### 3.5 Desdifusificación.

La salida del controlador difuso es en sí una cantidad difusa, la cual no puede ser interpretada por el proceso bajo control, ya que éste maneja conceptos y cantidades exactas. Por ello es necesario acondicionar las salidas del controlador de tal manera que la señal de salida sea 'desdifusificada' o mapeada en un dominio exacto. Existen varias técnicas que nos permiten realizar esta tarea, de entre las cuales destacan las siguientes:

- ◆ Desdifusificación por Centro de Gravedad (COG) ó por Centro de Area (COA).
- ◆ Desdifusificación por Centro de Sumas.
- ◆ Desdifusificación por Centro del Area más Extensa.
- ◆ Desdifusificación por el Inicio del Máximo.
- ◆ Desdifusificación por Altura.

**3.5.1 Desdifusificación por Centro de Gravedad.** Este método al que también se le conoce como el método del *centroide* o del *centro de masas* es el más popular hasta la fecha, y su expresión es la siguiente:

$$u^* = \frac{\sum_{i=1}^I u_i \cdot \mu_{u_i}}{\sum_{i=1}^I \mu_{u_i}}$$

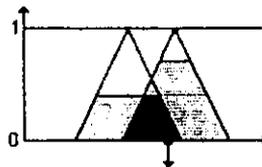
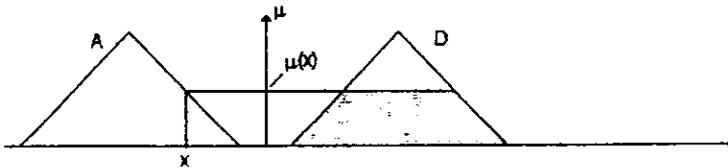


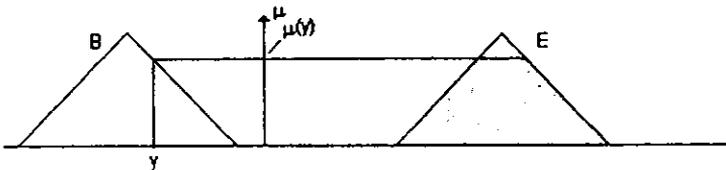
Fig. 3.14 Desdifusificación por centro de gravedad

Como se puede observar en la figura 3.14, el valor exacto producto de la desfusificación, está determinado por la parte central de la totalidad del área de los conjuntos recortados. Con el fin de aclarar mejor este método, y debido también a que es uno de los más utilizados, a continuación se muestra un ejemplo un poco más detallado, en el que se presupone el uso del método de inferencia max-min. El ejemplo se ve en cuatro etapas, mostrando en las primeras tres los conjuntos resultantes del disparo de las respectivas reglas (proceso de inferencia). La última etapa muestra el área formada con la combinación de los tres conjuntos recortados, de la cual se extrae el valor exacto que representa a la cantidad desfusificada.

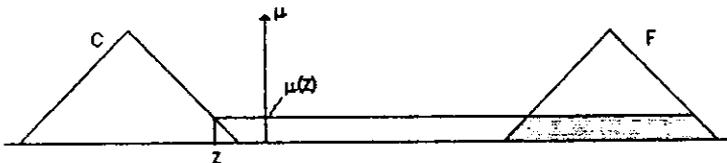
Regla 1: SI  $x$  es A ENTONCES  $n$  es D:



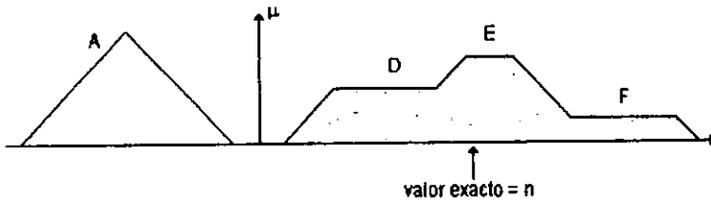
Regla 2: SI  $y$  es B ENTONCES  $n$  es E:



Regla 3: SI  $z$  es C ENTONCES  $n$  es F:



Desdifusificación:



**3.5.2 Desdifusificación por Centro de Sumas.** Este método es muy similar al anterior pero tiene la ventaja de que es un poco más rápido. La motivación para usar este método, es para simplificar los cálculos. La idea es considerar la contribución de áreas de cada función recortada en forma individual. Matemáticamente, el método del centro de gravedad obtiene el valor final a partir de la unión de las áreas recortadas, mientras que el método de Centro de Sumas considera no solo la unión, sino el total de la suma de las funciones.

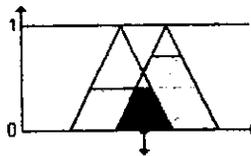


Fig. 3.15 Desdifusificación por centro de sumas

**3.5.3 Defusificación por el Centro de la Área Mayor.** El centro de la área mayor es usado en el caso de que alguna función de membresía de salida sea no convexa, siendo necesario que existan por lo menos 2 que sean convexas. Entonces el método determina el subconjunto difuso convexo con el área más extensa y define el valor de salida exacto  $u^*$ , el cual es el centro de área de ese subconjunto en particular.

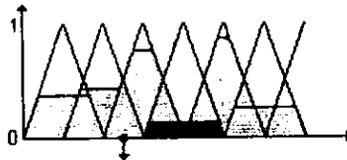


Fig. 3.16 Desdifusificación por centro de área mayor. El área oscura representa el traslapamiento de dos subconjuntos convexas.

**3.5.4 Desdifusificación por el Inicio del Máximo.** El valor exacto, producto de este método está determinado por el valor más pequeño del dominio, en el cual se encuentra definido el subconjunto con el mayor grado de membresía, tal como se muestra en la figura. Así,  $u^*$  se da mediante la siguiente expresión:

$$u^* = \inf \{ u \in U \mid \mu_U(u) = \text{hgt}(U) \}$$

donde  $U$  = universo de discurso de la variable de salida  
 $\text{hgt}(U)$  = grado de membresía del subconjunto de más altura  
 $\inf$  = el elemento inferior (el de más a la izquierda) del dominio en el intervalo que que ocupa el máximo.

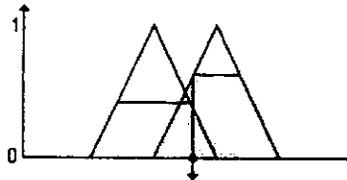


Fig. 3.17 Desdifusificación por el inicio del máximo

Además de la desdifusificación mediante la parte inicial del máximo, también existen tanto la desdifusificaciones que consideran la parte intermedia como la final del máximo.

**3.5.5 Desdifusificación por Altura.** Este método toma en consideración cada uno de los conjuntos recortados de salida. Los picos de cada uno de estos conjuntos son sumados, construyéndose así un peso global de esos picos. De esta forma, ni el soporte, ni la forma del conjunto se involucran en el cálculo de  $u^*$ . Las ventajas de la desdifusificación por altura, es que resulta un método muy simple y rápido. Siendo entonces  $c^{(k)}$  el valor del pico de un conjunto difuso de salida, y  $f_k$  la altura del mismo conjunto recortado, entonces el método de desdifusificación por altura es un sistema de  $m$  reglas, que formalmente se da por:

$$u^* = \frac{\sum_{k=1}^m c^{(k)} \cdot f_k}{\sum_{k=1}^m f_k}$$

**3.5.6 Comparación y evaluación de los métodos de desfusificación.** En esta parte se sugieren algunos criterios, los cuales se considera debería satisfacer el método ideal de desfusificación. En base a estos criterios, se realiza una evaluación para elegir el método adecuado para nuestra aplicación en particular. Algunos de estos criterios son los siguientes:

1.- *Continuidad:* Un pequeño cambio en la entrada del controlador no debería de resultar en un gran cambio en la salida.

2.- *Definición:* En la figura 3.18 se presentan dos áreas igualmente extensas, cubiertas por dos subconjuntos difusos convexos, los cuales forman el total de la salida de control. Así, el método de desfusificación no puede elegir entre esas dos áreas. En otras palabras, la situación es ambigua. Este criterio por lo tanto es importante a considerar para el caso del método de desfusificación por el área más extensa.

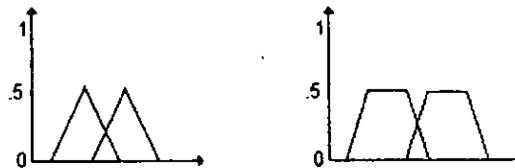


Fig. 3.18

3.- *Plausibilidad:* Cada salida de control desfusificada tiene una componente horizontal  $u^*$ , y una componente vertical  $\mu_u(u^*)$ . Decimos que  $u^*$  es plausible si ésta cae aproximadamente en la mitad del soporte de la variable de salida y tiene un alto grado de membresía en la función. El método del centroide, aplicado a la siguiente figura no satisface esta propiedad: aunque el centro del área cae en la mitad del soporte, su grado de membresía es uno de los menores.

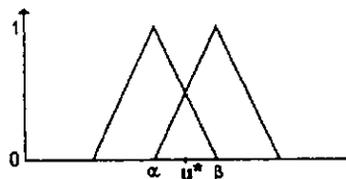


Fig. 3.19

4.- *Complejidad de cómputo:* Este criterio es particularmente importante en aplicaciones prácticas de controladores. El método de desfusificación por altura, junto con el inicio del máximo, son métodos rápidos, mientras que el de centro de sumas depende de la forma de la función de membresía de salida. En el caso de la desfusificación del centroide y del centro de sumas, una representación tabular de los conjuntos difusos, en combinación con los conjuntos difusos recortados, hacen más lento el procedimiento de desfusificación.

### 3.6 Controladores Difusos Equivalentes.

El equivalente difuso de los controladores P (PID,PD,PI,P) es normalmente construido mediante reglas que relacionan el error ( $e$ ) y el cambio de error ( $\Delta e$ ) con la salida del controlador. Por ejemplo:

SI  $e$  es PM Y  $\Delta e$  es NS ENTONCES  $u$  es PB

Si se requiere diseñar un controlador difuso equivalente a cualquiera de los controladores proporcionales P, PD, PI o PID, esto implica ya la elección tanto de las variables de proceso como de las variables de control de salida, al igual que los contenidos de los antecedentes y las consecuencias de las reglas. Las variables de proceso que representan los contenidos de los antecedentes de la regla (parte SI) son elegidas de entre :

- error, denotado por ' $e$ '
- cambio de error, denotado por ' $\Delta e$ '
- suma de errores, denotada por ' $\delta e$ '

La variable de control para la(s) salida(s) representa los contenidos de la consecuencia de la regla (parte ENTONCES), y es elegida de entre:

- cambio del control de salida, denotado por ' $\Delta u$ '
- control de salida, denotado por ' $u$ '

Aún más, por analogía con un controlador convencional, tenemos que:

- $e(k) = y_{sp} - y(k)$
- $\Delta e = e(k) - e(k-1)$
- $\Delta u(k) = u(k) - u(k-1)$

En estas expresiones,  $y_{sp}$  es el punto de ajuste o valor deseado para el control de salida, " $y$ " es la variable de proceso para la salida y  $k$  es el tiempo de muestreo.

**3.6.1 Equivalente difuso P.** La representación simbólica de una regla para un controlador difuso P es la siguiente:

SI  $e$  es <antecedente> ENTONCES  $u$  es <consecuencia>

**3.6.2 Equivalente difuso PD.** La ecuación dada para un controlador PD convencional es:

$$u = k_p \cdot e + K_D \cdot \Delta e$$

donde  $k_p$  y  $K_D$  son las constantes de ganancia proporcional y derivativa respectivamente. Entonces un equivalente difuso del controlador PD consiste de reglas cuya descripción simbólica es:

SI  $e(k)$  es <antecedente 1> Y  $\Delta e(k)$  es <antecedente 2>  
 ENTONCES  $u(k)$  es <consecuencia>

donde los antecedentes son los nombres simbólicos de valores de variables lingüísticas.

**3.6.3 Equivalente difuso PI.** La ecuación dada para un controlador PI convencional es:

$$u = k_p \cdot e + K_I \cdot \int e dt$$

donde  $k_p$  y  $K_I$  son las constantes de ganancia proporcional e integral respectivamente. Cuando se deriva con respecto al tiempo la expresión anterior, ésta se transforma en la siguiente expresión equivalente:

$$u' = k_p \cdot e' + K_I \cdot e$$

Dada esta nueva expresión, el equivalente difuso del controlador PI se forma con reglas del tipo:

SI  $e(k)$  es <antecedente 1> Y  $\Delta e(k)$  es <antecedente 2>  
 ENTONCES  $\Delta u(k)$  es <consecuencia>

o en forma simplificada,

SI  $e$  es <antecedente 1> Y  $\Delta e$  es <antecedente 2>  
 ENTONCES  $\Delta u$  es <consecuencia>

En este caso para obtener el valor de la variable de control de salida  $u(k)$ , el cambio del control de salida  $\Delta u(k)$  es sumado a  $u(k-1)$ .

**3.6.4 Equivalente difuso PID.** La ecuación que describe a un controlador PID convencional es:

$$u = k_p \cdot e + K_D \cdot \Delta e + K_I \cdot \int e dt$$

Así, en el caso discreto de un controlador difuso PID, se tiene una variable de proceso adicional, conocida como:

- suma de errores, denotada por  $\delta e$  y calculada como:

$$\delta e(k) = \sum_{i=1}^{k-1} e(i)$$

Entonces la expresión simbólica para una regla de un controlador difuso PID es:

SI  $e(k)$  es <antecedente 1> Y  $\Delta e(k)$  es <antecedente 2> Y  $\delta e(k)$  es <antecedente 3>  
 ENTONCES  $u(k)$  es <consecuencia>

o en forma simplificada,

SI  $e$  es <antecedente 1> Y  $\Delta e$  es <antecedente 2> Y  $\delta e(k)$  es <antecedente 3>  
 ENTONCES  $u$  es <consecuencia>

En la siguiente tabla se resumen las ecuaciones anteriores junto con sus controladores lineales correspondientes. El utilizar derivadas del error en lugar de valores previos, es una acción que se asemeja más cercanamente a la forma humana de utilizar las señales de error, para clasificar las desviaciones entre la señal de proceso y la de referencia.

Tipo de regla difusa	P	I	D
SI $e$ es .... ENTONCES $u$ es ....	x		
SI $e$ es .... ENTONCES $\Delta u$ es ....		x	
SI $e(k)$ es .... Y $\Delta e(k)$ es .... ENTONCES $u(k)$ es ....	x		x
SI $e(k)$ es .... Y $e(k-1)$ es .... ENTONCES $u(k)$ es ....	x		x
SI $e(k)$ es .... Y $\Delta e(k)$ es .... ENTONCES $\Delta u(k)$ es ....	x	x	
SI $e(k)$ es .... Y $e(k-1)$ es .... ENTONCES $\Delta u(k)$ es ....	x	x	
SI $e(k)$ es .... Y $\Delta e(k)$ es .... Y $\Delta^2 e(k)$ es .... ENTONCES $\Delta u(k)$ es ....	x	x	x
SI $e(k)$ es .... Y $e(k-1)$ es .... Y $e(k-2)$ es .... ENTONCES $\Delta u(k)$ es ....	x	x	x

Por otra parte, un controlador difuso es superior a un controlador PID convencional y por eso puede ser usado para obtener un mejor control (por ejemplo en términos de velocidad) de un proceso lineal. Debido a que los controladores difusos tienen un mayor número de grados de libertad, éste puede ser usado para obtener por lo menos el mismo desempeño que podría ser logrado con el controlador PID convencional. Además los controladores difusos PID pueden ser usados para implementar controladores PID no lineales.

### 3.7 Diseñando un controlador difuso.

Como ejemplo general, consideremos el diseño de un controlador difuso para una turbina de vapor. El diagrama a bloques de este sistema de control aparece como sigue:

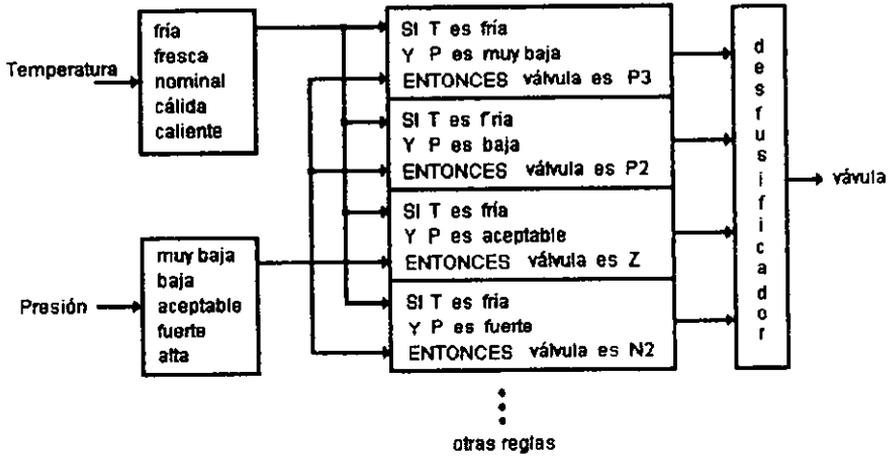
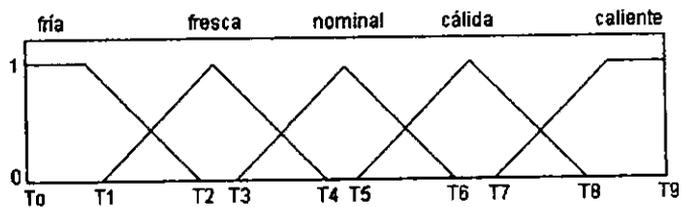
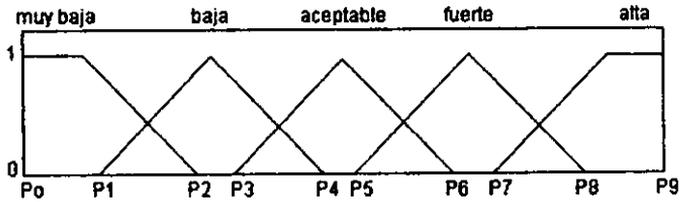


Fig. 3.20

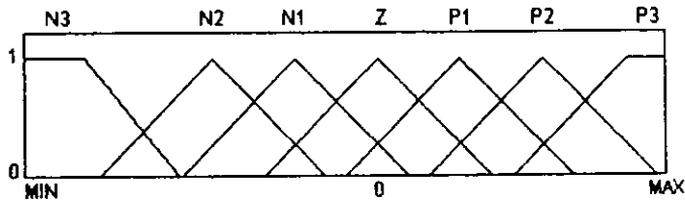
La variable de entrada " Temperatura " se mapea en los siguientes conjuntos difusos:



... mientras que la variable de entrada "Presión" se representa como:



La variable de salida (ajuste de válvula) se mapea como:



donde:

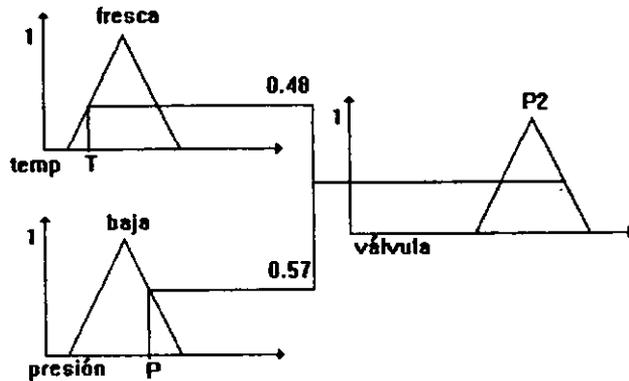
- N3: muy negativo
- N2: medio negativo
- N1: poco negativo
- Z: zero
- P1: poco positivo
- P2: medio positivo
- P3: muy positivo

El conjunto de reglas incluye se

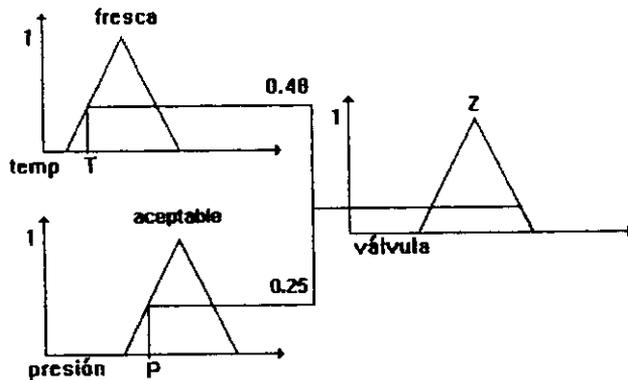
- regla 1: SI temperatura es fresca Y presión es muy baja  
ENTONCES válvula es P3.
- regla 2: SI temperatura es fresca Y presión es baja  
ENTONCES válvula es P2.
- regla 3: SI temperatura es fresca Y presión es aceptable  
ENTONCES válvula es Z.
- regla 4: SI temperatura es fresca Y presión es fuerte  
ENTONCES válvula es N2.

Considerando que la temperatura está en el estado "fresca", y la presión en los estados "baja" y "aceptable", los valores de presión aseguran que sólo las reglas 2 y 3 serán disparadas:

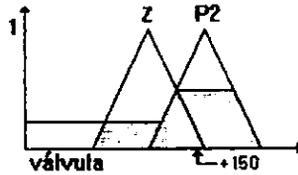
Evaluación de la regla 2:



Evaluación de la regla 3:



Las dos salidas son posteriormente desdifusificadas mediante el método del centroide:



El valor de la salida ajustará la posición de la válvula, y posteriormente el ciclo de control comenzará nuevamente a generar el próximo valor.

**Construcción del controlador.**

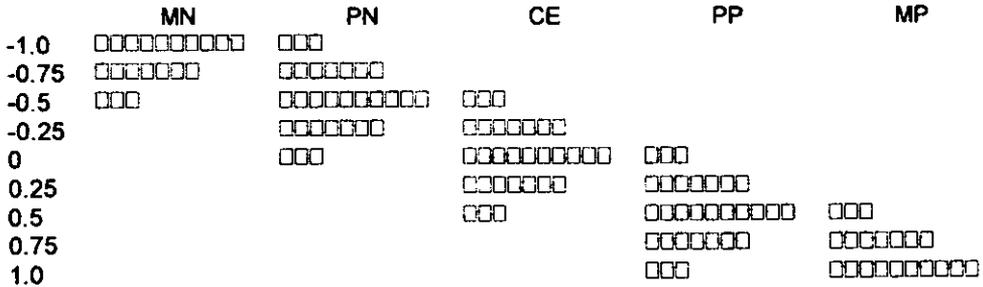
Consideremos la implementación del controlador realimentado en base a un circuito microcontrolador, y definamos un conjunto difuso para la variable error "e", el cambio de error "Δe", y la salida de la siguiente manera:

- MP:** muy positivo
- PP:** poco positivo
- CE:** cero
- PN:** poco negativo
- MN:** muy negativo

Si los rangos de error van -1 a +1, con una resolución de 0.25 en el convertidor A-D usado, entonces los conjuntos difusos de las variables de entrada, pueden ser descritos de una manera muy simple con una tabla, en la cual, el error, el cambio de error y la salida muestran sus posibles valores en el renglón superior, y los valores de verdad para cada función de membresía forman un arreglo en los renglones inferiores.

	-1	-0.75	-0.5	-0.25	0	0.25	0.5	0.75	1
$\mu(MP)$	0	0	0	0	0	0	0.3	0.7	1
$\mu(PP)$	0	0	0	0	0.3	0.7	1	0.7	0.3
$\mu(CE)$	0	0	0.3	0.7	1	0.7	0.3	0	0
$\mu(PN)$	0.3	0.7	1	0.7	0.3	0	0	0	0
$\mu(MN)$	1	0.7	0.3	0	0	0	0	0	0

o en forma gráfica, donde cada "□" tiene un valor de 0.1:



Supongamos que este sistema difuso tiene la siguiente base de reglas:

- regla 1: SI  $e = CE$  Y  $\Delta e = CE$  ENTONCES salida = CE
- regla 2: SI  $e = CE$  Y  $\Delta e = PP$  ENTONCES salida = PN
- regla 3: SI  $e = PN$  Y  $\Delta e = PN$  ENTONCES salida = MP
- regla 4: SI  $e = MP$  Y  $\Delta e = MP$  ENTONCES salida = MN

Estas salidas pueden ser desdifusificadas usando el método del centroide:

$$\sum (l = 1 \text{ to } 4 \text{ OF } (\mu(l) * \text{salida}(l))) / \sum (l = 1 \text{ to } 4 \text{ OF } \mu(l))$$

ahora si suponemos que en un tiempo dado tenemos:

$$e = 0.25$$

$$\Delta e = 0.5$$

Entonces esto da lo siguiente:

	e	$\Delta e$
$\mu(MP)$	0	0.3
$\mu(PP)$	0.7	1
$\mu(CE)$	0.7	0.3
$\mu(PN)$	0	0
$\mu(MN)$	0	0

Tomando esto en consideración para el caso de la regla 1, nos da:

**regla 1:** SI  $e = CE$  Y  $\Delta e = CE$  ENTONCES  $salida = CE$

$$\mu(1) = \min(0.7, 0.3) = 0.3$$

$$salida(1) = 0.3$$

donde:

- $\mu_A(1)$  :

valor de verdad de la función de membresía resultante para la regla 1.

- $salida(1)$  :

valor (para la regla 1) en donde la función de membresía resultante (CE) es máxima sobre el rango de la variable difusa de salida.

Las otras reglas producen lo siguiente :

**regla 2:** SI  $e = CE$  Y  $\Delta e = PP$  ENTONCES  $salida = PN$

$$\mu(2) = \min(0.7, 1.0) = 0.7$$

$$salida(2) = -0.5$$

**regla 3:** SI  $e = PN$  Y  $\Delta e = PN$  ENTONCES  $salida = MP$

$$\mu(3) = \min(0.0, 0.0) = 0.0$$

$$salida(3) = 0.75$$

**regla 4:** SI  $e = MP$  O  $\Delta e = MP$  ENTONCES  $salida = MN$

$$\mu(4) = \max(0.0, 0.3) = 0.3$$

$$salida(4) = -1$$

Los cálculos de la desfusificación produce lo siguiente:

$$\frac{\mu(1) * salida(1) + \mu(2) * salida(2) + \mu(3) * salida(3) + \mu(4) * salida(4)}{\mu(1) + \mu(2) + \mu(3) + \mu(4)}$$
$$= \frac{(0.3 * 0) + (0.7 * -0.5) + (0 * 0.75) + (0.3 * -1)}{0.3 + 0.7 + 0.3}$$
$$= \frac{0 - 0.35 + 0 - 0.3}{1.3}$$
$$= -0.65/1.3$$
$$= -0.5$$

### 3.8 Aplicaciones.

La tecnología producto de la lógica difusa ha experimentado una remarcada popularidad desde hace algunos años, gracias a la abundancia de aplicaciones comerciales. Sin embargo, con la excepción de algunos proyectos, la mayoría de los éxitos comerciales, involucran problemas relativamente simples para aplicaciones en el hogar y en el mercado electrónico de consumo. A continuación se muestran algunas aplicaciones que van más allá de lo puramente comercial, proporcionando solución a problemas en varias disciplinas.

Desde 1974, cuando la primera aplicación de controladores difusos fue reportada, han habido muchos avances. Ahora tenemos un número de conceptos teórico-prácticos muy valiosos. Distinguimos dos períodos principales: el primero se concentra en los experimentos en laboratorio y desarrollo de prototipos con instalaciones industriales relativamente complejas. El segundo, comienza a finales de la década de los 80's, junto con un vasto número de aplicaciones comerciales, visible en forma particular en lo conocido como electrónica de consumo. Tales aplicaciones comprenden principalmente aparatos de uso doméstico, que con la incorporación de esta tecnología en sus diseños, se vuelven más prácticos para el usuario. Posteriormente, fueron apareciendo más aplicaciones de uso industrial.

Los resultados obtenidos de los controladores difusos fueron promisorios; estudios con simulación claramente indicaron un rendimiento óptimo de el modelo. Pappis y Mamdani discutieron el uso de un controlador difuso para control de tráfico en una intersección de dos calles de sentido único. Cuando se establece el promedio de retardo de los vehículos como índice de eficiencia, el controlador difuso trabaja mejor que con un controlador convencional. Por otra parte, también se realizaron experimentos por Lemke y Kickert en un proceso de agua cálida. Su principal objetivo fue controlar la temperatura del agua que sale de un tanque, así como el flujo de la misma. El proceso estaba sujeto a ruido, asimetrías alrededor del punto de ajuste, y falta de linealidad. Una comparación del controlador difuso con un controlador digital PI óptimamente ajustado, mostró que el controlador difuso tuvo una respuesta más rápida con la misma precisión. Versiones simplificadas del controlador difuso, arrojaron respuestas escalón más rápidas, pero también oscilaciones más largas alrededor del punto de ajuste.

Con la finalidad de mostrar en una forma más clara el rol que los conjuntos difusos han jugado en el desarrollo de algunas aplicaciones, a continuación se describen de una manera muy condensada 4 de ellas, para posteriormente hacer mención de otras en las que la explicación es un poco más abundante.

1.- Matsushita Electrical introdujo una máquina lavadora automática controlada difusamente. Esta máquina es quien decide si una carga es pequeña o grande, en base a lo cual establece las acciones de control relevantes. Por supuesto, existe un poco de subjetividad en este tipo de control, por ejemplo, en la decisión de que si una carga pequeña, con una gran cantidad de grasa en la ropa necesitará más lavado que una carga no muy grande de ropa ligeramente sucia. Los controladores difusos toman atención de ello, y procesan este tipo de información mediante el uso de valores lingüísticos. La lavadora tiene sensores adicionales que pueden detectar y medir el tamaño de la carga, la calidad y cantidad de la suciedad (ésto es hecho midiendo la turbiedad del agua de lavado), y el tipo de detergente que será usado. El resultado de las mediciones (las cuales no son del todo precisas) son posteriormente alimentadas al controlador difuso en forma de entradas de información, con lo cual se produce una secuencia de decisiones de control.

2.- La misma compañía liberó otro producto que incorpora un controlador difuso en una lavadora de vacío. Es evidente que el poder de succión de esta aplicación depende de la cantidad de polvo en la superficie. La lavadora de vacío, percibe tanto las condiciones del piso como la calidad de polvo por medio de sus sensores. El controlador difuso controla el poder de absorción para un nivel óptimo.

3.- Mitsubishi en su división de industria pesada, introdujo un sistema de aire acondicionado usando 25 reglas de control para el calentamiento y otras 25 para el proceso de enfriamiento, registrándose una asombrosa reducción de cerca del 24% en el consumo de energía en comparación con algunos modelos previos. El desarrollo del sistema completo fue rápido; el conjunto inicial de reglas fue escrito en 3 días; todas las funciones de membresía utilizadas en las reglas fueron juntadas e incorporadas en un mes; el refinamiento final no tomó más de tres meses.

4.- La video cámara de 8mm de Sanyo, usa conjuntos difusos para evaluar el enfoque y las condiciones de luminosidad. Las reglas son usadas para controlar la apertura del diafragma. Otra aplicación útil de un controlador difuso es una videogradora equipada con un estabilizador de imagen, desarrollado por Matsushita. Este estabilizador ayuda a corregir los movimientos involuntarios de la cámara durante el proceso de grabación.

**3.8.1 Mejorando la respuesta de sensores de temperatura con técnicas de control difuso.** En temperaturas de procesos industriales se requiere una rápida respuesta en los dispositivos de control y medición. La velocidad del sistema de medición está esencialmente determinada por el tiempo de respuesta del sensor primario. Por lo tanto, es necesario mejorar la conducta dinámica de tal sensor. En una primera aproximación, la reacción transitoria del sensor se asume exponencial. La constante de tiempo relativamente larga está condicionada por la construcción mecánica del dispositivo, determinando la temperatura final en tiempos largos. Existen 2 soluciones a este problema, una implica el cambio de la construcción del sensor, lo cual es muy difícil, y la otra es conectar en forma serial un elemento que presente una respuesta inversa a la del sensor. Ahora se presenta un método alterno que se basa en los principios de la lógica difusa.

La idea fundamental es forzar una respuesta rápida del sensor. Esto puede ser realizado mediante una aproximación rápida de la "temperatura eigen" del sensor a la temperatura a medir, haciendo circular una corriente determinada a través del sensor.

Existen dos problemas que deben tomarse en consideración:

- 1.- ¿Cómo sabe el sensor, si la temperatura eigen es mayor, menor o igual a la temperatura a medir?
- 2.- Qué corriente es necesario aplicar al sensor, sobre todo porque la temperatura a medir se desconoce, al igual que la diferencia entre la temperatura eigen y la temperatura a medir.

Para resolver el problema se desarrolló un nuevo método basado en comparaciones inteligentes de temperatura. Primeramente, el sensor es alimentado con una corriente nominal durante un intervalo de tiempo exactamente definido, observándose los cambios en el sensor. Este cambio es proporcional a la diferencia entre la temperatura intrínseca de la corriente y la temperatura a medir. La corriente necesaria para un calentamiento acelerado es calculada y aplicada al sensor. Después de esto, el sensor es alimentado nuevamente durante un nuevo intervalo de tiempo exacto con la corriente nominal, observándose los cambios en el sensor. Después de este cambio de temperatura, el sensor es alimentado nuevamente pero ahora con una corriente controlada

difusamente para un posterior autocalentamiento. El control de la corriente que circula por el sensor, mediante dispositivos analógicos es una tarea relativamente complicada. Son varios los factores que influyen en el valor de esta corriente, complicándose así una posible solución matemática.

La corriente que calienta al sensor es controlada como una variable de salida de un controlador difuso, teniendo éste dos variables de entrada, que son, la diferencia de temperatura  $\Delta T$  y la temperatura medida  $T$ . La siguiente figura muestra el sistema con sus dos entradas y su salida.

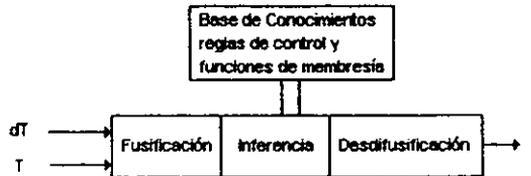


Fig. 3.21 Sistema de control

Como parte del desarrollo, fue efectuada una simulación con software, sin tomar en consideración la compensación de calor dentro del sensor. El resultado de esta simulación es mostrado en la siguiente gráfica.

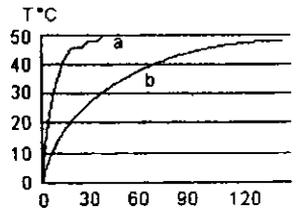


Fig. 3.22 Respuesta mejorada

- donde a= Característica con corriente controlada difusamente.  
b= Característica natural del sensor.

La figura muestra que la temperatura final es alcanzada en un tiempo mucho más corto con el control de temperatura. Así también se realizó una prueba experimental en base a un sensor de respuesta lenta, observándose los resultados mostrados en la gráfica 3.23. Es evidente que la temperatura final es alcanzada después del primer procedimiento de calentamiento. Después de quitar la corriente de aceleración, la temperatura desciende a causa del efecto de compensación de calor dentro del sensor. Tiempo después, el sensor es calentado nuevamente vía el controlador difuso. El proceso es repetido hasta que la temperatura final es alcanzada. El tiempo de medición con el sensor controlado, es considerablemente reducido, en comparación con el sensor sin controlar.

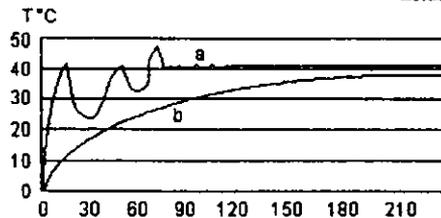


Fig. 3.23 Respuesta mejorada de un sensor lento

donde a= Característica con corriente controlada difusamente.  
b= Característica natural del sensor.

**3.8.2 Control difuso de un horno de cemento.** En este caso se considera un proceso físico como objeto de control, el cual se describe a continuación.

El cemento es fabricado calentando una mezcla de arcilla, cal, arena, y hierro en bruto, a una temperatura la cual permite la formación del complejo compuesto que es el cemento, silicato dicálcico ( $C_2Si$ ), silicato tricálcico ( $C_3Si$ ), aluminato tricálcico, y aluminoferrita tricálcica ( $CrAlF$ ). En la primera etapa del proceso de homeado, la mezcla es secada extrayendo de ella el exceso de agua. En la segunda etapa toma lugar una calcinación con la descomposición del carbonato de calcio en óxido de calcio y en dióxido de carbono. En la última etapa, la temperatura dentro del horno alcanza de los 1250 a los 1450 °C y la cal libre se combina con los otros ingredientes para formar el cemento. El producto final de este proceso es conocido como "clinker". El horno es un caparazón de hierro de 130m de largo y de 5m de diámetro, el cual se encuentra montado con un pequeño ángulo de inclinación con respecto a la horizontal, y su interior es recubierto con una pared de material refractario (tabique refractario). Este enorme cilindro se encuentra girando a una velocidad muy baja, que generalmente es de 1 rev/min. La materia prima se introduce por la parte de arriba del horno, y gracias a la inclinación y a la rotación de este, el material es transportado hasta la salida del horno en un tiempo de aproximadamente 3 horas y 15 minutos, permaneciendo el producto otros 45 minutos en el enfriador de clinker. El calor en el horno es proporcionado por carbón pulverizado y mezclado con aire, lo que se conoce como aire primario. Los gases calientes de la combustión son aspirados a través del horno por un ventilador de inducción, colocado en la parte posterior del cilindro.

El principal problema en la modelación matemática para una estrategia de control, es que la relación entre las variables de entrada (características medidas del proceso) y las variables de control es compleja, no lineal, y contiene retardos de tiempo e interrelaciones, aparte, la respuesta del horno a las entradas de control, depende de las condiciones prevalecientes del horno. Esta fue ciertamente la razón por la que se pensó en un sistema de control difuso.

De las muchas posibles variables de entrada y de control, las siguientes fueron escogidas como relevantes.

Variables de entrada:

- 1.- Temperatura de fatiga del gas - temperatura posterior (BT).
- 2.- Temperatura intermedia del gas - temperatura de anillo (RT).
- 3.- Temperatura de zona de quemado (BZ).
- 4.- Porcentajes de oxígeno en los gases de fatiga (O2), y
- 5.- Peso de litro (LW) - indica la calidad del clinker.

El proceso es controlado variando las siguientes variables de control:

- 1.- Proceso del horno (KS).
- 2.- Alimentación de carbón (CS) - combustible, y
- 3.- Succión inducida - velocidad del ventilador (BF).

Los cálculos para la acción de control se basaron en las siguientes cuatro etapas:

- 1.- Cálculo del error presente y de su razón de cambio.
- 2.- Conversión de los valores de error en variables difusas.
- 3.- Evaluación de las reglas de decisión usando la regla composicional de inferencia.
- 4.- Evaluación de la entrada determinística requerida para regular el proceso.

Con respecto a las estrategias de control, citemos lo siguiente.

El objetivo del sistema de control computarizado del horno, es automatizar la rutina de control del hornero experimentado. Las estrategias aplicadas están basadas en estudios detallados de las experiencias del operador del proceso, las cuales incluyen un modelo cualitativo de la influencia de las variables de control sobre las variables medidas.

1.- Si la cantidad de carbón alimentado se incrementa, la carga del controlador del horno así como la temperatura en la cámara de humo, se incrementarán, mientras el porcentaje de oxígeno y el contenido de cal libre disminuyen.

2.- Si el flujo de aire se incrementa, la temperatura en la cámara de humo y el contenido de cal libre se incrementarán, mientras la carga del controlador de horno y el porcentaje de oxígeno se decrementan.

En las bases de una discusión continua con los operadores, Jensen [1976] definió 75 condiciones de operación como sentencias condicionales difusas del tipo:

IF el gradiente de la carga del controlador es	(OL,SL,OK,SH,DH)
AND la carga del controlador es	(OL,SL,OK,SH,DH)
AND la temperatura en la cámara de humo es	(L,OK,H)
THEN el cambio de porcentaje de oxígeno es	(VN,N,SN,ZN,OK,ZP,SP,P,VP)
PLUS el cambio del flujo de aire es	(VN,N,SN,ZN,OK,ZP,SP,P,VP)

Los siguientes términos difusos primarios son usados para las variables medidas:

- 1.- DL = Drásticamente bajo
- 2.- L = Bajo
- 3.- SL = Ligeramente bajo
- 4.- OK = OK
- 5.- SH = Ligeramente alto
- 6.- H = Alto
- 7.- DH = Drásticamente alto

Los siguientes términos difusos primarios son usados para las variables de control:

- 1.- VN = Muy negativo
- 2.- N = Negativo
- 3.- SN = Poco negativo
- 4.- ZN = Cero negativo
- 5.- OK = OK
- 6.- ZP = Cero positivo
- 7.- SP = Poco positivo
- 8.- P = Positivo
- 9.- VP = Muy positivo

Las 13 condiciones de operación son definidas tomando sólo algunas de las combinaciones consideradas e incluyendo también valores previos del gradiente de carga del controlador. Con el fin de decidir que debería ser cambiado, si el punto de ajuste del porcentaje de oxígeno o el flujo de aire, se formulan tres reglas difusas adicionales para cada condición de operación, basadas en los valores actuales del porcentaje de oxígeno y la temperatura de la cámara de humo, lo cual resulta en 39 reglas de control.

**3.8.3 Control difuso de un sistema de operación de trenes.** La aplicación que a continuación se describe es un control difuso para un sistema de transporte público en Sendai, Japón. Ya que las condiciones de tráfico dentro de las áreas metropolitanas empeora al paso del tiempo, la ciudad de Sendai planea el diseño de un sistema de metro que pueda proporcionar a los ciudadanos los más altos niveles de confort, seguridad y eficiencia. Para tal propósito, el control difuso para la operación del tren parece el apropiado para lograr una aceleración suave, desaceleración y frenado. Las estrategias de operación se dividen en dos: control de velocidad constante (CSC) y control automático de detención del tren (TASC). CSC arranca al tren y mantiene su velocidad bajo un límite especificado. la operación TASC desacelera al tren con el fin de detener su avance en una posición destino.

El control difuso convencional se convierte en una desventaja en esta aplicación en particular, ya que ésta no evalúa los resultados de los comandos de control elegidos. Para subsanar esto, fue necesario hacer uso del control difuso predictivo, donde las reglas son como sigue:

$R_i : "Si (u \text{ es } C_i \implies 'x' \text{ es } A_i \text{ Y } 'y' \text{ es } B_i) \text{ ENTONCES } u \text{ es } C_i"$

En términos lingüísticos, esta regla dice: "Si la variable de estado  $x$  es  $A_i$  y la variable de estado  $y$  es  $B_i$ , cuando la acción de control es elegida para  $C_i$ , ENTONCES la acción de control  $C_i$  es usada como la salida del controlador; ' $x$ ' y ' $y$ ' son variables de estado de esos sistemas y  $u$  es la acción de control. Las variables de estado son evaluaciones del sistema de tren, de acuerdo a criterios específicos, y definidos por conjuntos difusos como sigue:

- A. Índice de seguridad (S)
- 1.- Peligro (SD)
  - 2.- Seguro (SS)

B. Índice de confort (C)

- 1.- Buen confort (CG)
- 2.- confort pobre (CP)

C. Índice de ahorro de energía (E)

- 1.- Registro de energía ahorrada
- 2.- Registro de energía no ahorrada

D. Índice de trazabilidad (T)

- 1.- Buen trazo (TG)
- 2.- Trazo preciso (TA)
- 3.- Baja velocidad (TL)

E. Índice de tiempo en activo (R)

- 1.- En zona TASC (RT)
- 2.- Fuera de zona TASC (RF)

F. Índice de frenado (G)

- 1.- Buen frenado (GG)
- 2.- Frenado preciso (GA)

La definición exacta de las respectivas funciones de membresía son omitidas aquí. Las posibles acciones de control son :

- 1.- Pn = porción de energía
- 2.- Bn = porción de frenado
- 3.- DN = Diferencia de porciones
- 4.- N = comando de control de porción (porción de energía / porción de frenado)

Basado en esas variables de estado y acciones de control, se formulan las reglas del sistema difuso predictivo. Ejemplos de tales reglas se dan a continuación:

1.- Reglas CSC

- SI (N es 0  $\Rightarrow$  S es SS, C es CG Y E es ES) ENTONCES N es 0.  
SI (DN es 0  $\Rightarrow$  S es SS Y T es TG) ENTONCES DN es 0.

2.- Reglas TASC

- SI (DN es 0  $\Rightarrow$  R es RT Y G es GG) ENTONCES DN es 0  
SI (N es 1  $\Rightarrow$  R es RT Y C es CG) ENTONCES N es 1

En el sistema ATO, se elige una acción de control evaluando todas las reglas cada 100 ms. En estudios de simulación el control predictivo difuso ATO es comparado con el convencional ATO de acuerdo al criterio "Confort óptimo", "Precisión de frenado", "Consumo de energía" y "Tiempo en activo".

## CAPITULO 4:

### *Selección y descripción del Microcontrolador*

---

La técnica de control difuso es una herramienta que por naturaleza es factible de implementarse digitalmente, ya sea en computadora o en un circuito electrónico basado en algún microprocesador. Existen algunas aplicaciones que requieren demasiados recursos de hardware, por lo cual una PC parecería ser la opción ideal para realizar el controlador. En nuestro caso afortunadamente un sistema mínimo basado en algún microprocesador conocido será más que suficiente.

El desarrollo de un circuito de control microprocesado involucra diferentes etapas, siendo una de ellas la que se refiere a la selección del tipo de arquitectura necesaria. Existen infinidad de opciones en base a las cuales puede llevarse a cabo el diseño, en este sentido podemos iniciar haciéndonos una pregunta como : ¿ qué utilizar, microprocesador o microcontrolador ? La respuesta implica considerar varios aspectos como serían los siguientes:

- capacidad de direccionamiento
- número de bits del bus de datos
- conjunto de instrucciones
- voltaje de polarización
- velocidad de trabajo
- costo

Existen microprocesadores y microcontroladores que comparten por igual algunas o la mayoría de las características citadas. Pero entonces, en qué radica la diferencia entre estos dos dispositivos. En sí, un microcontrolador no es otra cosa que un microprocesador con un determinado número de elementos incorporados en el mismo encapsulado, de tal forma que se logra integrar en muchos casos una buena parte de lo que se conoce como sistema mínimo, el cual típicamente estaba formado por el microprocesador, las memorias RAM y ROM, el circuito de reloj, el circuito de RESET, los buses de datos y dirección, los puertos, sistemas de temporización, etc. Es por esto que algunos textos se refieren a los microcontroladores también como microcomputadoras integradas.

Algunas características que se pueden encontrar en estos dispositivos pueden ser las siguientes:

- memoria RAM (por lo general menos de 2 kbytes)
- memoria ROM, EPROM ó EEPROM
- temporizadores
- contadores

- sistema de comunicación serial asincrónica y síncrona
- convertidores analógico - digital
- puertos de entrada y salida
- diferentes modos de operación
- independencia de memorias externas
- alta resistencia al ruido eléctrico
- buen desempeño inclusive ante ambientes ruidos de trabajo (industriales)
- bajos costos

Por todas estas características mencionadas no resulta difícil decidir entre microprocesador y microcontrolador, evidentemente el segundo resulta más adecuado para la totalidad de las aplicaciones en las que comúnmente se utilizaba un microprocesador. La cuestión ahora es decidir que microcontrolador emplear. Para esto se evalúan las necesidades propias del proyecto, aunque por lo general se considera el más completo, debido a que de esa forma se simplifica el diseño electrónico. Para nuestra aplicación se considera un microcontrolador de la familia 68HC11 de Motorola, el cual se ajusta adecuadamente a las necesidades del proyecto. A continuación se describen de manera breve las principales características del dispositivo así como los bloques que lo integran.

#### 4.1 Características generales del microcontrolador MC68HC11A1.

El MC68HC11A1 es un  $\mu$ controlador de 8 bits con características de integración altamente sofisticadas. La tecnología HCMOS usada en su diseño combina tamaños pequeños y altas velocidades con el bajo consumo de potencia y la alta inmunidad al ruido de la tecnología CMOS. En este circuito se integran varias características que lo convierten en uno de los más completos del mercado, tales características son:

- Modos de ahorro de consumo de energía (STOP y WAIT)
- 512 bytes de EEPROM integrada con protección de bloque
- 512 bytes de RAM integrada
- Sistema de temporización de 16 bits
- Acumulador de pulsos de 8 bits
- Circuito de interrupciones de tiempo real
- Sistema de vigilancia de operación de cómputo adecuada (COP ó perro guardián)
- Interfase serial periférica síncrona
- Interfase serial asíncrona
- Convertidor analógico - digital de 8 bits de resolución
- 38 pines de propósito general de entrada y salida
- frecuencia de trabajo de 0 a 12 Mhz

**4.1.1 Descripción de los pines.** En la figura 4.3 se presenta la distribución de pines del circuito en su presentación DIP de 48 pines. A continuación se describe la función de cada uno de los pines.

- **VDD** y **V<sub>SS</sub>**. Terminales de polarización (5 volts típica).
- **RESET**. Terminal de inicialización.

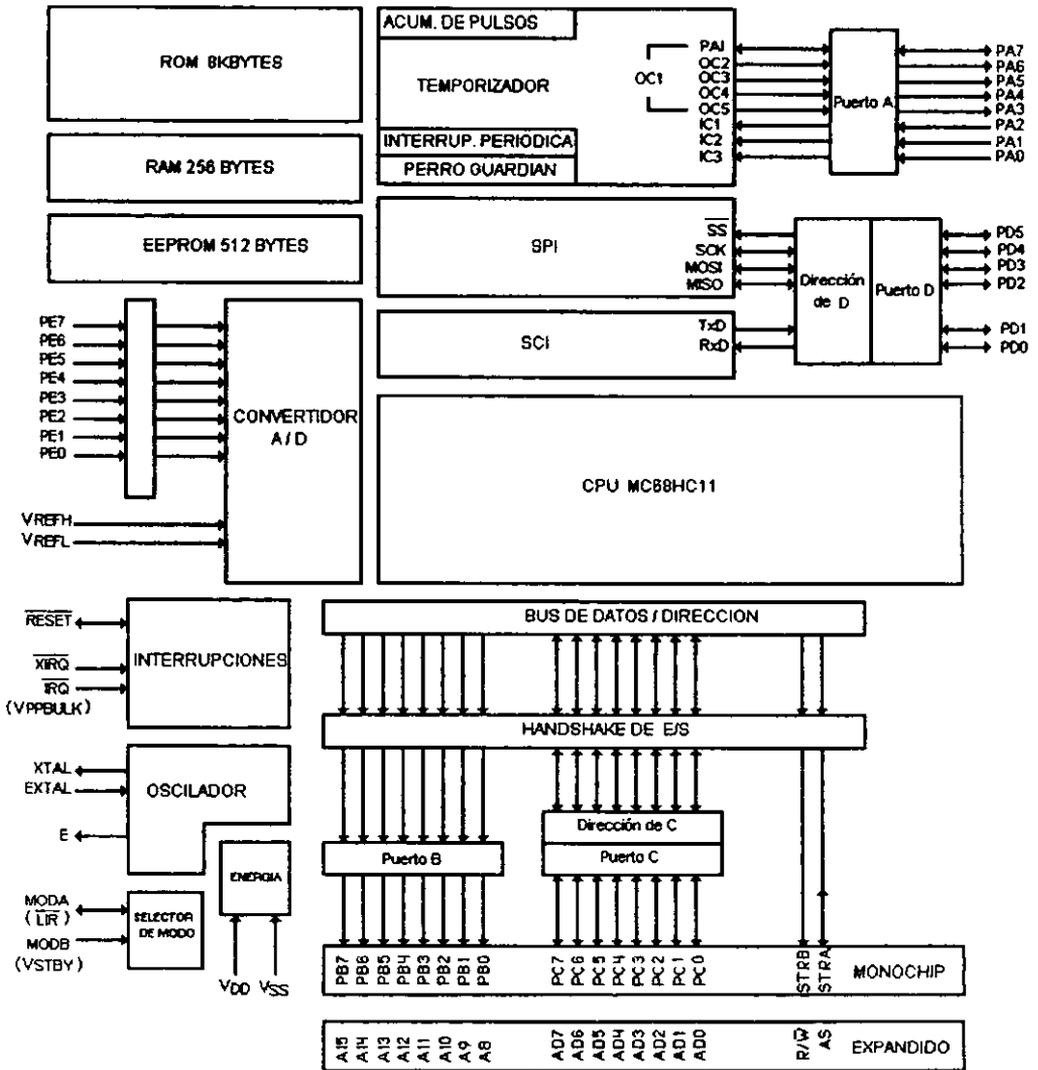


Fig. 4.1 Diagrama a bloques del MC68HC11A1

- **XTAL y EXTAL.** Terminales para señal de reloj, la cual puede ser directa o externa. En la siguiente figura se muestran las conexiones para cada caso.

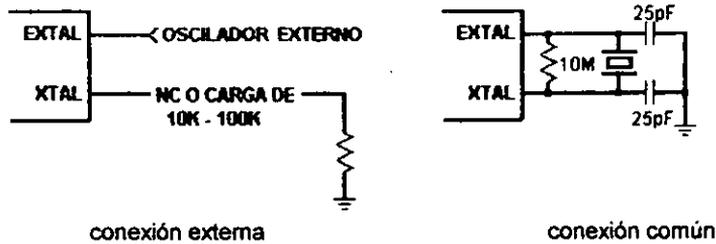


Fig. 4.2 Configuraciones de reloj

- Salida E-Clok (E). Esta terminal emite una señal proveniente del reloj E, la cual es usada como referencia de temporización. La frecuencia de esta señal es un cuarto de la frecuencia del cristal. Cuando la señal presenta un cero tiene lugar un proceso interno. Cuando presenta un alto, el microcontrolador accesa información.
- Petición de Interrupción ( $\overline{\text{IRQ}}$ ). Esta terminal de entrada proporciona una forma de solicitar una petición asíncrona de interrupción a la MCU.
- Interrupción no Enmascarable ( $\overline{\text{XIRQ}}$ ). Proporciona una forma de solicitar una petición de interrupción no enmascarable.
- **MODA/LIR** y **MODB/V<sub>STBY</sub>**. Durante un reset, MODA y MODB seleccionan uno de los cuatro modos de operación. Después de que ha sido elegido el modo de operación, la señal LIR indica cuando una instrucción ha comenzado. El pin V<sub>STBY</sub> es utilizado para proporcionar a la RAM interna un voltaje alternativo para la protección de la información.
- **VRL** y **VRH**. Estas dos entradas proporcionan el voltaje de referencia para el convertidor analógico-digital interno. Por lo general VRL tiene el mismo nivel bajo de polarización del circuito, en tanto que para una operación apropiada de conversión VRH debe ser 3 volts mayor que VRL, sin que este voltaje de referencia rebase los límites establecidos por VRL y VRH.
- **STRA/AS**. Este pin realiza cualquiera de dos funciones, dependiendo del modo de operación. En el modo monochip, STRA realiza una función de habilitación de entradas. En el modo expandido, AS proporciona una función de validación de direcciones.
- **STRB/ $\overline{\text{RW}}$** . El pin de habilitación del puerto B y de read/write, actúa ya sea como habilitador de salida del puerto B o como indicador de dirección del bus de datos, todo esto dependiendo del modo de operación.
- Señales de puertos. Los pines de los puertos tienen diferentes funciones en los diferentes modos de operación. Las funciones de los pines para los puertos A, D y E son independientes de los modos de operación. Los puertos B y C sin embargo, si son afectados por los modos de operación.

- **Puerto A.** En todos los modos de operación, este puerto puede ser configurado para manejar 3 funciones de captura de entrada (IC), y cinco funciones de comparación de salida (OC), o 4 IC's y 4 OC's.

El pin **PA7** puede funcionar como entrada/salida de propósito general o como OC1. Este pin también puede funcionar como entrada para el acumulador de pulsos.

**PA[6-4]**. Este conjunto de pines pueden servir como salidas de propósito general, OC's 2-4 u OC1.

**PA3** puede ser un pin de entrada/salida de propósito general, pin del captador IC4 o del comparador OC5, así como pin del comparador OC1.

**PA[2-0]**. Estos pines pueden ser entradas de propósito general o pines para los captadores IC1-3.

- **Puerto B.** Durante el modo monochip, todos los pines del puerto B son salidas de propósito general. En el modo expandido, todos sus pines funcionan como el byte más significativo del bus de direcciones.
- **Puerto C.** Durante el modo monochip, éste se convierte en puerto de entrada/salida de propósito general, en tanto que en modo expandido, cumple la función de bus de datos, así como también proporciona el byte menos significativo del bus de direcciones.
- **Puerto D.** Los pines **PD[5-0]** pueden ser usados para señales de entrada/salida de propósito general. Alternativamente, los pines sirven como señales de las interfaces señales de comunicación.

El pin **PD0** es la señal de recepción de datos (**RxD**) para la SCI (interfase serial de comunicaciones).

El pin **PD1** es la señal de transmisión de datos (**TxD**) para la SCI (interfase serial de comunicaciones).

Los pines **PD[2-5]** están dedicados para la SPI (interface serial de periféricos), donde **PD2** es la señal de salida **MISO** (maestro entrada - esclavo salida), **PD3** es la señal **MOSI** (maestro salida - esclavo entrada).

**PD4** es la señal **SCK** (reloj serial), y **PD5** es la entrada de selector de esclavo ( $\overline{SS}$ ).

- **Puerto E.** Este puerto puede usarse ya sea como entradas de propósito general, o como entradas analógicas para el convertidor A/D.

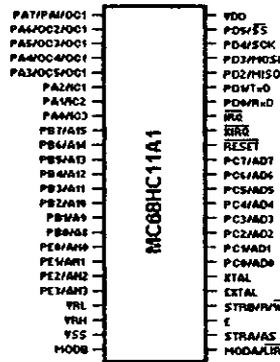


Fig. 4.3 Distribución de pines del 68HC11A1

**4.1.2 CPU y registros.** La CPU está diseñada para manejar tanto las direcciones de periféricos como las de memoria dentro del mismo mapa de 64 kbytes. No hay instrucciones especiales de entrada/salida para diferenciar entre periféricos y memoria. Esta arquitectura también permite acceder un operando desde una localidad externa de memoria sin penalización de tiempo. Los registros de la CPU son una parte integral de la misma y no son direccionados como localidades de memoria. La descripción gráfica de estos registros así como la función de cada uno son como sigue:

- **Acumuladores A, B y D.** Estos acumuladores son registros de 8 bits de propósito general, los cuales pueden almacenar operandos y resultados de operaciones aritméticas. En algunas instrucciones se unen estos dos registros para formar un registro de 16 bits, recibiendo entonces el nombre de registro D.
- **Registro Índice IX.** Este registro provee un valor indexado de 16 bits, el cual puede ser sumado a un valor fijo de 8 bits con la finalidad de formar una dirección efectiva. El IX puede ser usado también como contador o como registro de almacenamiento de propósito general.
- **Registro Índice IY.** El índice IY realiza una tarea similar al IX. Sin embargo, la mayoría de las instrucciones que usan el registro IY requieren un byte extra de código de máquina, así como un ciclo extra de tiempo en su ejecución.
- **Registro de Apuntador de Pila (SP= Stack Pointer).** La CPU tiene una pila programable, la cual puede ser colocada en cualquier espacio del mapa de direccionamiento, y también puede tener cualquier tamaño (64 Kbytes máximo). Normalmente, el registro SP es inicializado por alguna de las primeras instrucciones del programa. La pila es configurada como una estructura de datos que crece descendientemente desde una posición alta de la memoria RAM hasta una baja. Cada vez que un nuevo dato es introducido a la pila, el registro SP es decrementado automáticamente por la CPU, en tanto que cada vez que se extrae un dato, el SP es incrementado.

Cuando una subrutina es llamada por una instrucción JSR o por una BSR, la dirección de la instrucción posterior a la de brinco es almacenada en la pila, entrando primero el byte menos significativo. Cuando la rutina es finalizada, se ejecuta una instrucción de retorno de subrutina (RTS), con lo que se extrae de la pila la dirección de retorno para ser cargada en el registro PC. Una vez hecho esto, la ejecución del programa continúa en este punto.

- **Contador de Programa (PC).** El contador de programa es un registro de 16 bits, el cual contiene la dirección de la próxima instrucción a ser ejecutada. Después del reset, el PC es inicializado desde uno de seis posibles vectores, dependiendo del modo de operación y la causa del reset.

	POR o pin RESET	Monitor de reloj	Perro guardián
<b>Normal</b>	\$FFFE,F	\$FFFC,D	\$FFFA,B
<b>Test o Boot</b>	\$BFFE,F	\$BFFC,D	\$BFFA,B

- **Registro de Código de Condición (CCR) o registro de banderas.** Este registro de 8 bits contiene 5 indicadores de código de condición (C, V, Z, y H), 2 bits de máscara para interrupción (IRQ y XIRQ), y un bit de deshabilitación de STOP (S). Los códigos de condición son automáticamente actualizados por varias de las instrucciones. La descripción de cada una de estas banderas se detalla a continuación.
  - **Bandera de acarreo (C).** Este bit es puesto en 1 si la unidad aritmética-lógica (ALU) realiza un acarreo o un préstamo durante una instrucción aritmética. Este bit también actúa como indicación de error en instrucciones de división o multiplicación.
  - **Bandera de Sobreflujo (V).** El bit de sobreflujo se pone en 1 si una operación causa un sobreflujo aritmético.
  - **Bandera de Cero (Z).** Este bit toma el valor de uno cuando la ALU realiza una operación aritmética o lógica que dé como resultado un cero.
  - **Bandera de Negativo (N).** El bit N se pone en 1, si el resultado de una operación aritmética o lógica es negativo (BMS=1).
  - **Máscara (I) de Interrupción.** La máscara de petición de interrupción ( $\overline{\text{IRQ}}$ ) es una máscara global que deshabilita todas las fuentes de interrupción enmascarables. Mientras el bit I esté puesto en 1, las interrupciones pueden quedar pendientes, continuando de esta forma la ejecución normal del programa hasta que se ponga en cero el bit I y se presente una interrupción por el pin IRQ.
  - **Bandera de Medio Acarreo (H).** Este bit toma el valor de 1 cuando ocurre un acarreo entre los bits 3 y 4 al finalizar la ejecución de alguna de las instrucciones de suma.
  - **Máscara (X) de Interrupción.** El bit X deshabilita las interrupciones procedentes del pin XIRQ. Después de un reset, este bit es puesto automáticamente en 1 por la CPU, por lo cual si es necesario debe ser puesto a cero por alguna de las primeras instrucciones del programa.
  - **Deshabilitación del STOP (S).** Deshabilitar el STOP mediante el pin (S), evita que el CPU sea puesto en estado de paro cuando sea ejecutada la instrucción STOP.

**4.1.3 Tipos de Direccionamientos.** El  $\mu$ controlador HC11 cuenta con seis modos de direccionamiento: inmediato, directo, extendido, indexado, inherente y relativo. Estos modos son detallados a continuación.

1. **Direccionamiento Inmediato.** En este direccionamiento, el argumento de la instrucción se encuentra en el siguiente byte de memoria donde se ubica el código de operación.
2. **Direccionamiento Directo.** En el modo de direccionamiento directo, el byte menos significativo de la dirección del operando, es colocado en memoria un byte después del código de operación, en tanto que el byte más significativo es asumido como \$00. De esta forma, se pueden acceder las primeras 256 direcciones del mapa de memoria sin necesidad de utilizar direcciones de 16 bits, lo cual trae como consecuencia un ahorro de localidades de memoria y una disminución del tiempo de ejecución que involucra la transferencia de información.
3. **Direccionamiento Extendido.** En el modo de direccionamiento extendido, la dirección efectiva del argumento se encuentra contenida en los 2 bytes que siguen al byte del código de operación.
4. **Direccionamiento Indexado.** En este modo, se utiliza un número de 1 byte para ser sumado a la dirección a la que apuntan el registro IX o el IY, formando de esta manera una dirección efectiva. Este byte de offset ocupa el byte siguiente al que almacena el código de operación.
5. **Direccionamiento inherente.** En este modo toda la información necesaria para ejecutar la instrucción se encuentra contenida en el código de operación, es decir, al cargarse y ejecutarse el código de operación, todos los recursos que se necesiten se encuentran dentro del  $\mu$ controlador, por lo cual no son necesarios accesos adicionales a la memoria para completar la instrucción.
6. **Direccionamiento Relativo.** El modo de direccionamiento relativo es utilizado únicamente por las instrucciones de branch. Si la condición de la instrucción de branch se cumple, se suma un valor offset de 8 bits al PC para formar una dirección efectiva. Estas son por lo general instrucciones de 2 bytes.

**4.1.4 Modos de operación.** El  $\mu$ controlador HC11 cuenta con 4 modos de operación directamente seleccionables mediante la combinación de los estados lógicos de los pines MODA y MODB. Después que se da una acción de reset, la CPU monitorea el estado de MODA y MODB para de esa forma arrancar al micro en el modo seleccionado. Los modos monochip y expandido son los modos normales, mientras que el TEST y el BOOTSTRAP son los modos especiales. De acuerdo a como se observa en la tabla, el pin MODB conmuta entre los modos normales y los especiales, así aplicando un nivel lógico alto en este pin se seleccionan los modos normales. El pin MODA por su parte selecciona entre los modos expandidos y los monochip.

MODB	MODA	MODO
1	1	Monochip
1	0	Expandido
0	1	Bootstrap
0	0	Test

Tabla 4.1

- **Modo Monochip.** En este modo los puertos B y C, así como los pines STRA y SRTB se encuentran disponibles como puertos y pines de transferencia de datos respectivamente. En estas condiciones, todos los accesos a memoria se realizan dentro del dispositivo. Cuando el micro se energiza comienza ejecutando el programa residente en la memoria EEPROM, utilizando los 256 bytes de memoria RAM interna como memoria de trabajo.
- **Modo Extendido.** Este modo de operación permite acceder memoria externa así como periféricos vía un bus de datos/dirección multiplexado en tiempo. De acuerdo a los datos desplegados en la tabla 4.1, este modo se selecciona poniendo en 1 a los dos pines de selección (MODA y MODB). Al salir el  $\mu$ controlador de una situación de reset, extrae un vector de inicio el cual se ubica en la memoria externa. De esta manera es posible colocar el programa en cualquier parte del mapa de memoria.
- **Modo Test.** El modo especial Test es una variación del modo expandido, y es principalmente usado por Motorola en sus pruebas de producción; sin embargo, está disponible para programar el registro CONFIG, programar datos de calibración en la EEPROM, y como soporte en la emulación y la depuración durante el desarrollo.
- **Modo Bootstrap.** Cuando el  $\mu$ controlador es reseteado en el modo bootstrap, una pequeña cantidad de ROM integrada en el dispositivo es habilitada en las direcciones \$BF00 - \$BFFF. Esta ROM contiene un programa de carga (bootloader) y un conjunto especial de vectores de interrupción y de reset. El micro obtiene el vector de reset, para en seguida ejecutar el bootloader. En estas condiciones, el HC11 queda preparado para recibir el dato \$FF, mediante el cual se le indica que comienza la transferencia de un programa, que quedará almacenado en la RAM interna. Una vez que ha sido transferido el programa, comienza su ejecución desde la localidad \$0000.

**4.1.5 Mapa de Memoria.** El modo de operación determina el mapa de memoria y cualquier configuración puede ser establecida considerando tanto memoria interna como externa. La figura 4.4, la cual ilustra el mapa de memoria correspondiente a cada uno de los cuatro modos de operación. Las localidades de memoria para los recursos incorporados en el micro son las mismas tanto para los modos expandidos como para los modos monochip. Los bits de control del registro CONFIG permiten habilitar y deshabilitar a las memorias EPROM o EEPROM del mapa. Los 256 bytes de RAM se localizan en la dirección \$0000, pero pueden ser colocados en otros segmentos del mapa escribiendo un valor apropiado al registro INIT. El bloque de registros de 64 bytes se encuentra en la dirección \$1000 después del reset, e igualmente puede ser reposicionada en alguna de varias franjas de 4 kbytes mediante el registro INIT.

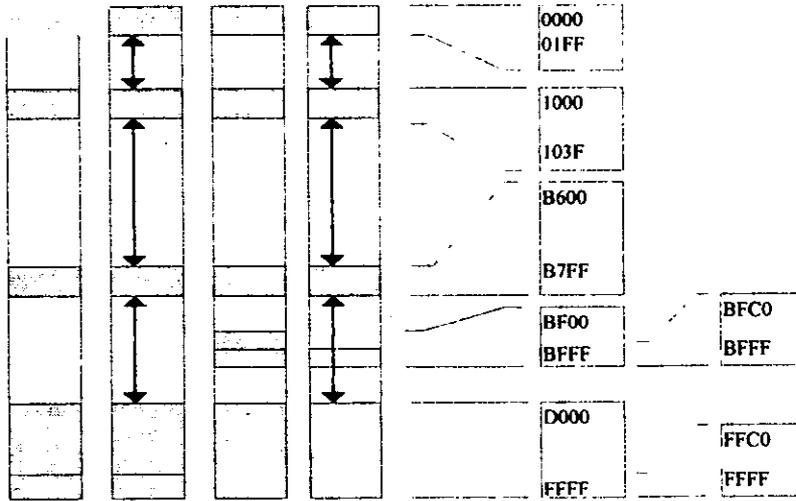


Fig. 4.4 Mapa de Memoria

\$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
\$1001									RESERV.
\$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC
\$1003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
\$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
\$1005	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	PORTCL
\$1006									RESERV.
\$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
\$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
\$1009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
\$100A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
\$100B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
\$100C	OCM17	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OCM1
\$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
\$100E	BIT 15	14	13	12	11	10	9	8	TCNT (H)
\$100F	BIT 7	6	5	4	3	2	1	0	TCNT (L)

Selección y descripción del microcontrolador

\$1010	BIT 15	14	13	12	11	10	9	8	TIC1 (H)
\$1011	BIT 7	6	5	4	3	2	1	0	TIC1 (L)
\$1012	BIT 15	14	13	12	11	10	9	8	TIC2 (H)
\$1013	BIT 7	6	5	4	3	2	1	0	TIC2 (L)
\$1014	BIT 15	14	13	12	11	10	9	8	TIC3 (H)
\$1015	BIT 7	6	5	4	3	2	1	0	TIC3 (L)
\$1016	BIT 15	14	13	12	11	10	9	8	TOC1 (H)
\$1017	BIT 7	6	5	4	3	2	1	0	TOC1 (L)
\$1018	BIT 15	14	13	12	11	10	9	8	TOC2 (H)
\$1019	BIT 7	6	5	4	3	2	1	0	TOC2 (L)
\$101A	BIT 15	14	13	12	11	10	9	8	TOC3 (H)
\$101B	BIT 7	6	5	4	3	2	1	0	TOC3 (L)
\$101C	BIT 15	14	13	12	11	10	9	8	TOC4 (H)
\$101D	BIT 7	6	5	4	3	2	1	0	TOC4 (L)
\$101E	BIT 15	14	13	12	11	10	9	8	TI4/O5 (H)
\$101F	BIT 7	6	5	4	3	2	1	0	TI4/O5 (L)
\$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
\$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
\$1022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	TMSK1
\$1023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	TFLG1
\$1024	TOI	RTI	PAOVI	PAI	0	0	PR1	PR2	TMSK2
\$1025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
\$1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0	PACTL
\$1027	BIT 7	6	5	4	3	2	1	0	PACNT
\$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
\$1029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
\$102A	BIT 7	6	5	4	3	2	1	0	SPDR
\$102B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
\$102C	R8	T8	0	M	WAKE	0	0	0	SCCR1
\$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
\$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
\$102F	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDR
\$1030	CCF	0	SCAN	MULT	CD	CC	CB	CA	ADCTL

Selección y descripción del microcontrolador

\$1031	BIT 7	6	5	4	3	2	1	0	ADR1
\$1032	BIT 7	6	5	4	3	2	1	0	ADR2
\$1033	BIT 7	6	5	4	3	2	1	0	ADR3
\$1034	BIT 7	6	5	4	3	2	1	0	ADR4
\$1035	0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT
\$1036									Reservado
\$1037									Reservado
\$1038									Reservado
\$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
\$103A	BIT 7	6	5	4	3	2	1	0	COPRST
\$103B	ODD	EVEN	ELAT	BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
\$103C	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
\$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
\$103E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1
\$103F	0	0	0	0	NOSEC	NOCOP	ROMON	EEON	CONFIG

De los registros mostrados en la tabla se pueden tomar algunos para realizar configuraciones del sistema, tales como la organización del mapa de memoria y el modo de operación. Para este último caso se considera el registro localizado en la dirección hexadecimal \$103C, el cual recibe el nombre de HPRIO. En el, existen dos bits SMOD y MDA, cuyos estados lógicos se relacionan de manera directa con los estados lógicos correspondientes a los pines MODB y MODA. El bit SMOD refleja el estado negado del pin MODB, mientras que el bit MDA presenta el mismo estado lógico del pin MODA. Al regresar de un estado de reset, el microcontrolador actualiza estos bits del registro HPRIO, los cuales no pueden ser modificados sino únicamente cuando el dispositivo se encuentra operando en alguno de los modos especiales, de esta forma es posible cambiar de un modo especial a un modo normal simplemente poniendo en 0 el bit SMOD. A continuación se presenta la tabla de verdad en la que se muestran las distintas combinaciones pertenecientes a cada modo de operación.

**\$103C HPRIO.**

7							0
RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0

MODB	MODA	Modo	SMOD	MDA
1	0	Monochip	0	0
1	1	Expandido	0	1
0	0	Bootstrap	1	0
0	1	Test	1	1

Anteriormente se mencionó la característica de poder ubicar a la memoria RAM y el bloque de registros en alguna de 16 distintas franjas de 4k. Esto es posible mediante el uso del registro INIT cuyos bits se muestran a continuación. Como puede verse existen 2 juegos de 4 bits, los 4 más significativos pertenecen a la memoria RAM, y los 4 restantes al bloque de registros especiales. De esta forma se pueden seleccionar cualquiera de las siguientes combinaciones.

RAM [3:0]	Dirección	REG [3:0]	Dirección
0000	\$0000 - \$01FF	0000	\$0000 - \$003F
0001	\$1000 - \$11FF	0001	\$1000 - \$103F
0010	\$2000 - \$21FF	0010	\$2000 - \$203F
0011	\$3000 - \$31FF	0011	\$3000 - \$303F
0100	\$4000 - \$41FF	0100	\$4000 - \$403F
0101	\$5000 - \$51FF	0101	\$5000 - \$503F
0110	\$6000 - \$61FF	0110	\$6000 - \$603F
0111	\$7000 - \$71FF	0111	\$7000 - \$703F
1000	\$8000 - \$81FF	1000	\$8000 - \$803F
1001	\$9000 - \$91FF	1001	\$9000 - \$903F
1010	\$A000 - \$A1FF	1010	\$A000 - \$A03F
1011	\$B000 - \$B1FF	1011	\$B000 - \$B03F
1100	\$C000 - \$C1FF	1100	\$C000 - \$C03F
1101	\$D000 - \$D1FF	1101	\$D000 - \$D03F
1110	\$E000 - \$E1FF	1110	\$E000 - \$E03F
1111	\$F000 - \$F1FF	1111	\$F000 - \$F03F

\$103D INIT.

7				0			
RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0

## 4.2 Memoria EEPROM.

El microcontrolador MC68HC11A1 cuenta con un bloque de 512 bytes de memoria EEPROM la cual comienza en la dirección \$B600. Esta es la memoria principal cuando se trabaja en modo monochip, en el modo expandido se utiliza como memoria no volátil de escritura/lectura. No existe ninguna variación en la forma en la que se obtienen datos de ella en comparación con otro tipo de memorias (RAM, ROM o EPROM). No así para el proceso de escritura, para el cual se contempla un intervalo de 10 ms para lograr grabar el dato. A continuación se presentan los registros relacionados con esta memoria así como los procedimientos para escribir y borrar datos de ella.

**\$103F CONFIG**

7							0
0	0	0	0	NOSEC	NOCOP	ROMON	EEON

Mediante el bit 0 de este registro se habilita (EEON=1) o se deshabilita (EEON=0) a la memoria EEPROM dentro del mapa de memoria del sistema.

**\$103B PPROG**

7							0
ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM

Los 2 bits más significativos se emplean únicamente en el modo TEST por lo que para nuestro caso no interesa una explicación sobre su función.

El Bit BYTE se emplea en las funciones de borrado para indicar si se requiere borrar un simple byte (BYTE=1) o más de uno (BYTE=0).

Con el bit ROW se especifica si se borrará un bloque de 16 bytes (ROW=1) o el total de la memoria (ROW=0).

Con ERASE=1 se habilita el modo de borrado, mientras que con ERASE=0 permanecerá deshabilitado.

El bit EELAT configura los buses de datos y dirección para modo de programación de EEPROM cuando tiene un valor lógico 1, mientras que configura los buses para lectura y escritura normales cuando vale 0.

Mediante el bit EEGPM se activa (EEPGM=1) o desactiva (EEPGM=0) el voltaje de programación.

**Algoritmo de Programación.**

1. - Se activa el modo de programación poniendo a '1' el bit EELAT.
2. - Se envía el dato a la localidad requerida.
3. - Posteriormente se activa el voltaje de programación con el bit EEGPM puesto en '1' durante 10 ms.
4. - Finalmente se desactiva el voltaje de programación poniendo en '0' el bit EEGPM.

Existen tres tipos de borrado : de byte, de renglón, y borrado total. El primero como su nombre lo indica borra únicamente el byte seleccionado, mientras que con los otros se borran 16 y 512 bytes respectivamente.

**Algoritmo de borrado de 1 byte.**

1. - Se ponen en '1' los bits BYTE, ERASE y EELAT para indicar modo de borrado de un byte de la memoria EEPROM.
2. - Enviar cualquier dato a la localidad que será borrada.
1. - Poner a '1' los bits BYTE, ERASE , EELAT y EEPGM durante 10 ms.
2. - Desactivar el voltaje de programación poniendo en '0' el bit EEPGM.

**Algoritmo de borrado de renglón.**

1. - Se ponen en '1' los bits ROW, ERASE y EELAT para indicar modo de borrado de renglón en la memoria EEPROM.
2. - Enviar cualquier dato a la localidad que será borrada.
3. - Poner a '1' los bits ROW, ERASE , EELAT y EEPGM durante 10 ms.
4. - Desactivar el voltaje de programación poniendo en '0' el bit EEPGM.

**Algoritmo de borrado total de la memoria .**

1. - Se pone en '1' el bit EELAT.
2. - Se envía un dato a cualquier dirección de la EEPROM.
3. - Poner en '1' los bits EELAT y EEPGM durante 10 ms.
4. - Desactivar el voltaje de programación poniendo en '0' el bit EEPGM.

En forma adicional a estos registros, existe otro que tiene como finalidad proteger bloques o la totalidad de la memoria EEPROM de escrituras o de acciones de borrado indeseables. Para este caso se emplea el registro BPROT localizado en la dirección \$1035. Los bits activos de este registro se ponen en '1' después del reset y pueden cambiarse a '0' únicamente durante los primeros 64 ciclos E. Cuando esos bits son puestos a cero, la sección de EEPROM asociada puede ser programada o borrada.

**\$ 1035 BPROT.**

7							0
0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0

Bit	Bloque Protegido	Tamaño del bloque
BPRT0	\$B600 - \$B61F	32 bytes
BPRT1	\$B620 - \$B65F	64 bytes
BPRT2	\$B660 - \$B6DF	128 bytes
BPRT3	\$B6E0 - \$B7FF	288 bytes

### 4.3 Interfase Serial de Comunicaciones (SCI).

Esta interfase no es otra cosa que una UART (Universal Asynchronous Receiver-Transmitter) ó Receptor-Transmisor asincrono, el cual es uno de dos subsistemas seriales de entrada/salida con los que cuenta el  $\mu$ controlador. Esta interfase maneja el formato estándar de no retorno a cero NRZ, o sea: un bit de comienzo, ocho o nueve bits de datos, y un bit de paro. Así también cuenta con varias tazas de velocidad. Tanto el receptor como el transmisor son independientes el uno del otro, pero manejan las mismas velocidades y el mismo formato de datos, el cual requiere las siguientes condiciones:

1. - Una línea dormida en estado alto antes de la recepción o transmisión de un mensaje.
2. - Un bit de comienzo, el cual debe ser un cero lógico recibido o transmitido, que indique el comienzo de un mensaje.
3. - Datos que sean transmitidos y recibidos (el bit menos significativo primero).
4. - Un bit de paro, un 1 lógico, usado para indicar el fin de una trama (una trama consiste de un bit de comienzo, un caracter de 8 ó 9 bits, y un bit de paro).
5. - Un break (definido como la transmisión o recepción de un número determinado de ceros lógicos durante varias tramas)

**4.3.1 Transmisión.** El transmisor de la SCI incluye un registro paralelo de datos conocido como SCDR, así como un registro serial de desplazamiento. Los contenidos del registro serial pueden únicamente ser escritos a través del registro SCDR. Esta doble operación permite desplazar un caracter hacia afuera del  $\mu$ controlador mientras se escribe el próximo caracter a ser transmitido. La salida del registro serial de desplazamiento es aplicada por la línea TxD durante el tiempo que dure la transmisión o durante el tiempo que permanezca habilitado el bit TE del registro SCCR2.

**4.3.2 Recepción.** Durante la operación de recepción, se invierte la secuencia de transmisión, es decir, el registro serial de desplazamiento recibe datos y los transfiere al registro paralelo SCDR en forma de un byte completo. Esta doble operación permite que continúen entrando datos bits por la línea RxD, mientras se dispone de datos para ser leídos en el registro paralelo. Un avanzado esquema de recuperación de datos distingue información válida y logra separar el ruido que puede ir incluido en el tren de datos. La entrada de datos es selectivamente muestreada para detectar datos recibidos, y un circuito de validación determina el valor e integridad de cada bit.

- **Característica de Activación.** La característica de activación reduce los servicios de cabecera de la SCI en múltiples sistemas de recepción. El software para cada receptor evalúa el primer carácter de cada mensaje. El receptor es colocado en modo *wakeup* (despierto) poniendo a '1' el bit RWU del registro SCCR2. Aunque es posible poner en '0' este bit mediante programación, usualmente se prefiere hacerlo mediante hardware. En el momento en el que un mensaje comienza, la lógica alerta a los receptores que se encuentran dormidos (en espera de datos) a despertar y evaluar el carácter inicial del nuevo mensaje.

Se dispone de dos métodos de despertado (activación): línea dormida y marca-dirección. En el método de línea dormida un receptor dormido despierta tan pronto como se rompe la condición de sueño en la línea RxD. En el método de marca-dirección, basta un '1' lógico en el bit más significativo de un carácter para despertar a todos los receptores dormidos.

- **Detección de Errores.** Existen tres circunstancias que pueden provocar que el HC11 produzca igualmente 3 condiciones de error, esto mediante 3 bits (OR, NF y FE) en el registro de estados de comunicación serial SCSR. El bit de error de sobreescritura ó *overrun* (OR) toma el valor lógico '1' cuando el siguiente byte está listo para ser transferido del registro serial de desplazamiento al registro paralelo SCDR y éste último no ha sido vaciado o leído ( en esta condición el bit RDRF está puesto en '1'). Cuando ocurre un error de este tipo, el dato que originó la sobreescritura se pierde. El bit OR es puesto a '0' automáticamente cuando se realiza una lectura del registro SCDR.

El bit de la bandera de ruido (NF) se pone a '1' si se detecta ruido en cualquiera de los bits recibidos, incluyendo a los bits de comienzo y de paro. Este bit se pone en '0' automáticamente cuando se realiza una lectura del registro SCSR seguida por una lectura del registro SCDR.

Cuando no se detecta un bit de paro en la trama, el bit de error de trama (FE) es puesto a '1'. FE toma el valor de '1' al mismo tiempo que lo hace el bit RDRF. Si el byte recibido causa tanto error de trama como error de sobreescritura, el procesador solo reconoce el error de sobreescritura. El bit FE es puesto en '0' cuando se realiza una lectura del registro SCSR, seguida de una lectura del registro SCDR.

#### 4.3.3 Registros de datos de la comunicación serial.

**\$102C SCCR1** (registro 1 del control de la comunicación serial).

7							0
R8	T8	0	M	WAKE	0	0	0

**R8:** Recibe dato del bit 8. Si el bit **M=1**, **R8** guarda el noveno bit del carácter recibido.

**T8:** Transmite el bit 8. Si el bit **M=1**, **T8** guarda el noveno bit que será transmitido en el carácter.

**M:** Selector de modo de carácter. Si **M=0**, se elige el formato de 8 bits  
Si **M=1**, se elige el formato de 9 bits

**WAKE:** Selector de tipo de activación. Si **WAKE=1**, se elige el método de activación de línea dormida.  
Si **WAKE=1**, se elige el método de activación de marca-dirección.

**\$102D SCCR2** (registro 2 del control de la comunicación serial).

7							0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

**TIE:** Habilita la interrupción de transmisión. Si **TIE=1**, la SCI solicita una interrupción siempre y cuando el bit de bandera **TDRE** esté puesto en 1.

**TCIE:** Habilita la interrupción de transmisión completa. Si **TCIE=1**, la SCI solicita una interrupción siempre y cuando el bit de bandera **TC** esté puesto en 1.

**RIE:** Habilita la interrupción de recepción. Si **RIE=1**, la SCI solicita una interrupción siempre y cuando el bit **RDRF** o **OR** esté puesto en 1.

**ILIE:** Habilitación de interrupción de línea dormida. Si **ILIE=1**, la SCI solicita una interrupción siempre y cuando el bit de bandera **IDLE** esté puesto en 1.

**TE:** Habilitación de transmisión.

**RE:** Habilitación de recepción.

**RWU:** Control de activación del receptor. con **RWU=1**, se reconocen los métodos de activación así como las interrupciones correspondientes.

**SBK:** Envía un Break.

**\$ 102E SCSR** (Registro de estatus de la SCI).

7							0
TDRE	TC	RDRF	IDLE	OR	NF	FE	0

**TDRE:** Bandera de registro de transmisión vacío. Si **TDRE=1**, el SCDR está vacío.

**TC:** Bandera de transmisión completa. Si **TC=1** el transmisor está inactivo, de lo contrario está dándose una operación de transmisión.

**RDRF:** Bandera de registro de datos recibidos lleno. Si **RDRF=1**, existe un carácter listo para ser leído del registro SCDR.

**IDLE:** Bandera de detección de línea dormida. Si **IDLE=1**, la línea RxD está dormida (inactiva).

**OR:** Bandera de error de sobreescritura. Si **OR=1**, un error de sobreescritura ha sido detectado.

**NF:** Bandera de error de ruido.

**FE:** Bandera de error de trama.

**\$102B BAUD** (Registro de fijación de tasa de velocidad para transmisión/recepción).

7							0
TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0

**TCLR:** Limpia los contadores de la tasa de baudios (válido en el modo de operación TEST)

**RCKB:** Checa el reloj de la tasa de baudios (válido en el modo de operación TEST)

**SCP[1:0]:** Selectores del pre-escalador de tasa de baudios

SCP [1:0]		Dividir al reloj interno por:
0	0	1
0	1	3
1	0	4
1	1	16

**SCR[2:0]:** Selecciona la tasa de baudios de la SCI.

SCR[2:0]			Divide al pre-escalador por:	Taza mayor de baudios		
				4800	9600	38.4K
0	0	0	1	4800	9600	38.4K
0	0	1	2	2400	4800	19.2K
0	1	0	4	1200	2400	9600
0	1	1	8	600	1200	4800
1	0	0	16	300	600	2400
1	0	1	32	150	300	1200
1	1	0	64	--	150	600
1	1	1	128	--	--	300

## 4.4 Sistema de Temporización.

El sistema de temporización del MC68HC11A1 está compuesto de cinco cadenas divisoras de reloj. La cadena principal incluye un contador libre de 16 bits, el cual está manejado por un pre-escalador programable. El pre-escalador principal proporciona uno de cuatro tasas de frecuencia, para manejar el contador de 16 bits. Todas las actividades del sistema principal de temporización están en referencia al contador libre. El contador comienza su cuenta desde \$0000 después del reset y continúa hasta que alcanza el valor \$FFFF, después del cual pasa a \$0000 nuevamente, lo que provoca que se active una bandera de sobreflujo.

Básicamente son 5 las funciones relacionadas con la sección de temporización: Captura de entradas (IC), Comparación de salidas (OC), Acumulación de pulsos (PAI), Interrupción de Tiempo Real (RTI), y la función de chequeo de operación apropiada de cómputo (COP). Cada uno de los tres canales de IC cuenta con su propio registro de 16 bits, al igual que cada uno de los cinco canales de OC. Así también, cada función cuenta con sus propias banderas y vectores de interrupción. El Acumulador de pulsos contiene un contador de 8 bits y una lógica de selección de etapa.

**4.4.1 Captura de Entradas.** La función de captura de entradas, determina el tiempo que dura un evento externo, mediante la grabación de la cuenta actual del contador libre en el momento en que se detecta al evento en el pin respectivo. Mediante software se almacenan los instantes de tiempo en que se dieron los sucesos para de esa forma poder conocer tanto la periodicidad como la duración de ellos. Por ejemplo, guardando los tiempos de dos etapas sucesivas de una señal entrante, un programa puede determinar el periodo y el ancho de pulso de la señal. Para medir el periodo, se capturan dos etapas sucesivas de la misma polaridad. Para medir anchos de pulso, se capturan dos etapas sucesivas de polaridad alterna. Los bits de control y de estado que implementan las funciones de captura, son contenidos por los registros PACTL, TCTL2, TMSK1, y TFLG1.

### \$1021 TCTL2 (2o. control del temporizador)

7							0
EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A

EDGxB	EDGxA	Configuración
0	0	captura deshabilitada
0	1	captura etapa de ascenso
1	0	captura etapa de descenso
1	1	captura cualquier etapa

Se usan estos bits de control para programar las etapas y sus polaridades que serán identificadas en las entradas de captura. Cada una de estas funciones de captura puede ser configurada en forma independiente de tal forma que puedan detectar únicamente etapas de ascenso, etapas de descenso, alguna de las dos, o deshabilitar la función.

Cuando una etapa ha sido detectada y sincronizada, el valor del contador libre es transferido al par de registros de captura, los cuales dicho sea de paso no son afectados por las acciones del reset.

Los valores de los registros de captura pueden ser leídos como números de 16 bits o de 8 bits. La lectura del byte más significativo inhibe una nueva captura durante un ciclo de bus (E). Debido a que se cuenta con tres registros de captura, igualmente se tienen tres pares de registros identificados como TIC1, TIC2 y TIC3 respectivamente.

#### \$1010 - \$1015 TIC1-TIC3 (Registros de Captura de Entrada).

\$1010 

BIT 15	14	13	12	11	10	9	8
--------	----	----	----	----	----	---	---

 TIC1 (H)

\$1011 

BIT 7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---

 TIC1 (L)

\$1012 

BIT 15	14	13	12	11	10	9	8
--------	----	----	----	----	----	---	---

 TIC2 (H)

\$1013 

BIT 7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---

 TIC2 (L)

\$1014 

BIT 15	14	13	12	11	10	9	8
--------	----	----	----	----	----	---	---

 TIC3 (H)

\$1015 

BIT 7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---

 TIC3 (L)

**4.4.2 Comparadores de salida (OC).** Los comparadores de salida se emplean para programar acciones en un instante de tiempo dado, esto cuando el contador libre alcanza una cuenta previamente especificada. Para cada uno de los 5 OC's, existe un registro de 16 bits y un comparador igual de 16 bits. El valor en el registro es comparado en cada ciclo de bus con el valor actual del contador libre. Cuando el dato del registro comparador es igual a la cuenta del contador, se activa una bandera de estado, para darse entonces una acción previamente programada en el pin correspondiente al OC que está siendo utilizado.

Para producir un pulso de determinada duración, se escribe un valor en uno de los OC's, valor que representará el tiempo en el que deberá dar comienzo el pulso. El circuito de comparación es configurado para poner un estado lógico '1' o '0' dependiendo de la polaridad que se desee en el pulso. Después de que ocurre la igualación de valores, se configura el pin para ahora tomar un estado lógico contrario al que se tiene actualmente. Una vez hecho esto se deposita en el registro de comparación el nuevo dato que determinará el ancho del pulso. Para generar un tren de pulsos de una frecuencia y ciclo útil específicos, se repite el procedimiento descrito.

Existen 5 OC's: TOC1, TOC2, TOC3, TOC4 y TOC5, cada uno de los cuales se inicializa con un valor de \$FFFF después del reset. Cuando se da una comparación exitosa de alguno de estos registros, se activa una bandera de estado ubicada en el registro 1 de banderas. Si la interrupción correspondiente se encuentra habilitada (mediante las máscaras del registro TMSK1), se genera entonces una interrupción. La acción de los pines de los OC's es determinada por los pares de bits Omx y OLx en el registro TCTL1.

El OC1 es diferente de los cuatro OC's restantes en que una comparación exitosa del OC1 puede afectar a cualquiera de los cinco pines de comparadores. La acción de salida del OC1 está determinada por bits de control de los registros OC1D y OC1M. OC1M especifica cual pin o pines serán afectados por la acción del OC1, mientras que OCD1 determinará el tipo de acción (un alto o un bajo) que se presentará en los pines programados.

**\$1016 - \$101D TOC1 - TOC5 (Registros de comparación de salida).**

**\$1016** BIT 15 14 13 12 11 10 9 8 TOC1 (H)

**\$1017** BIT 7 6 5 4 3 2 1 0 TOC1 (L)

**\$1018** BIT 15 14 13 12 11 10 9 8 TOC2 (H)

**\$1019** BIT 7 6 5 4 3 2 1 0 TOC2 (L)

**\$101A** BIT 15 14 13 12 11 10 9 8 TOC3 (H)

**\$101B** BIT 7 6 5 4 3 2 1 0 TOC3 (L)

**\$101C** BIT 15 14 13 12 11 10 9 8 TOC4 (H)

**\$101D** BIT 7 6 5 4 3 2 1 0 TOC4 (L)

**\$101E** BIT 15 14 13 12 11 10 9 8 TOC5 (H)

**\$101F** BIT 7 6 5 4 3 2 1 0 TOC5 (L)

**\$100C OC1M (Máscara de los OC1).**

7						0
OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0

**OC1M7-OC1M3:** Máscaras de los OC1. Si **OC1Mx=1**, se habilita la acción de OC1 en el pin respectivo

**\$100D OC1D (Dato del OC1).**

7						0
OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0

**OC1D7-OC1D3:** Acción que tomará lugar en una comparación exitosa de OC1. Si **OC1Dx=1**, el estado

lógico presente en estos bits se reflejará automáticamente en el pin correspondiente una vez que se ha dado una comparación exitosa de OC1.

**\$1020 TCTL1 (control 1 de los temporizadores).**

7						0
OM2	OL2	OM3	OL3	OM4	OL4	OM5
						OL5

Este par de bits determinan el tipo de acción que se tendrá en los pines de los OC's.

OMx	OLx	Acción
0	0	Sin acción
0	1	Conmutación de estados
1	0	Estado lógico 0
1	1	Estado lógico 1

• **Registros de banderas e interrupciones.**

**\$1023 TFLG1** (registro 1 de banderas de las interrupciones OC's e IC's).

7							0
OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F

**\$1022 TMSK1** (registro 1 de habilitación de interrupciones de OC's e IC's).

7							0
OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I

**\$1024 TMSK2** (registro 2 de habilitación de interrupciones de temporización)

7							0
TOI	RTII	PAOVI	PAII	0	0	PR1	PR0

**TOI:** Habilidad de la interrupción de sobreescritura del contador libre.

**RTII:** Habilidad de la interrupción de Tiempo Real.

**PAOVI:** Habilidad de la interrupción de sobreescritura del Acumulador de Pulsos.

**PAII:** Habilidad de la interrupción de la etapa de entrada del Acumulador de Pulsos.

**PR[1:0]:** Selector del pre-escalador del temporizador.

**\$1025 TFLG2** (registro 2 de banderas del sistema de temporización).

7							0
TOF	RTIF	PAOVF	PAIF	0	0	0	0

**TOF:** Bandera de la interrupción de sobreescritura del contador libre.

**RTIF:** Bandera de la interrupción de Tiempo Real.

**PAOVF:** Bandera de sobreescritura del Acumulador de Pulsos.

**PAIF:** Bandera de la interrupción de etapa de entrada del Acumulador de Pulsos.

**4.4.3 Interrupción de Tiempo Real (RTI).** La interrupción de tiempo real es utilizada para generar interrupciones de hardware en intervalos de tiempo fijos, es controlada y configurada mediante dos bits (RTR1 y RTR2) del registro de control del acumulador de pulsos (PACTL). La fuente de reloj para la función RTI es un reloj libre que no puede ser detenido o interrumpido más que mediante un reset. Este reloj es el que determina la duración de los intervalos entre RTI sucesivas, los cuales son independientes de los retardos asociados con la puesta a cero de las banderas y los servicios relacionados a ellas. Por esta razón, un periodo de una RTI comienza desde el retardo previo, y no a partir de cuando fue puesta a cero la bandera RTIF.

Una práctica común de software es organizar las rutinas que componen el programa en secuencia de las subrutinas más utilizadas. La longitud de tiempo requerido para ejecutar cada una de las rutinas es una variable, la cual depende de cuanto tiene que hacer cada una, pero en el peor de los casos debe conocerse el tiempo que consume la ejecución de la totalidad de las subrutinas. Después de haberse ejecutado todas, se introduce un retardo de tiempo, el cual dura hasta que se detecta una señal de referencia. Al detectarse esta señal, se realiza un brinco a la parte inicial del conjunto de rutinas para volver a ejecutarlas. Conociendo el tiempo entre señales de referencia sucesivas, una rutina puede medir el tiempo real que consumen las subrutinas

**4.4.4 Acumulador de Pulsos.** El  $\mu$ controlador MC68HC11A1 cuenta con un contador de 8 bits que puede ser configurado para operar ya sea como un simple contador de eventos, o como contador disparado por tiempo, esto dependiendo del estado del pin PAMOD en el registro PACTL. En el modo de conteo de eventos, el contador es preparado para incrementar su cuenta mediante la condición de un pin externo. En el modo de disparo por tiempo, el contador es manejado por un reloj libre de frecuencia E/64, pero solo mientras el pin externo PAI esté activado.

El control de este contador se realiza a través de 4 bits del registro PACTL, tal y como se muestra.

**\$1026 PACTL** (control del acumulador de pulsos).

7							0
DDRA7	PAEN	PAMOD	PEDGE	0	0	RTR1	RTR0

**DDRA7:** Control de dirección de dato para el bit 7 del puerto A. El acumulador de pulsos utiliza el bit 7 del puerto A como entrada PAI, pero este pin puede ser utilizado también como pin de E/S de propósito general o como OC1.

**PAEN:** Sistema de acumulación de pulsos habilitado.

**PAMOD:** Modo del acumulador de pulsos. Si **PAMOD=1**, el PAI se incrementa por disparo de tiempo.  
Si **PAMOD=0**, el PAI se incrementa por evento.

**PEDGE:** Este bit toma diferentes funciones dependiendo del estado del bit **PAMOD**, tal como se muestra en la siguiente tabla.

PAMOD	PEDGE	Acción
0	0	incremento por etapa de descenso en el pin PAI
0	1	incremento por etapa de ascenso en el pin PAI
1	0	un cero en el pin PAI inhibe el conteo
1	1	un uno en el pin PAI inhibe el conteo

**\$1027 PACNT** (cuenta del acumulador de pulsos).

7							0
BIT 7	6	5	4	3	2	1	0

## 4.5 CONVERTIDOR ANALOGICO-DIGITAL

El 68HC11 cuenta con un convertidor A/D de aproximaciones sucesivas, el cual utiliza una técnica de redistribución de carga capacitiva para realizar la conversión de la señal analógica a un valor digital. El sistema se compone de 4 entradas analógicas multiplexadas. No se requiere de circuitos externos de muestreo y retención, debido a la técnica de redistribución utilizada. La temporización del convertidor puede ser sincronizada con el sistema de reloj del bus (E), o con un reloj interno tipo RC. Este reloj interno es recomendado para frecuencias de 750 kHz e inferiores. El sistema consiste de cuatro bloques funcionales: multiplexor, convertidor analógico, control digital, almacenamiento de resultados.

- **Multiplexor.** El multiplexor selecciona una de 12 entradas de conversión. La selección de la entrada es controlada por el valor de los bits CD-CA en el registro ADCTL. Los cuatro pines del puerto E son entradas analógicas de dirección fija para el multiplexor. Este puerto puede ser también usado como entradas digitales, aunque se recomienda no realizar lecturas digitales del puerto mientras se lleva a cabo el muestreo.
- **Convertidor Analógico.** La conversión de una entrada analógica seleccionada por el multiplexor ocurre en este bloque. Este contiene un arreglo digital-analógico de capacitores (DAC), un comparador, y un registro de aproximaciones sucesivas. Cada conversión es una secuencia de ocho operaciones de comparación, comenzando por el bit más significativo. Cada comparación determina el valor de un bit en el registro de aproximaciones sucesivas. Por su parte el arreglo DAC realiza dos funciones, actúa como circuito de muestreo y mantenimiento, y también proporciona un voltaje de comparación al comparador durante cada comparación sucesiva.

El resultado de cada comparación es almacenado en el registro SAR. Cuando se acompleta una secuencia de conversión, los contenidos del SAR son transferidos al registro de resultados apropiado. Una bomba de carga proporciona el voltaje en las puertas de los switches analógicos en el multiplexor. La salida de la bomba de carga debe estabilizarse entre 7 y 8 volts antes de que el convertidor pueda ser usado. Este dispositivo es activado a través de un bit del registro OPTION, el cual se identifica como ADPU.

### \$1039 OPTION

7							0
ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR2

Cuando ADPU=1, se energiza la bomba de carga

- **Control Digital.** Todas las operaciones del convertidor A/D son controladas mediante los bits del registro ADCTL. Además de seleccionar la entrada analógica que será procesada, el registro también indica algunos estados de la conversión, así como controla el tipo de conversión: simple o continua. Finalmente los bits del ADCTL determinan si la conversión se realizará considerando uno o todos los pines del puerto.
- **Registros de Resultado.** El sistema de conversión A/D cuenta con cuatro registros de almacenamiento de resultados. Cada uno de estos registros puede ser accesado mediante el procesador en la CPU. La bandera de conversión completa (CCF) indica cuando un dato válido está presente en los registros de resultados.

La secuencia de conversión comienza un ciclo E después de que se realiza una escritura en el registro ADCTL. Un voltaje de entrada igual a  $V_{RL}$  convierte a \$00, mientras que un voltaje de entrada igual a  $V_{RH}$  convierte a \$FF, sin indicación de sobrescritura.

#### **Tipos de conversión.**

- **Operación Monocanal.** Hay dos tipos de operación monocanal: cuando el bit SCAN=0, el canal elegido es convertido 4 veces consecutivas, guardando el primer resultado en el registro ADR1, y así el cuarto en el ADR4. Después de que se llevó a cabo la cuarta conversión, todas las actividades de conversión se detienen hasta que un nuevo comando de conversión es enviado al registro ADCTL. En el segundo tipo, cuando SCAN=1, las conversiones continúan realizándose sobre el canal elegido almacenando la quinta conversión en el registro ADR1 reemplazando el valor correspondiente a la primera conversión, la sexta conversión se almacena en ADR2, y así sucesivamente.
- **Operación Multicanal.** Existen dos tipos de operación multicanal: cuando el bit SCAN=0, se realiza una conversión de cuatro canales al mismo tiempo. El primer resultado, correspondiente al canal uno, es almacenado en el registro ADR1, el resultado correspondiente al canal dos, es almacenado en el registro ADR2, y así sucesivamente hasta que queda almacenado el cuarto dato. Después de la cuarta conversión, todas las actividades de conversión son detenidas hasta

que se envía un nuevo comando de conversión al registro ADCTL. En el segundo tipo de operación multicanal, esto cuando **SCAN=1**, las conversiones continúan, quedando la quinta conversión (segunda del primer canal) en el registro ADR1 y así sucesivamente.

**\$1030 ADCTL**

7							0
CCF	0	SCAN	MULT	CD	CC	CB	CA

**CCF:** Bandera de conversión completa.

**SCAN:** Control de actualización continua. Si **SCAN=1**, la conversión es continua.  
Si **SCAN=0**, se realizan únicamente 4 conversiones.

**MULT:** Control de operación monocanal y multicanal. Si **MULT=1**, se realizan las conversiones considerando los 4 canales. Si **MULT=0**, se realizan las conversiones considerando un único canal.

**CD-CA:** Selección de canal.

CD	CC	CB	CA	Canal	Resultado en ADRx si MULT=1
0	0	0	0	AN0	ADR1
0	0	0	1	AN1	ADR2
0	0	1	0	AN2	ADR3
0	0	1	1	AN3	ADR4

**\$1031 - \$1034 ADR[4:1]**

**\$1031**

BIT 7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---

**ADR1**

**\$1032**

BIT 7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---

**ADR2**

**\$1033**

BIT 7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---

**ADR3**

**\$1034**

BIT 7	6	5	4	3	2	1	0
-------	---	---	---	---	---	---	---

**ADR4**

## CAPITULO 5:

### *Diseño del Algoritmo Difuso de control*

---

A continuación, se presenta el desarrollo del algoritmo difuso, mediante el cual se realizará el control basado en la arquitectura a la que se hace referencia en el capítulo anterior. Para simplificar tanto el desarrollo como la comprensión de este trabajo, se propone realizar algoritmos separados para cada etapa del diseño, de esta forma, se inicia con el desarrollo del algoritmo de difusificación.

#### 5.1 Creación de las Funciones Difusas.

**5.1.1 Selección de la forma de las funciones de membresía.** De acuerdo a lo citado en el capítulo 4, la selección de la forma de las funciones de membresía es uno de los pasos iniciales en el diseño de un controlador difuso. De las tres formas que comúnmente se manejan, como ya se mencionó, la triangular resulta la más conveniente, debido a que implica menor complejidad al representarla en un programa. Para su definición se eligen únicamente algunos puntos, de esta forma, y tal como se muestra en la figura 5.1, pueden utilizarse digamos 3 coordenadas, las cuales representen a los vértices del triángulo.

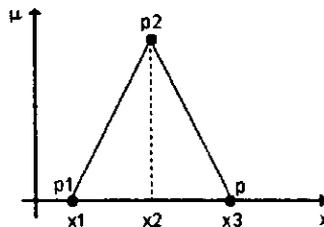


Fig. 5.1

**5.1.2 Selección de los Universos de Discurso y los valores de las variables difusas.** Los universos de discurso como se vió en el capítulo 2, no son otra cosa que los conjuntos de valores exactos en base a los cuales se definen los conjuntos difusos. En el caso del ejemplo de la figura 5.1, el universo de discurso lo forma el conjunto de valores pertenecientes al eje 'x'. Viéndolos desde el punto de vista de los conjuntos difusos, los universos de discurso representarían a la totalidad de los soportes de los distintos valores (conjuntos difusos) de una variable difusa dada. Cada universo, es definido en la mayoría de los casos en base a los conocimientos que se tienen del proceso bajo control. Por otra parte existen procesos que no requieren de un conocimiento experto para la definición de los universos de discurso, como sería nuestro caso. Por ejemplo, para el control de temperatura e iluminación con fines de confort, los universos de discurso podrían obtenerse a partir de lo que un número de personas (no necesariamente expertos) percibe como muy frío, medio caliente, etc, para el caso de la temperatura, o semi oscuro, muy luminoso, etc, para el caso de la iluminación.

El número de universos de discurso a definir, depende directamente de la cantidad de variables difusas con las que se va a trabajar, entre las cuales se contemplan tanto las variables de entrada como las de salida. Para el caso de nuestro control, se tienen contempladas únicamente dos variables de entrada y cuatro de salida, tal y como se muestra a continuación.

***Variables difusas para el control de la iluminación:***

- **Nivel\_Iluminación.** Variable de entrada para el control de la iluminación.
- **Error\_Iluminación.** Variable de entrada que indica la diferencia entre el nivel de iluminación actual y el nivel deseado.
- **Angulo1.** Variable de salida mediante la cual se controlará el disparo de un tiristor, en un ángulo dado de la onda senoidal.
- **Angulo2.** Variable similar a la anterior, sólo que definida en un intervalo más corto

***Variables difusas para el control de la Temperatura:***

- **Temperatura.** Variable de entrada para el control de temperatura.
- **Error\_Temperatura.** Variable de entrada que indica la diferencia entre la temperatura actual y la indicada en el Set-Point.
- **Angulo3.** Variable de salida para el control del disparo de un tiristor, con el objeto de variar la energía entregada a un calentador eléctrico
- **Angulo4.** Variable de salida para el control del disparo de un tiristor, con el objeto de variar la velocidad del motor de un ventilador.

Como se menciona en el capítulo 3, no existe ninguna receta, la cual nos pueda servir como guía para determinar el número óptimo de valores de una variable difusa. Por lo tanto queda a criterio del diseñador decidir el número de valores (conjuntos difusos) para cada variable. Los valores propuestos para este diseño se muestran a continuación.

**Nivel\_Illuminación = { Oscuro, Penumbra, Semi-Oscuro, Semi-Illuminado, Illuminado, Brillante}**

**Error\_Illuminación = { Pequeño, Regular, Grande}**

**Angulo1 = { Muy Abierto, Abierto, Semi-Abierto, Regular, Semi-Cerrado, Cerrado, Completo }**

**Angulo2 = { Muy Abierto, Abierto, Semi-Abierto, Regular, Semi-Cerrado, Cerrado, Completo }**

**Temperatura = { Fresco, Normal, Cálido, Caliente }**

**Error\_Temperatura = { Pequeño, Regular, Grande }**

**Angulo2 = { Muy Abierto, Abierto, Semi-Abierto, Regular, Semi-Cerrado, Cerrado, Completo }**

**Angulo3 = { Muy Abierto, Abierto, Semi-Abierto, Regular, Semi-Cerrado, Cerrado, Completo }**

Cada uno de estos conjuntos difusos, se definen dentro de intervalos de un universo de discurso tal y como se muestra en la figura 1, para el cual el intervalo es:  $[x1, x3]$ . A continuación, se muestran en forma de tabla los intervalos de definición para cada uno de los conjuntos difusos manejados en este diseño.

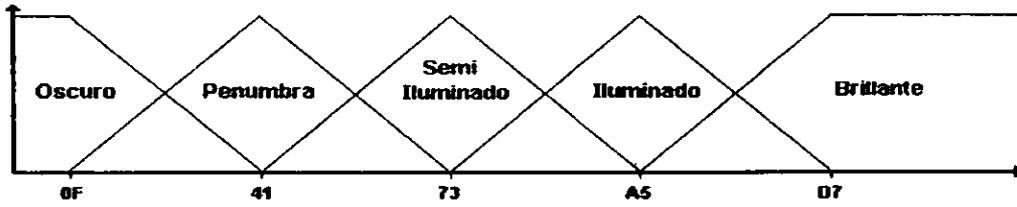
Conjunto	Intervalo 0-FF	Variable
Oscuro	00h-41h	Nivel_Illuminación
Penumbra	0Fh-73h	"
Semi-Illuminado	41h-A5h	"
Illuminado	73h-D7h	"
Brillante	A5h-FFh	"
Pequeño	00h-7Dh	Error_Illuminación
Regular	2Dh-CDh	"
Grande	7Dh-FFh	"
Muy Abierto	1Eh	Angulo1
Abierto	3Ch	"
Semi-Abierto	5Ah	"
Regular	78h	"

<b>Semi-Abierto</b>	5Ah	“
<b>Regular</b>	78h	“
<b>Semi-Cerrado</b>	96h	“
<b>Cerrado</b>	B4h	“
<b>Completo</b>	F0h	“
<b>Muy Abierto</b>	1Eh	<b>Angulo2</b>
<b>Abierto</b>	23h	“
<b>Semi-Abierto</b>	28h	“
<b>Regular</b>	2Dh	“
<b>Semi-Cerrado</b>	32h	“
<b>Cerrado</b>	37h	“
<b>Completo</b>	3Ch	“
<b>Fresco</b>	00h-69h	<b>Temperatura</b>
<b>Normal</b>	2Ch-A5h	“
<b>Cálido</b>	69h-E2h	“
<b>Caliente</b>	A5h-FFh	<b>Temperatura</b>
<b>Pequeño</b>	00h-7Dh	<b>Error_Iluminación</b>
<b>Regular</b>	2Dh-CDh	“
<b>Grande</b>	7Dh-FFh	“
<b>Muy Abierto</b>	1Eh	<b>Angulo3</b>
<b>Abierto</b>	3Ch	“
<b>Semi-Abierto</b>	5Ah	“
<b>Regular</b>	78h	“
<b>Semi-Cerrado</b>	96h	“
<b>Cerrado</b>	B4h	“
<b>Completo</b>	F0h	“
<b>Muy Abierto</b>	1Eh	<b>Angulo4</b>
<b>Abierto</b>	3Ch	“
<b>Semi-Abierto</b>	5Ah	“
<b>Regular</b>	78h	“
<b>Semi-Cerrado</b>	96h	“
<b>Cerrado</b>	B4h	“
<b>Completo</b>	F0h	“

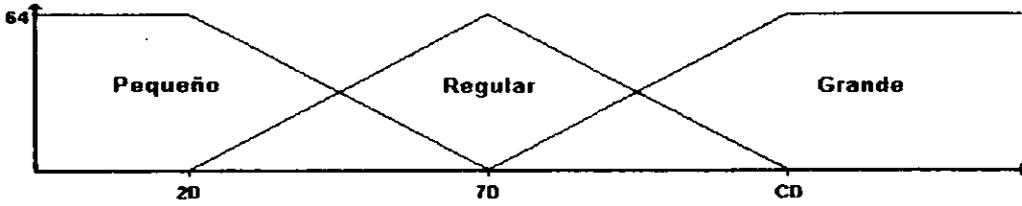
Tabla 5.1

En forma gráfica se tienen las siguientes representaciones de los conjuntos difusos expuestos en la tabla 5.1

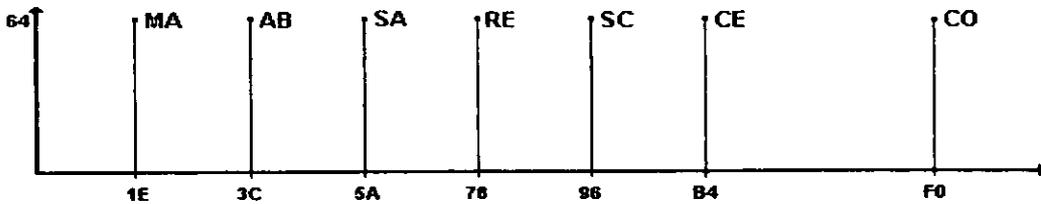
**Nivel\_Illuminación:**



**Error\_Illuminación:**



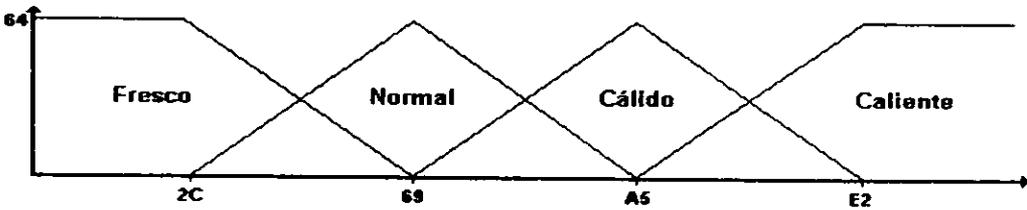
**Angulo1:**



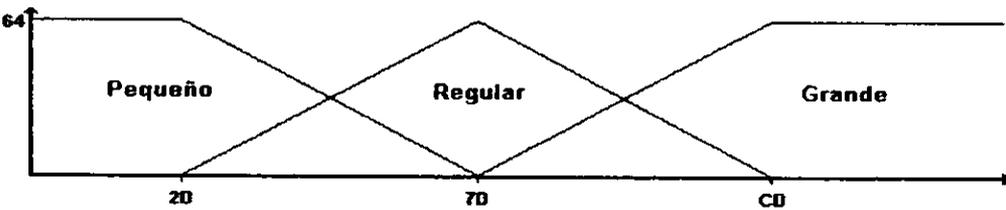
**Angulo2:**



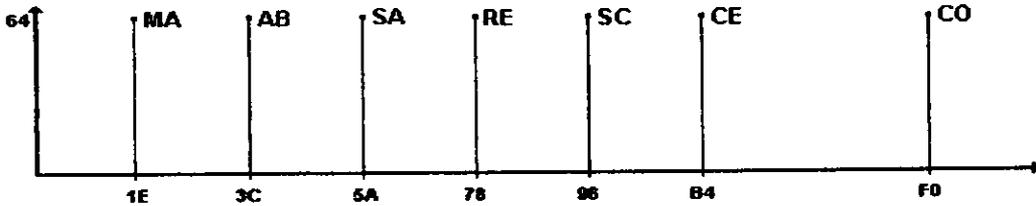
**Temperatura:**



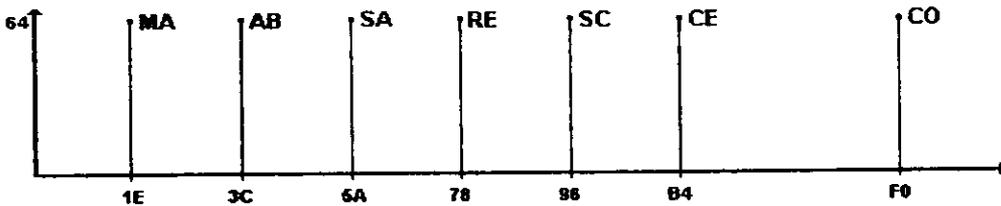
**Error\_Temperatura:**



**Angulo3:**



**Angulo4:**



**Fig. 5.2 Conjuntos Difusos**

**5.2 Algoritmo de Difusificación.**

La información mostrada en la sección anterior, forma parte de lo que se conoce como *base de conocimientos*, mediante la cual se lleva a cabo el proceso de difusificación. En este proceso como ya se mencionó, se asigna un grado de membresía a cada valor digitalizado. El grado de membresía por lo general es un valor perteneciente al intervalo cerrado  $[0,1]$ , aunque no es forzoso verlo como la única opción. Para nuestra aplicación se utilizará el intervalo  $[0,100]$ , esto por dos razones, una de ellas es que es posible utilizar cualquier intervalo, siempre y cuando presente un orden lógico que pueda ser utilizado para representar distintos grados de membresía. La otra razón es que este intervalo resulta mejor para manejarlo digitalmente, ya que el uso de cantidades en punto flotante involucra un grado de complejidad mayor en la programación.

El mecanismo que se sigue para difusificar un dato exacto, en sí, no es muy complicado, sobre todo cuando se utilizan funciones de membresía formadas por rectas. Cuando ésto sucede, simplemente se hace uso de la ecuación fundamental de la recta:  $y = mx + b$ ,

donde como se sabe,  $m$  representa a la pendiente,  $x$  la variable (la lectura exacta) , y  $b$  la distancia entre el eje de las abcisas y el punto donde la recta corta al eje de las ordenadas.

Debido a que nuestras funciones difusas son del tipo triangular, se considerará la siguiente estructura de datos para representar digitalmente a las funciones de membresía. Tomando en consideración únicamente 2 de las 3 rectas que conforman al triángulo, denominaremos *primera recta* a la que cuenta con una pendiente positiva, formada por los puntos  $p1$  y  $p2$  en la figura 5.1, mientras que la *segunda recta* será aquella que forman los puntos  $p2$  y  $p3$ , la cual a su vez presenta una pendiente negativa.

[Byte 1] = Valor inicial del dominio de la función (punto No. 1 de acuerdo a la figura 5.1).

[Byte 2] = Valor intermedio del dominio de la función (punto medio).

[Byte 3] = Valor final del dominio de la función (punto No. 3 de acuerdo a la figura 5.1).

[Byte 4] = Parte entera de la primera pendiente.

[Byte 5] = Parte fraccionaria de la primera pendiente.

[Byte 6] = Byte más significativo de la primera constante  $b$ .

[Byte 7] = Byte menos significativo de la primera constante  $b$ .

[Byte 8] = Parte entera de la segunda pendiente.

[Byte 9] = Parte fraccionaria de la segunda pendiente.

[Byte A] = Byte más significativo de la segunda constante  $b$ .

[Byte B] = Byte menos significativo de la segunda constante  $b$ .

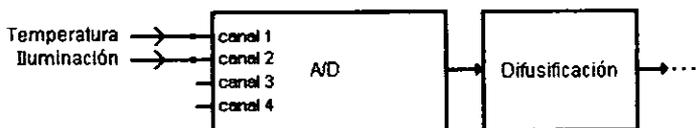


Fig. 5.3 Módulos de digitalización y fuzificación

En la figura 5.3 se muestran a bloques dos etapas del controlador difuso, entre las cuales destaca la de difusificación, estando constituida por un bloque de código de microcontrolador que procesa la información proveniente del convertidor A/D. Una de las acciones previas a la difusificación consiste precisamente en seleccionar la señal que será difusificada, esto mediante la activación de uno de los

canales del convertidor A/D. A continuación se detalla el proceso que se lleva a cabo en el bloque de difusificación, considerando la existencia de datos válidos en los registros de resultados del convertidor A/D.

El algoritmo de difusificación por lo tanto comprenderá los siguientes pasos.

1. **Lectura del dato digitalizado  $V_i$  proveniente del canal activo del CAD (convertidor A/D) y obtención del error  $E$  ( $E = \text{Set Point} - V_i$ ).**
2. **Verificar si el dato pertenece a la variable temperatura ( $LEC=T$ ) o a Iluminación ( $LEC=I$ ).**
3. **Indicar ciclo de difusificación del valor de entrada  $V_i$  mediante la carga de un \$00 en la variable  $SW$ , es decir,  $SW = \$00$ .**
4. **Si  $LEC=T$ , se consultan las bases de conocimientos ubicadas en la dirección  $BC\_TEMP$ . Esta acción se logra cargando en el registro índice  $IX$  el valor de  $BC\_TEMP$ .**
5. **Si  $LEC=I$ , se consultan las bases de conocimientos ubicadas en la dirección  $BC\_ILUM$ . Esta acción se logra cargando en el registro índice  $IX$  el valor de  $BC\_ILUM$ .**
6. **Si  $(SW \text{ AND } \$04) = \$04$  y  $LEC = T$ ,  $IX$  toma el valor de la variable  $BC\_ERTEMP$ .**
7. **Si  $(SW \text{ AND } \$04) = \$04$  y  $LEC = I$ ,  $IX$  toma el valor de la variable  $BC\_ERILUM$ .**
8. **Se carga la variable  $CONT$  con 1. Esta variable representa el conjunto difuso actual.**
9. **Se carga la variable  $CONT1$  con 0. Esta variable llevará el conteo del número de funciones difusas para las cuales el valor de entrada es parte del dominio de la función.**
10. **Comparar el valor de entrada ( $V_i$ ) con el 3er. punto ( $p_3$ ) de la función difusa actual, ubicado en la dirección  $IX+3$ .**
11. **Si  $V_i$  es mayor que  $p_3$ , entonces se pasa al punto #26.**
12. **Comparar a  $V_i$  con el 1er. punto ( $p_1$ ) de la función difusa actual, ubicado en la dirección  $IX+1$**
13. **Si  $V_i$  es menor que  $p_1$ , brinca al punto #26.**
14. **Comparar a  $V_i$  con el 2o. punto ( $p_2$ ) de la función difusa actual, el cual se ubica en la dirección  $IX+2$ .**
15. **Si  $V_i$  es igual a  $p_2$ , la variable  $U$  (variable de grado de membresía) toma el valor hexadecimal 64 (100 decimal) y brinca al punto #23.**
16. **Si  $V_i$  es mayor que  $p_2$ , brinca al punto #20.**

17. Si no es mayor, entonces checar si el dato de la dirección IX+4 (pendiente#1) es igual a cero.
18. Si la pendiente es cero, entonces  $U = \$64$  (100 decimal) y brinca al punto #23.
19.  $U = Vi * pendiente\#1 + b1$ , donde pendiente#1 es un dato almacenado en las direcciones IX+4 e IX+5, y b1 es igualmente otro dato almacenado pero en las direcciones IX+6 e IX+7. Posteriormente brincar a la línea #23.
20. Checar si el dato de la dirección IX+8 (pendiente#2) es igual a cero.
21. Si es igual a cero, entonces  $U = \$64$  (100 decimal) y brinca al punto #23.
22.  $U = Vi * pendiente\#2 + b2$ , donde pendiente#2 es un dato almacenado en las direcciones IX+8 e IX+9, y b2 es igualmente otro dato almacenado pero en las direcciones IX+A e IX+B.
23. FUNCION [CONT1] = CONT.
24. FUNCION [CONT1+4] = U.
25.  $CONT1 = CONT1 + 1$ .
26.  $CONT = CONT + 1$ .
27.  $IX = IX + \$0B$ .
28. Si  $CONT < [IX + 0]$ , pasa a la línea #10.
29. Si  $(SW \text{ AND } \$02) = \$02$ , se pasa al siguiente algoritmo (inferencia o evaluación de reglas).
30.  $SW = \$02$ .
31. Si  $Vi < SETPOINT$  entonces  $Vi = SETPOINT - Vi$ ,  $SW = (SW \text{ OR } \$04)$ .
32. Si  $Vi > SETPOINT$  entonces  $Vi = Vi - SETPOINT$ ,  $SW = (SW \text{ AND } \$04)$ .
33. Brinca a la línea #6

### 5.3 Creación de la Base de Conocimientos.

La base de conocimientos dentro de un controlador difuso, es el módulo en el que se representan las experiencias y conocimientos que se tienen acerca del proceso a controlar. Aunque existen distintas técnicas para representar artificialmente el conocimiento, las reglas son el método utilizado para implementar la base de conocimientos de los controladores difusos, debido a lo cual, a esta también se le conoce como 'base de reglas'.

Para nuestro caso, el módulo estará formado por reglas con dos antecedentes, tal y como se muestra en la siguiente expresión:

**SI <x1 es A> Y <x2 es B> ENTONCES y1 es C**

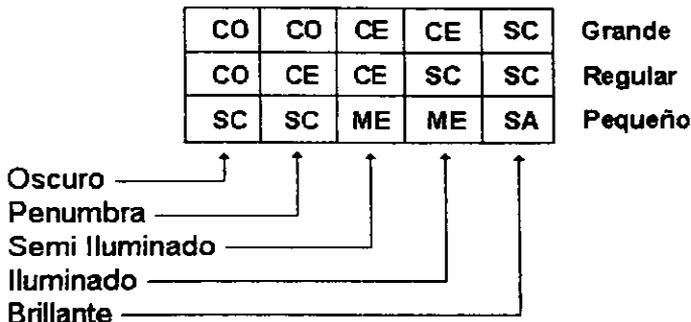
donde uno de los antecedentes será una sentencia que involucre a un valor difusificado de entrada con un conjunto difuso de la variable correspondiente, por ejemplo, <Temperatura es Caliente>, mientras que el otro antecedente será igualmente una sentencia, la cual en esta ocasión relacione a cualquiera de las variables de error, tal como Error\_Iluminación o Error\_Temperatura con sus conjuntos difusos correspondientes.

Un sistema de control completo debería tener una lista de reglas similares a esta, las cuales unidas describan la respuesta del controlador. El orden de las reglas en esta lista no afecta la respuesta del sistema. Imaginariamente, todas las reglas son evaluadas en un mismo tiempo, aunque en los sistemas basados en programación, estas son evaluadas secuencialmente.

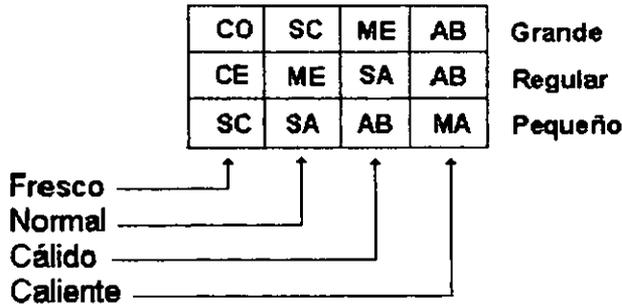
A manera de apoyo al momento de escribir las reglas, se acostumbra hacer uso de una herramienta que se conoce como 'Matriz Asociativa Difusa', la cual resulta bastante útil para sobre todo asegurar que no se omita ninguna regla en el sistema. Esta matriz se construye con los valores de las variables de entrada (conjuntos difusos). De esta forma, si se manejan reglas de 2 antecedentes y 1 salida, las columnas representarían por ejemplo al primer antecedente, mientras que los renglones estarían relacionados con el segundo, o viceversa.

A continuación se muestran las matrices correspondientes a las reglas para la iluminación y la temperatura respectivamente:

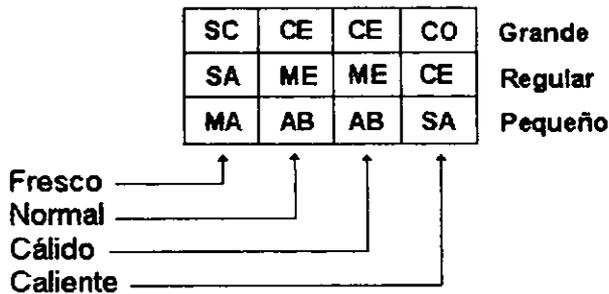
#### Base de Reglas para Iluminación



**Base de Reglas para Temperatura (calentar)**



**Base de Reglas para Temperatura (enfriar)**



Como se puede observar, los renglones de las dos matrices están formados por los valores de las variables difusas de error, mientras que las columnas se forman con los valores de las variables difusas Iluminación y Temperatura respectivamente.

En base a lo indicado por estas matrices difusas, podemos construir cualquier regla mediante la combinación de las columnas y los renglones, obteniendo así la salida de la regla por la etiqueta localizada en la intersección de los dos antecedentes. Si por ejemplo la temperatura fuera *Cálida*, y el error de temperatura fuera *Regular*, entonces el ángulo sería *Semi-Abierto*. Esto expresado en una regla común se indicaría como:

**SI** <Temperatura es Cálido> **Y** <Error es Regular> **ENTONCES** <Angulo3 es Semi\_Abierto>

Para implementar las matrices asociativas que representen al conjunto de reglas de nuestro sistema, debemos desarrollar una técnica lo más óptima posible, que nos permita acceder los datos en un tiempo breve, y así también ocupe una mínima cantidad de memoria ROM. La dimensión de las tres matrices es variable. La técnica propuesta para disparar las reglas adecuadas se basa en la utilización de dos conjuntos de vectores (direcciones de memoria). Los primeros vectores apuntan a las direcciones en donde se ubican los segundos vectores. A su vez, estos últimos vectores apuntan a las

direcciones en donde se encuentran almacenadas las conclusiones. Por lo tanto la estructura de la base de reglas se verá de la siguiente forma:

### VECTORES DE RENGLONES

Con IX = BR\_ENFRIAR :

$$\begin{aligned} [0 + IX] &= \$REN\_ENFRIAR \\ [2 + IX] &= \$REN\_ENFRIAR+3 \\ [4 + IX] &= \$REN\_ENFRIAR+6 \\ [6 + IX] &= \$REN\_ENFRIAR+9 \end{aligned}$$

Con IX = BR\_CALENTAR

$$\begin{aligned} [0 + IX] &= \$REN\_CALENTAR \\ [2 + IX] &= \$REN\_CALENTAR+3 \\ [4 + IX] &= \$REN\_CALENTAR+6 \\ [6 + IX] &= \$REN\_CALENTAR+9 \end{aligned}$$

Con IX = BR\_ILUM

$$\begin{aligned} [0 + IX] &= \$REN\_ILUM \\ [2 + IX] &= \$REN\_ILUM+3 \\ [4 + IX] &= \$REN\_ILUM+6 \\ [6 + IX] &= \$REN\_ILUM+9 \\ [8 + IX] &= \$REN\_ILUM+12 \end{aligned}$$

### VECTORES DE CONCLUSIONES

$$\begin{aligned} [\$REN\_ILUM - \$REN\_ILUM+2] &= CONILUM1\_1 - CONILUM1\_3 \\ [\$REN\_ILUM+3 - \$REN\_ILUM+5] &= CONILUM2\_1 - CONILUM2\_3 \\ [\$REN\_ILUM+6 - \$REN\_ILUM+8] &= CONILUM3\_1 - CONILUM3\_3 \\ [\$REN\_ILUM+9 - \$REN\_ILUM+11] &= CONILUM4\_1 - CONILUM4\_3 \\ [\$REN\_ILUM+12 - \$REN\_ILUM+14] &= CONILUM5\_1 - CONILUM5\_3 \end{aligned}$$

$$\begin{aligned} [\$REN\_CAL - \$REN\_CAL+2] &= CONCAL1\_1 - CONCAL1\_3 \\ [\$REN\_CAL+3 - \$REN\_CAL+5] &= CONCAL2\_1 - CONCAL2\_3 \\ [\$REN\_CAL+6 - \$REN\_CAL+8] &= CONCAL3\_1 - CONCAL3\_3 \\ [\$REN\_CAL+9 - \$REN\_CAL+11] &= CONCAL4\_1 - CONCAL4\_3 \end{aligned}$$

$$\begin{aligned} [\$REN\_ENF - \$REN\_ENF+2] &= CONENF1\_1 - CONENF1\_3 \\ [\$REN\_ENF+3 - \$REN\_ENF+5] &= CONENF2\_1 - CONENF2\_3 \\ [\$REN\_ENF +6 - \$REN\_ENF+8] &= CONENF3\_1 - CONENF3\_3 \\ [\$REN\_ENF +9 - \$REN\_ENF+11] &= CONENF4\_1 - CONENF4\_3 \end{aligned}$$

De acuerdo a como se muestra en el desarrollo del algoritmo anterior, los conjuntos difusos se representan mediante números, siguiendo el mismo orden que llevan en su representación gráfica de la figura 5.2. De tal forma que si el valor de entrada perteneciera al conjunto *Penumbra* de la variable *Iluminación*, entonces el programa reconoce al valor digitalizado como perteneciente al conjunto #2. De esta forma al evaluar la regla, el controlador sabe que debe consultar la segunda columna de la MAF (Matriz Asociativa Difusa), y para finalmente obtener la conclusión, se suma la posición del conjunto difuso correspondiente a la variable *Error*. Así, suponiendo que *n* y *m* representan a los antecedentes de la regla. De esta forma, '*n + IX*' forman una dirección que apunta a la zona de vectores de renglones pertenecientes a la columna *n*, a su vez [*n + IX*] + *m* apunta al byte que almacena a la conclusión relacionada con los 2 antecedentes.

$$[n + IX] + m \rightarrow \text{Conclusión } p$$

Con el fin de comprender un poco más esta técnica considérese una lectura de temperatura, que una vez que ha sido difusificada, se le asigna un grado de pertenencia  $\mu$  con respecto al conjunto "Cálido". Por otra parte el error de temperatura es "Pequeño". Con estos datos y de acuerdo a la matriz asociativa #3, se tendría una regla de la forma:

**SI** <Temperatura es *Cálida*> **Y** <Error\_Temp es *Pequeño*> **ENTONCES** <Angulo3 es *Abierto*>

esta conclusión se obtiene considerando que la función difusa *Cálido* es el tercer valor de la variable *Temperatura*; así accedando el dato de la dirección *4 + IX*, se obtiene un valor que servirá para formar una segunda dirección que apunte al vector *REN\_ENF+6*. En este último vector se almacenan las 3 posibles conclusiones pertenecientes al antecedente "Temperatura es *Cálido*". Ahora tomando en consideración que *Pequeño* es la primera función difusa de la variable *Error\_Temp*, se tiene que la conclusión estará dada por el tercer elemento del vector *REN\_ENF+6*, o sea, *Abierto*.

## 5.4 Algoritmo de evaluación de reglas por el método de inferencia Max-Min.

El desarrollo del método de inferencia max-min puede llevarse a cabo en los siguientes pasos:

1. Si  $LEC=T$  y  $(SW \text{ AND } \$04) = \$00$ , cargar el registro índice *IX* con la dirección de la Base de Reglas ubicada en la dirección  $\$BR\_FRIO$  (base de reglas para el enfriador). Si no entonces  $IX = \$BR\_CALIENTE$ .

2. Si  $LEC=I$ , cargar el registro índice IX con la dirección de la Base de Reglas ubicada en la dirección \$BR\_ILUM.
3. Limpiar la variable CONT, es decir,  $CONT=0$ .
4. De la dirección  $FUNCION + CONT$ , obtener el primer antecedente de la regla
5. Si el antecedente es cero, pasar a la línea #28.
6. Almacenar en la variable U1 el dato almacenado en la dirección  $FUNCION+CONT+4$ .
7. Multiplicar por 2 al primer antecedente y posteriormente restarle 2 al resultado. Este resultado queda almacenado en el registro B.
8. Cargar al registro IX con el dato que empieza en la dirección  $BR\_..... + B$ .
9. Cargar en el registro IY con la dirección formada por  $(FUNCION+2) + CONT$ , o sea el segundo antecedente.
10. Almacenar en el registro B el dato ubicado en la dirección IY y decrementarlo en 1.
11. Cargar en CONCLU el dato ubicado en  $IX + B$ .
12. Obtener el valor de membresía mínimo entre las variables Error y U1, posteriormente guardarlo en MIN.
13. Comparar MIN con el dato almacenado en la variable SALIDA(CONCLU).
14. Si  $MIN > SALIDA(CONCLU)$ , entonces  $SALIDA(CONCLU)= MIN$ .
15. Si  $CONT= 0$ , pasa a la línea #22.
16. Cargar en el registro B el segundo antecedente de la segunda regla ubicado en la dirección  $FUNCION+3$ .
17. Si el dato anterior es cero, pasar a la línea #28, si no, decrementar a B en 1.
18. Obtener la conclusión de la segunda regla ubicada en la dirección  $(FUNCION+3)+B$  y guardarla en la variable CONCLU.
19. Obtener el valor de membresía mínimo entre la segunda variable de Error y U1, posteriormente guardarlo en MIN.
20. Comparar MIN con el dato almacenado en la variable SALIDA(CONCLU).
21. Si  $MIN > SALIDA(CONCLU)$ , entonces  $SALIDA(CONCLU)= MIN$ , y pasa a la línea #28.
22. Cargar en el registro IY con la dirección formada por  $(FUNCION+1) + CONT + B$ , o sea la dirección del segundo antecedente de la 4a. regla.
23. Almacenar en el registro B el dato ubicado en la dirección IY y decrementarlo en 1.

24. Cargar en **CONCLU** el dato ubicado en **IX + B**.
25. Obtener el valor de membresía mínimo entre las variables **Error** y **U1**, posteriormente guardarlo en **MIN**.
26. Comparar **MIN** con el dato almacenado en la variable **SALIDA(CONCLU)**.
27. Si **MIN > SALIDA(CONCLU)**, entonces **SALIDA(CONCLU) = MIN**.
28. **CONT = CONT + 1**.
29. Si **CONT = 2**, pasa al siguiente bloque (desdifusificación).
30. Pasa a la línea #4

## 5.5 Algoritmo de Desdifusificación.

Uno de los métodos más utilizados para realizar una desdifusificación es el del centroide, visto con detalle en el capítulo 3. Tal y como se vió, para la obtención del valor exacto a partir de éste método, se utiliza la siguiente expresión:

$$x_o = \frac{\sum (x_i \cdot \mu_i(x_i))}{\sum \mu_i(x_i)}$$

El utilizar esta ecuación involucra la realización de una cantidad considerable de operaciones de suma, multiplicación y división, las cuales en un momento dado y sobre todo dependiendo de los requerimientos de la aplicación, pueden llegar a representar un retardo de tiempo en la respuesta del controlador. Como solución a este problema se propone la utilización de SINGLETONS.

En las primeras etapas de implementación de la base de conocimientos se contempló la construcción de los conjuntos difusos de entrada y de salida. Los conjuntos difusos que representan a las salidas fueron definidos al igual que los de entrada considerando un universo de discurso. Los SINGLETONS son igualmente una representación de las salidas, pero a diferencia de los conjuntos difusos, su dominio no está formado por un conjunto de valores, sino por un valor único, el cual estaría ubicado en el punto central del rango que ocuparía el conjunto difuso, tal y como se muestra en la siguiente figura:

A continuación, se muestra la secuencia de pasos que sigue el algoritmo de desdifusificación, considerando SINGLETONS en lugar de conjuntos difusos.

1. **CONT = 1**.
2. Cargar con 0 a las variables **SUMPRODS** y **SUMGRADS**, es decir, **SUMPRODS = SUMGRADS = 0**.

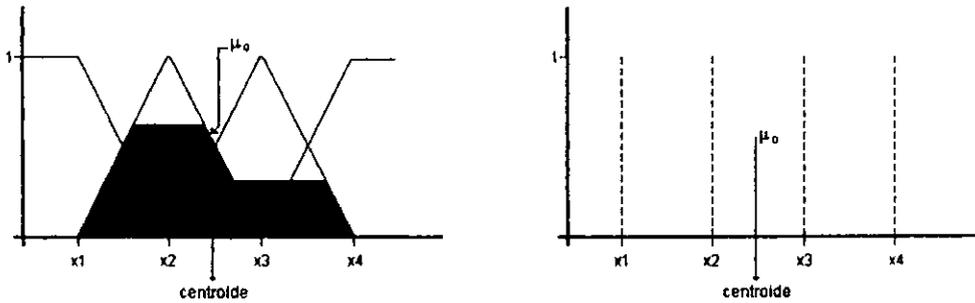


Fig. 5.4 Comparación entre conjuntos difusos y SINGLETONS

3. Verificar si el dato pertenece a la variable temperatura ( $LEC=T$ ) o a Iluminación ( $LEC=I$ ).
4. Si  $LEC=T$  y  $(SW \text{ AND } \$04)= \$04$ , entonces se consulta la base de conocimientos ubicada en la dirección BCS\_CALIENTE.
5. Si  $LEC=T$  y  $(SW \text{ AND } \$04)= \$00$ , entonces se consulta la base de conocimientos ubicada en la dirección BCS\_FRIO.
6. Checar si la diferencia entre  $V_i$  y SETPOINT es menor o igual a  $\$30$
7. Si el resultado anterior es menor o igual a  $\$30$ , entonces  $SW = (SW \text{ OR } \$80)$ .
8. Si  $LEC= I$  y  $(SW \text{ AND } \$80)= \$00$ , entonces se consulta la base de conocimientos ubicada en la dirección BCS\_ILUM.
9. Si  $LEC= I$  y  $(SW \text{ AND } \$80)= \$10$ , entonces se consulta la base de conocimientos ubicada en la dirección BCS\_ILUM1.
10. Cargar en VAR1 el dato almacenado en al dirección IX
11. Cargar el dato de la dirección IX+CONT en la variable XX.
12. Cargar en B el dato de la variable SALIDA(CONT).
13. Si  $B=0$ , entonces pasar a la línea #18.
14.  $SUMPRODS = SUMPRODS + (XX \cdot SALIDA(CONT))$
15.  $SUMGRADS = SUMGRADS + SALIDA(CONT)$
16.  $CONT = CONT + 1$

**17. Si  $CONT > VAR1$ , pasar al procedimiento #11**

**18.  $X_o = SUMPRODS / SUMGRADS$**

**19. Fin del algoritmo y se pasa al siguiente ciclo de difusificación.**

## CAPITULO 6:

### Implementación y Pruebas.

Una vez elaborado el algoritmo de control, el siguiente paso es la implementación en sí del proyecto en general. El desarrollo del controlador se llevó a cabo básicamente en 5 etapas, 4 de las cuales consistieron en el diseño y construcción de circuitos electrónicos, mientras que la última está relacionada a la programación. Concretamente las etapas a desarrollar fueron las siguientes:

1. Construcción (no diseño) y pruebas de un Sistema Mínimo en base al microcontrolador MC68HC11A1 de Motorola.
2. Diseño y construcción de un circuito detector de cruce por cero para el control de los triacs.
3. Diseño y construcción de la etapa de potencia a base de triacs.
4. Acondicionamiento de las señales provenientes de los sensores.
5. Diseño y elaboración del programa en lenguaje ensamblador para el controlador difuso.

En la siguiente figura se muestra un diagrama a bloques del sistema en su totalidad. Como se puede observar, los valores analógicos de temperatura e iluminación, son introducidos al sistema mínimo a través de las líneas analógicas 0 y 1 del microcontrolador. Una vez digitalizadas estas señales,

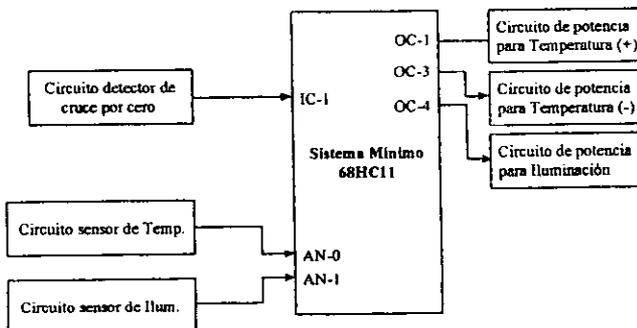


Fig. 6.1 Diagrama a bloques general del sistema.

son procesadas por el controlador difuso en sus cinco módulos los cuales son : módulo de normalización, módulo de difusificación, módulo de evaluación de reglas, módulo de desdifusificación, y módulo de desnormalización. El programa realiza continuamente un ciclo completo de procesamiento con estos cinco módulos para cada variable, arrojando como resultado una magnitud determinada de tiempo para el manejo del triac. Por otra parte, el papel que juega el circuito de detección de cruce por cero, es muy importante, ya que éste proporciona al sistema mínimo un pulso de sincronía cuando detecta que comienza un nuevo ciclo de la señal de alterna. Con este pulso de sincronía, el microcontrolador activa 2 temporizadores, los cuales combinados con el tiempo calculado por el controlador difuso, son los que proporcionan los pulsos de activación para los triacs.

Para una mejor comprensión del sistema, a continuación se describe de una forma más detallada cada una de las etapas de proyecto.

## 6.1 Construcción del Sistema Mínimo.

Cuando hablamos de un sistema mínimo, nos referimos a un número mínimo de dispositivos electrónicos, en los cuales se apoye un microprocesador o microcontrolador, para ejecutar sin restricciones un programa y llevar a cabo uno o varios procesos de control. El número y el tipo de circuitos necesarios, varían de acuerdo a los requerimientos de la aplicación, pero en términos generales, un sistema mínimo está formado como se muestra en la figura 6.2 por el  $\mu$ controlador o  $\mu$ procesador, una memoria ROM, una memoria RAM, circuito de reloj, y circuito de reset. Existirán algunas aplicaciones en las que se podrá prescindir de alguna de las 2 memorias o incluso de las 2, pero ya dependerá de un caso específico. Antes de la aparición de los  $\mu$ controladores, se incorporaban más circuitos de forma opcional, que hacían más poderoso al sistema. Algunos de estos circuitos eran los temporizadores y contadores, los puertos paralelo y serial, convertidores A/D y D/A, por citar a los más comunes. Sin embargo, en los  $\mu$ controladores actuales y concretamente en el nuestro, la mayoría de los circuitos opcionales citados, forman parte de un mismo dispositivo junto con el CPU.

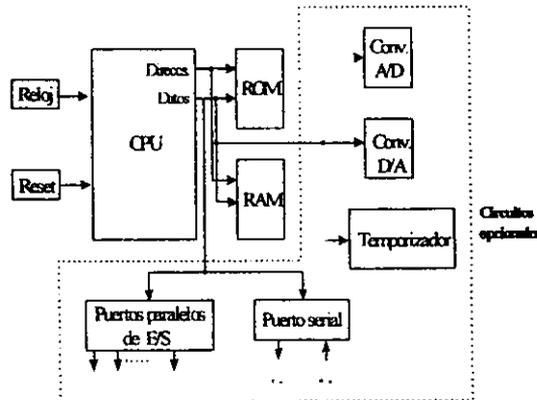


Fig. 6.2 Sistema mínimo generalizado

Para el caso de nuestro circuito, la arquitectura resulta muy parecida a la presentada en la figura. El  $\mu$ controlador 68HC11A1 cuenta como se menciona en el capítulo 4 con las siguientes características:

- 256 bytes de memoria RAM.
- 512 bytes de memoria ROM.
- Sistema de temporización de 16 bits compuesto por:
  - 3 canales de captura de entradas.
  - 5 canales de comparadores de salida.
- Interfase de comunicación serial síncrona.
- Interfase de comunicación serial asíncrona.
- Convertidor A/D de 4 canales.
- 38 terminales de propósito general de E/S.
- Sistema Watchdog (COP).

Debido a que para nuestra aplicación, la memoria interna del  $\mu$ controlador resulta muy limitada, se hace necesario incorporar en el diseño memorias externas, tanto RAM como ROM, junto con la circuitería necesaria para su habilitación, la cual en este caso se reduce a una GAL 20V8. Este dispositivo de lógica programable, contiene en su interior la circuitería de codificación necesaria para habilitar las diferentes señales con que operan las memorias, tales como las señales de lectura y escritura, así como la que se encarga de habilitar propiamente al chip. La ventaja de utilizar una GAL, es que se reduce el número de circuitos integrados necesarios para la codificación, ya que de lo contrario sería necesario incorporar en el diseño una cantidad considerable de compuertas. A esta ventaja se le suma la capacidad de cambiar las ecuaciones lógicas cuantas veces sea necesario, esto debido a que es un dispositivo reprogramable.

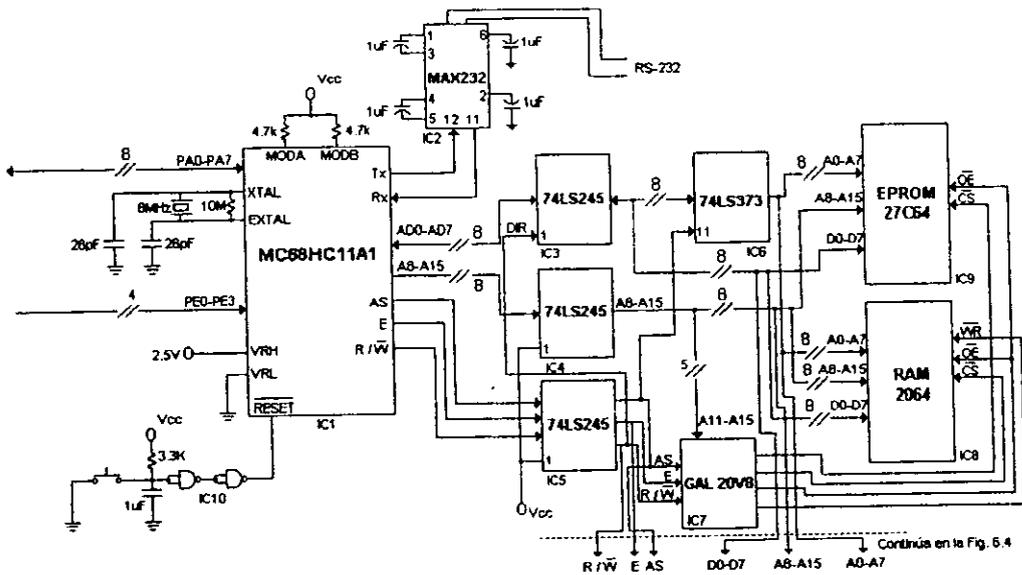


Fig. 6.3 Diagrama esquemático del sistema mínimo (parte 1)

- **Modo de operación.**

En el capítulo 4 se hace referencia a los cuatro modos en los que se puede operar al microcontrolador, estos modos están disponible mediante la combinación de los estados lógicos aplicados en los pines MODA y MODB del micro. Como se recordará, el modo mono-chip o single-chip, no trabaja con memorias externas, por lo tanto tiene los 5 puertos disponibles para operaciones de propósito general. En el caso de nuestro diseño, trabajamos en el modo expandido, tal y como se muestra en el diagrama al conectar a Vcc las dos terminales de selección.

- **Circuito de Reloj.**

En las terminales XTAL y EXTAL se encuentra conectado un circuito resonante formado por un cristal, dos capacitores de 26 pF y una resistencia de 10 M $\Omega$ . De acuerdo a la figura 6.4, la resistencia proporciona una corriente de polarización a la compuerta NAND. Esta resistencia puede tomar un valor en el rango de 1 a 20 M $\Omega$ , ya que a valores mayores, la resistencia se hace más sensible a la humedad, mientras que a valores menores, se reduce la ganancia del circuito amplificador, lo cual puede llegar a provocar una pérdida de la oscilación. Los capacitores por su parte, juegan un papel muy importante, ya que proporcionan un cambio de fase en la señal, el cual es indispensable para generar la oscilación.

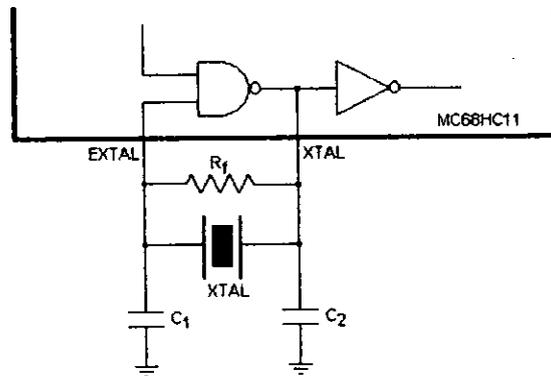


Fig. 6.4 Circuito de reloj

- **Circuito de Reset.**

Este circuito tiene como fin inicializar al microcontrolador mediante la presentación de un pulso de 0 volts en la terminal RESET. Cuando es energizado el circuito, se tiene un voltaje de 0 volts en la entrada del primer inversor, debido a que el capacitor inicialmente se encuentra descargado. Después de un tiempo  $t = RC$ , ya no circula corriente a través de éste, por lo que se tendrá un nivel de 5 volts en la entrada del primer inversor y en la salida del segundo.

Esta inicialización se conoce como reset de encendido. Así también, cuando es presionado el switch, se presenta un nivel bajo en la entrada del primer inversor, provocando un estado de reset externo.

Tanto en el reset de encendido y sobre todo en el externo, es muy grande la posibilidad de que se presenten estados transitorios, estos estados pueden provocar falsos resets. Para eliminar el problema se colocan las dos compuertas inversoras, las cuales deben ser de preferencia de tipo Schotky. Este tipo de compuertas funcionan de forma similar a como lo hace un disparador de Schmitt, de tal forma que únicamente el primer pulso que se presente será el que provoque el reset.

- **Buses de datos, direcciones y control.**

El 68HC11 maneja el concepto de bus compartido, lo cual le permite disponer de más terminales para incorporar así más funciones. Debido a esta situación se multiplexa en tiempo el puerto C, proporcionando inicialmente la parte baja del bus de direcciones, mientras que el puerto B entrega la parte alta. Después de unos cuantos ciclos de trabajo, el puerto C se convierte en bus de datos bidireccional para realizar las tareas de lectura o escritura. Existe una señal proporcionada por el mismo micro la cual avisa cuando inicia el ciclo de bus de datos en el puerto C, esta señal se conoce como AS (Address Strobe = Habilitación de dirección). Ya que el tiempo que dura presente el byte bajo de la dirección es un tiempo mucho muy corto (del orden de microsegundos), se hace necesario almacenar este byte en un registro de 8 bits, al cual se le conoce también como latch. El circuito latch que se utilizó en el diseño, es el 74LS373 cuya configuración interna se muestra en la figura 6.5, como puede verse el circuito integrado consta de 8 flip-flops del tipo D y dos líneas de habilitación marcadas como /OC y G, la primera de ellas que es habilitada en bajo, controla el estado de las salidas Q de los 8 flip-flops, mientras que la terminal G controla tanto las entradas D como a las salidas Q. Es por esto que en el circuito, el pin /OC se encuentra directamente conectado a tierra mientras que G se conecta con la terminal AS del micro.

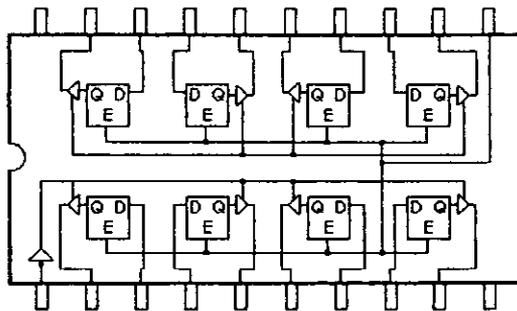


Fig. 6.5 Circuito latch 74LS373

Como es de suponerse, el 68HC11 tiene una capacidad de carga limitada, siendo posible conectar directamente a los buses tal vez una memoria y algunos periféricos. Cuando se rebasa esta capacidad, los niveles lógicos de voltaje disminuyen al grado que ya no son interpretados por los demás circuitos, de tal forma que es necesario incorporar amplificadores de señal (en corriente), para aumentar la capacidad de carga del micro. Esta etapa de amplificación está formada por una serie de circuitos buffers, los cuales proporcionan la corriente requerida. En la figura 6.3 se identifican a estos integrados con las etiquetas IC3, IC4 e IC5. Habiendo pasado por esta etapa de amplificación, las señales ya llegan reforzadas a las memorias y periféricos. En el caso del IC3, el sentido de la información esta determinado por la señal R/w proveniente del circuito IC5, el cual amplifica señales de control.

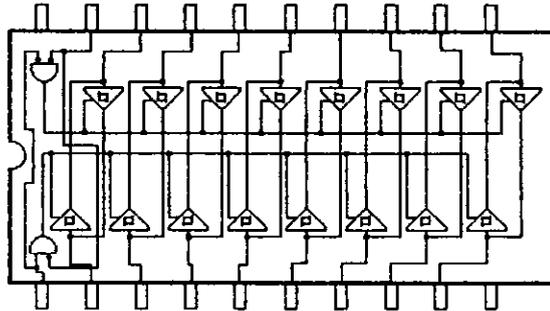


Fig. 6.6 Buffer 74LS245

Una vez formados los buses, ya pueden ser compartidos por los circuitos que los necesiten. En el caso de las memorias ROM y RAM, se puede observar que se encuentran conectadas en paralelo a los buses (direcciones, datos y control), así también los circuitos periféricos. Esto es posible ya que a pesar del número de circuitos que usen los buses, en todo momento únicamente uno estará habilitado, de esta forma pueden compartir los buses circuitos de entrada y de salida sin que se ocasione ningún corto circuito, gracias a los estados de alta impedancia con que cuentan diversos circuitos integrados.

#### • Memorias ROM y RAM.

Los circuitos marcados como IC8 e IC9, representan a las memorias RAM y ROM respectivamente en la figura 6.3. La ROM o memoria de programa, almacena el código binario que lee y ejecuta el HC11. Cada vez que inicia un ciclo de instrucción, el procesador requiere un código de operación el cual obtiene de la ROM, y para leer este dato, tienen lugar los siguientes eventos: 1) se coloca una dirección válida en el bus de direcciones de la memoria, dirección proveniente del buffer IC4 y del latch IC6, 2) por su parte el buffer IC3 se programa para lectura después de que el HC11 pone en alto su línea  $\_W/R$  y en bajo a  $AS$ . Estas dos últimas señales junto con la  $E$  y algunas líneas de dirección, se utilizan para codificar mediante la GAL20V8 una habilitación para la ROM, y 3) la memoria ROM en estas condiciones coloca en su bus de datos el byte almacenado en la dirección indicada en su bus de direcciones. Tal y como se muestra en el circuito, nuestra memoria ROM es en realidad una memoria EPROM 27C64, cuya capacidad de almacenamiento es de  $2^{13} = 8$  kb,

donde el número 13 representa la cantidad de líneas de dirección con que cuenta la memoria, es decir de A0 a A12. Por lo que respecta a la memoria RAM, se trata de una del tipo estático, con una capacidad de 8 kbytes. Su operación es parecida a la explicada para la EPROM, solo que ahora se cuenta con una línea adicional de habilitación y una para escritura. El microcontrolador utiliza a este dispositivo básicamente como memoria de trabajo, tanto para hacer consultas como para almacenar datos temporales, como pueden ser valores de variables, resultados parciales de operaciones, datos introducidos por un usuario o cualquier tipo de información que requiera ser almacenada dentro del sistema.

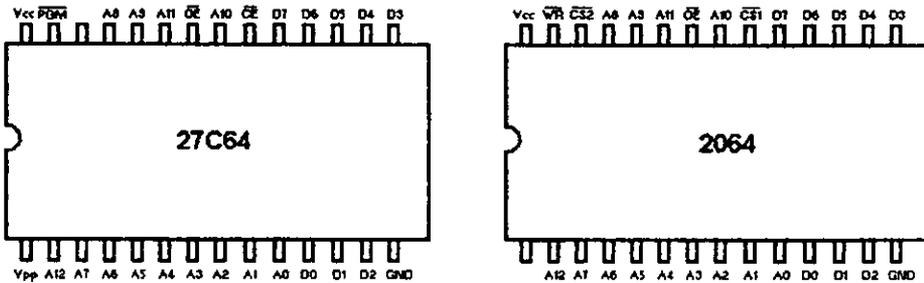


Fig. 6.7 Memorias EPROM y RAM.

- Puerto de comunicación serial.

De los 2 puertos seriales con los que cuenta el microcontrolador, se seleccionó el asíncrono, debido a que es mucho más utilizado sobre todo cuando el sistema mínimo tiene que comunicarse con dispositivos externos tales como computadoras, redes de controladores, periféricos seriales, etc. Debido a su característica *full-duplex*, el puerto cuenta con una línea de transmisión (Tx) y una de recepción (Rx), que junto con la de GND forman el circuito básico de comunicación serial asíncrona. Los niveles de voltaje con que trabajan estas señales son los mismos que maneja el micro para el resto de sus operaciones, es decir, niveles TTL.

En la práctica existen varias normas para comunicación serial asíncrona tales como la RS-422, RS-485, RS-232 y otras más, ninguna de las cuales utiliza niveles TTL. Nuestro sistema mínimo incorpora un circuito integrado fabricado por MAXIM para realizar la conversión a RS-232 cuyos niveles de voltaje son: +12v para el nivel lógico alto y -12v para el nivel lógico bajo. La ventaja que presenta este circuito es que incorpora en la misma pastilla tanto al transmisor (TTL → RS-232) como al receptor (RS-232 → TTL) sin necesidad de alimentarlo con fuente simétrica, ya que se alimenta con 5v. Lo único que se requiere es conectarle 4 capacitores en la forma indicada en la figura 6.3, de esta manera y gracias a las conversiones internas de voltaje que realiza, es capaz de suministrar en forma aproximada los voltajes del RS-232.

En la siguiente figura se presenta el diagrama básico del MAX232, en el cual se muestran los dos juegos de transceptores con los que cuenta.

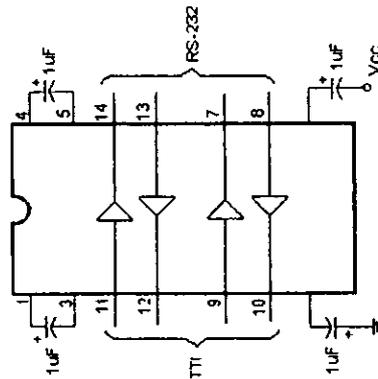


Fig. 6.8 MAX232

#### • Puerto A y periféricos.

El puerto A contempla varias funciones, ya que puede ser configurado para realizar tareas de E/S de propósito general, o para trabajar con las funciones del timer (OC = Output Compare y IC = Input Capture). En la figura 6.3 se puede observar que este puerto no se encuentra conectado a circuito alguno, aunque en realidad se ocupan 4 de sus líneas para la etapa de sincronía de la señal de alterna y para la etapa de potencia, sólo que estos circuitos auxiliares no forman parte del sistema mínimo, por lo que el puerto se encuentra disponible a través de un conector tipo header.

El diagrama mostrado en la figura 6.9 es el complemento del circuito de la figura 6.3 y se maneja con el nombre de expansión de puertos. En este circuito se puede observar nuevamente una GAL20V8, mediante la cual se habilita a cada uno de los 8 periféricos disponibles. Al igual que en el caso de la GAL anterior, la de este circuito trabaja con parte del bus de direcciones y con el de control. El listado del programa de codificación es el siguiente:

```
TITULO DECODIFICADOR DE EXPANSION DE PUERTOS PARA MODULO CON 68HC11A1
DISPOSITIVO GAL20V8
CHIP MEM_HCA1 GAL20V8
```

```
:PIN 1 2 3 4 5 6 7 8 9 10 11 12
    A15 A14 A13 A12 A11 A10 RW E AS A1 A0 GND
```

```
:PIN 13 14 15 16 17 18 19 20 21 22 23 24
    NC NC R3 R2 R1 R0 W0 W1 W2 W3 NC VCC
```

```
EQUATIONS
```

```
W0=/A15*/A14*/A13*/A12*/A11*A10*/A1*/A0*E*/RW
W1=/A15*/A14*/A13*/A12*/A11*A10*/A1*A0*E*/RW
W2=/A15*/A14*/A13*/A12*/A11*A10*A1*/A0*E*/RW
W3=/A15*/A14*/A13*/A12*/A11*A10*A1*/A0*E*/RW
/R0=/A15*/A14*/A13*/A12*/A11*A10*/A1*/A0*E*/RW
```

```
/R1=/A15*/A14*/A13*/A12*/A11*A10*/A1*A0*E*/RW
/R2=/A15*/A14*/A13*/A12*/A11*A10*A1*/A0*E*/RW
/R3=/A15*/A14*/A13*/A12*/A11*A10*A1*/A0*E*/RW
```

```
:END OF MEM_HCA1
```

Las ecuaciones lógicas del programa de la GAL, no son otra cosa que una combinación de las señales de dirección y control del HC11, con las cuales es posible implementar operaciones AND, OR, NOT, NAND, NOR, etc. y así obtener una o varias salidas. De esta forma las 4 primeras expresiones identificadas como W0 a W3, controlarán a los periféricos de escritura IOW0 - IOW3, mientras que las otras 4 harán lo propio con los periféricos de lectura IOR0 - IOR3.

En el circuito se ve claramente como cada uno de los buffers 74LS245 y los latches 74LS373 se encuentran conectados en paralelo al bus de datos del sistema, de esta forma son ya varios los circuitos que utilizan este bus, gracias a que como se dijo anteriormente, se trata de dispositivos que presentan el tercer estado (alta impedancia).

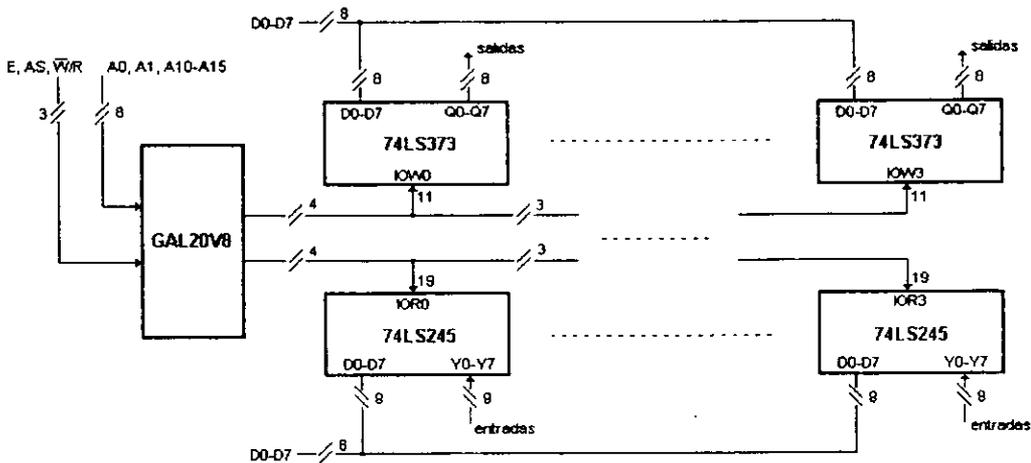
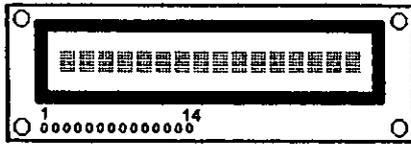


Fig. 6.9 Circuito de expansión de puertos (parte 2 del sistema mínimo).

### • Circuito de despliegue

Este circuito a pesar de que no es fundamental para el funcionamiento del controlador difuso, resultó de bastante utilidad en la etapa de implementación, y posteriormente se utilizó como interfase de visualización de los valores en tiempo real para la temperatura y la iluminación. Existen en el mercado varios dispositivos modulares de despliegue, e incluso es posible diseñar uno a base de displays de 7 segmentos, pero esto involucra más circuitería y sobre todo consumo de potencia. El dispositivo que se empleó en este proyecto, es uno bastante comercial y que presenta varias ventajas, entre las cuales podemos citar las siguientes: 1) incorpora en el mismo módulo al codificador de caracteres, 2) los caracteres son desplegados en una pantalla de cristal líquido de 16 posiciones visibles y 16 no visibles pero disponibles mediante desplazamiento de pantalla, 3) maneja código ASCII de tal manera que el procesador

ya no tiene que realizar esta codificación, 4) dispone de varios comandos para manejo de cursor y de la pantalla en general, 5) presenta un bajo consumo de potencia, y 6) es de dimensiones reducidas. La figura 6.10 presenta el diagrama básico de conexión del módulo, junto con la disposición y descripción de pines.



- |                    |         |
|--------------------|---------|
| 1.- GND            | 8.- D1  |
| 2.- Vcc            | 9.- D2  |
| 3.- Contraste (Vs) | 10.- D3 |
| 4.- RS             | 11.- D4 |
| 5.- W/R            | 12.- D5 |
| 6.- E              | 13.- D6 |
| 7.- D0             | 14.- D7 |

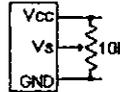


Fig. 6.10 Descripción del módulo de despliegue

El módulo se polariza mediante las líneas 1 y 2, mientras que el contraste se obtiene variando el potenciómetro. Una vez polarizado, el display toma la configuración de encendido, en la cual se encuentra deshabilitado el despliegue y el cursor, en tanto que únicamente funciona 1 línea. Respecto a éste último, el display maneja por default dos líneas, aunque nuestro modelo cuenta únicamente con una, considerando las primeras 8 posiciones como la primera línea y las 8 restantes como la segunda.

Mediante la terminal 4 (RS), se le indica al display si el byte presente en su bus de datos es un comando (RS=0), o un caracter (RS=1). La tabla 6.1 muestra una serie de comandos tanto para configuración como para el manejo en general del dispositivo. Por otra parte, aunque se trata básicamente de un periférico de escritura, es posible obtener mediante lectura ( $\_W/R = 1$ ), un byte de estatus, en el cual va insertada una bandera de aviso para el operador, la cual indica cuando se le puede introducir otro comando o caracter. La tercera línea de control es de habilitación, disponible mediante el pin 6. Una vez que ha sido habilitado el circuito (E=1), este realiza la función indicada mediante los estados de las terminales RS y  $\_W/R$ .

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	0	1	Limpia pantalla
0	0	0	0	0	0	0	0	1	X	Regresa al inicio
0	0	0	0	0	0	0	1	I/D	S	Establece modo de acuerdo con :
										I/D → Incrementa (I/D=1) o decrementa (I/D=0) el apuntador de direcciones de la memoria interna del display, provocando el movimiento del cursos.
										S → Desplaza a la derecha a la pantalla completa (cuando I/D=1 y S=1) o a la izquierda (cuando I/D=0 y S=1).

0	0	0	0	0	0	1	D	C	B	Control de Encendido/Apagado												
<p>D → El display es encendido (D=1) o apagado (D=0).                  C → El cursor es desplegado cuando C=1, y desaparece cuando C=0.                  B → Cuando B=1 el cursor se vuelve intermitente, mientras que con B=0 permanece estático.</p>																						
0	0	0	0	0	1	S/C	R/L	X	X	Desplazamiento del cursor o del display												
<p>S/C R/L</p> <table> <tr><td>0</td><td>0</td><td>Desplaza cursor a la izquierda</td></tr> <tr><td>0</td><td>1</td><td>Desplaza cursor a la derecha</td></tr> <tr><td>1</td><td>0</td><td>Desplaza display y cursor a la izquierda</td></tr> <tr><td>1</td><td>1</td><td>Desplaza display y cursor a la derecha</td></tr> </table>											0	0	Desplaza cursor a la izquierda	0	1	Desplaza cursor a la derecha	1	0	Desplaza display y cursor a la izquierda	1	1	Desplaza display y cursor a la derecha
0	0	Desplaza cursor a la izquierda																				
0	1	Desplaza cursor a la derecha																				
1	0	Desplaza display y cursor a la izquierda																				
1	1	Desplaza display y cursor a la derecha																				
0	0	0	0	1	DL	N	F	X	X	Establece función												
<p>DL → Asume bus de datos de 8 bits cuando DL=1.                  Asume bus de datos de 4 bits cuando DL=0.                  N → Activa 1 línea cuando N=0, mientras que si N=1, activa 2 líneas.                  F → Cuando F=0, los caracteres y el cursor son de 5x7 puntos, y cuando F=1, son de 5x10.</p>																						
0	0	0	1	A	A	A	A	A	A	Apunta a una dirección de la memoria reservada.												
0	0	1	An	A	A	A	A	A	A	Coloca el cursor en la posición a la que apuntan los bits D0-D5, dentro de la línea activa (An=0 activa línea 1, An=0 activa línea 2).												
0	1	BF	A	A	A	A	A	A	A	Lee el estado de la bandera BF, así como la dirección a la que apunta el apuntador de memoria.												

Tabla 6.1 Lista de comandos para el Display

Cuando se requiere desplegar un caracter, únicamente se coloca su código ASCII en el bus de datos y se proporcionan las siguientes señales de control RS=1, \_W/R=0, E=1. Conforme se despliegan los caracteres, el cursor va avanzando hacia la derecha automáticamente.

• **Análisis de señales y elaboración de pruebas para la validación del sistema mínimo.**

Una vez construido el circuito, se realizaron algunas pruebas para verificar que todas las etapas funcionaran adecuadamente. Para esto, se elaboraron algunos programas sencillos, los cuales fueron grabados en la EPROM y posteriormente ejecutados. Una vez estando corriendo el programa, se analizaron las señales que entregaba el microcontrolador, tanto en osciloscopio como en un analizador de estados lógicos. A continuación se presentan algunos programas de prueba así como los diagramas de temporización que se observaron en el analizador.

## 1.- Prueba A.

Esta prueba consistió en cargar ciclicamente el acumulador A con un valor cualquiera. El listado del programa es el siguiente

```

0000          org $e000

0000 8655 lb1  ldaa #$55
0002 20FC     bra lb1

```

la señalización que se recogió del analizador de estados lógicos, arrojó el siguiente diagrama :

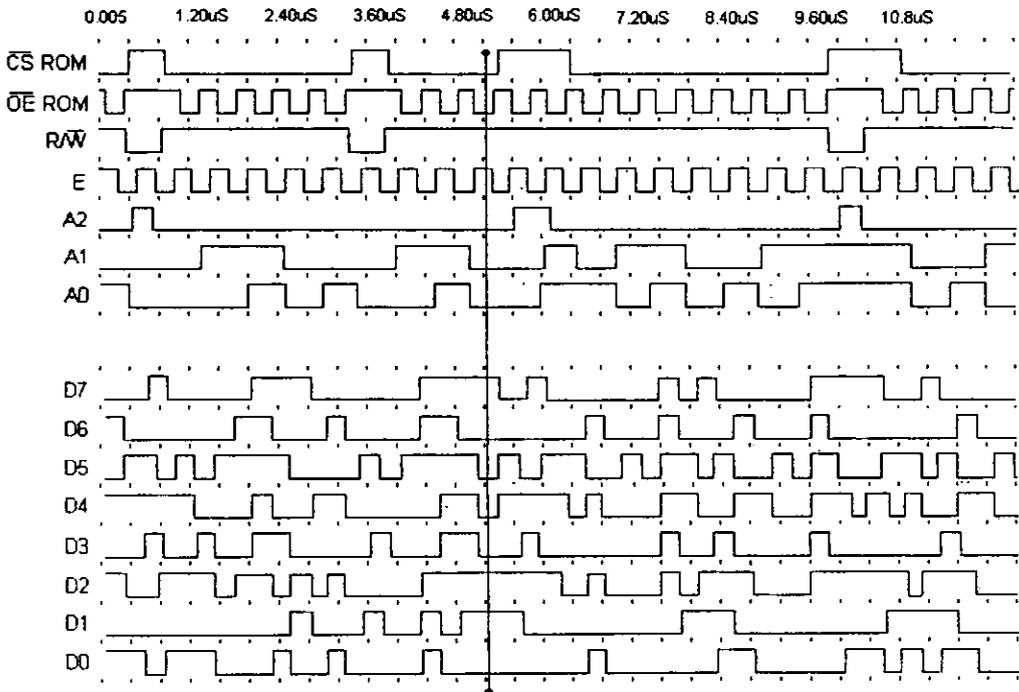


Fig. 6.11 Diagrama de temporización de la prueba No. 1

Como se puede observar, se analizaron varias señales, entre las cuales figuran 4 de control, 2 provenientes de memoria EPROM y dos directamente del microcontrolador. Así también se emplearon 3 líneas de memoria A0, A1 y A2 y el bus de datos completo. La intención al realizar esta prueba, era verificar por un lado que cada una de las señales se presentarán de acuerdo a como los manuales del fabricante lo indican en sus propias gráficas, y también realizar el seguimiento tanto de los códigos de operación del programa activo, como de las direcciones que contenían tales códigos. Para verificar esto último, se hechó mano de un archivo generado al momento de la compilación, cuya extensión es 'lst'. Este archivo contiene un listado completo del programa, junto con información adicional que comprende los códigos de operación y operandos del programa, así como las direcciones que ocuparán cada uno de ellos en la memoria.

Para hacer el seguimiento del programa en la figura 6.11, se tienen que ubicar las distintas combinaciones de señales que se generan al estar presente un dato válido en el bus de datos. La línea  $R/\overline{W}$ , estará permanentemente en alto debido a que el programa no realiza operaciones de escritura. Por lo que respecta a  $E$ , ésta indicará cuándo comienza un ciclo de acceso a la memoria. Las señales  $\overline{OE}$  y  $\overline{CE}$  como se mencionó antes, son un producto de  $E$  y  $R/\overline{W}$ , pero es importante considerarlas para poder validar el funcionamiento tanto de la GAL como de la misma EPROM. Así, para identificar en el diagrama un dato leído de la memoria, se toman los estados de las señales  $D0 - D7$  en el instante cuando  $E=1$ ,  $R/\overline{W}=1$ ,  $\overline{CE} = \overline{OE} = 0$ .

Para ejemplificar esta explicación, en el diagrama se trazó una línea vertical que corta a la totalidad de las señales, permitiendo ubicar el momento en el que inicia un ciclo del programa. Así, se puede identificar claramente la presencia del código de operación 86, que significa la carga de modo inmediato de un operando en el registro A. También se identifica la dirección \$00 en las líneas de dirección, todo esto cuando se presentan las habilitaciones ya comentadas en las líneas de control. El dato válido siguiente se presenta 3 ciclos  $E$  después, ya que durante este intervalo de tiempo la memoria se encuentra deshabilitada, al encontrarse en alto la señal  $\overline{CS}$ . De esta forma, se pueden ubicar los datos \$55 en la dirección \$0001, \$20 en la dirección \$0002, y \$FC en la \$0003, finalizando un ciclo completo de programa y dando comienzo al siguiente. Comparando esta información con los códigos en lenguaje de máquina que entrega el compilador, se puede entonces concluir que el circuito de la figura 6.3 se encuentra trabajando adecuadamente.

## 2.- Prueba B.

Esta segunda prueba estuvo dirigida a verificar el correcto funcionamiento de la expansión de puertos. Para realizar esta tarea, se escribió un programa para desplegar un carácter en la pantalla de cristal líquido, de tal forma que los resultados de la prueba pudieran verificarse directamente del periférico. No obstante se obtuvo un diagrama de temporización que se muestra en la siguiente figura, en la que aparecen las señales de control tanto de la GAL que habilita a los puertos, como de la memoria EPROM y del microcontrolador. En este caso, la señal con la etiqueta ' $\overline{OW0}$  GAL' es la que se toma de la GAL de la figura 6.9, y se trata de una señal de habilitación activa en bajo perteneciente al puerto de salida '0'. En la gráfica se puede ver como las señales de habilitación de este puerto, se presentan exactamente cuando las de la memoria EPROM están en alto, permitiendo que el display tome el dato que envía el microcontrolador.

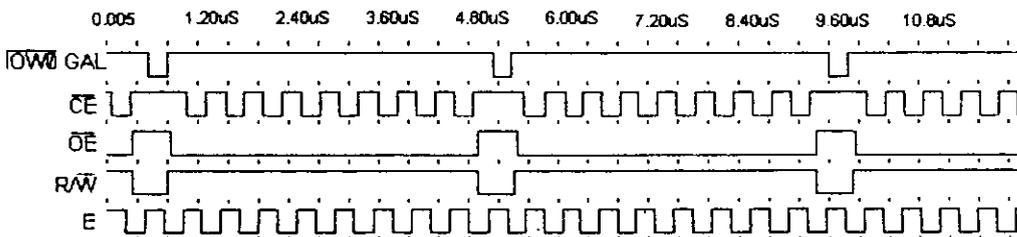


Fig. 6.12

## 6.2 Diseño de un circuito de detección de cruce por cero.

La importancia de esta etapa radica en la necesidad de contar con una señal que nos permita conocer con exactitud el momento en el que comienza un ciclo de la señal de corriente alterna, con la cual se implementarán los actuadores (calentador y lámparas o focos incandescentes). La configuración del circuito es sumamente básica tal y como se muestra en la figura 6.13.

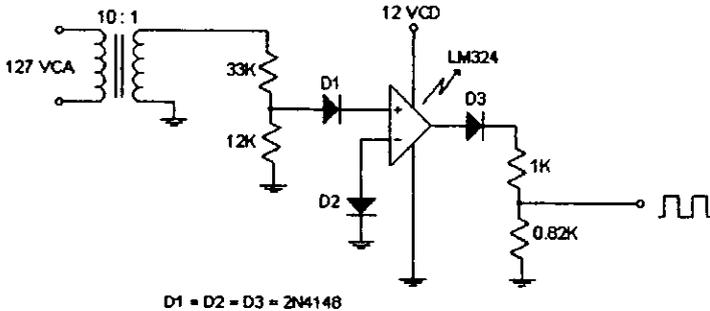


Fig. 6.13 Circuito detector de cruce por cero

Lo primero que se hace es reducir el voltaje de c.a. mediante un transformador con relación de 10:1, presentándose una magnitud de 12 v.c.a. en el secundario. El divisor de voltaje formado por las resistencias de 33k y 12k, permiten obtener una señal de aproximadamente una tercera parte de la que se tiene en el secundario, es decir de 4 volts; de esta forma es rectificadada en media onda mediante el diodo D1 e introducida a la entrada no inversora del amplificador diferencial. El amplificador entonces producirá un nivel lógico bajo cuando la magnitud de su entrada inversora sea mayor que la que se presenta en la no inversora, y por el contrario, se presentará un nivel lógico alto en el caso que la señal de mayor magnitud se tenga en la entrada no inversora. Una vez implementado el circuito se pudo observar en el osciloscopio un tren de pulsos con una frecuencia de 60 hz, perfectamente sincronizado con la señal de corriente alterna.

## 6.3 Diseño del circuito de potencia.

En el diseño de un circuito para el manejo de cargas, la magnitud de la potencia en consideración determina varias características del circuito, como son: mecanismos de protección, calibres de cables, capacidades y dimensiones de los elementos pasivos, capacidades de los circuitos de conmutación, así como las dimensiones de los disipadores de calor, circuitos de disparo, etc. En este proyecto la carga más grande no rebasa los 1000 watts, de forma que no se requiere de un diseño muy elaborado, considerando también que el tipo de carga que predomina es resistivo.

La etapa de potencia para el manejo de los actuadores consta de tres canales, formado cada uno por un triac MAC218, que tiene una capacidad de conducir hasta 8 amperes a un voltaje de 127 vca. Estos valores entonces resultan adecuados para el manejo de las cargas en consideración.

Por otra parte, existen diversas configuraciones de circuitos con triacs, diferenciándose por lo regular en base a los métodos de disparo y de protección contra transitorios. Tomando en consideración estos aspectos, se optó por una configuración sumamente elemental, que emplea un chip adicional para el disparo del triac. La ventaja de este método de disparo es que resulta adecuado para ser operado por circuitería digital, ya que únicamente se requiere un pulso de directa. Este chip de apoyo es un MAC3009, cuya configuración interna la integra un fototriac y un diodo. En la figura se muestra al circuito de potencia completo, formado como ya se mencionó, por tres canales, cada uno de los cuales se forma por un triac de potencia (MOC218), un circuito de encendido (MAC3009), y un circuito RC que sirve para proteger al triac contra transitorios.

Así también, para evitar que las terminales del microcontrolador manejen directamente a los diodos del MAC, se incorporan compuertas las cuales cumplen una función similar a la de los buffers que forman los buses del sistema mínimo.

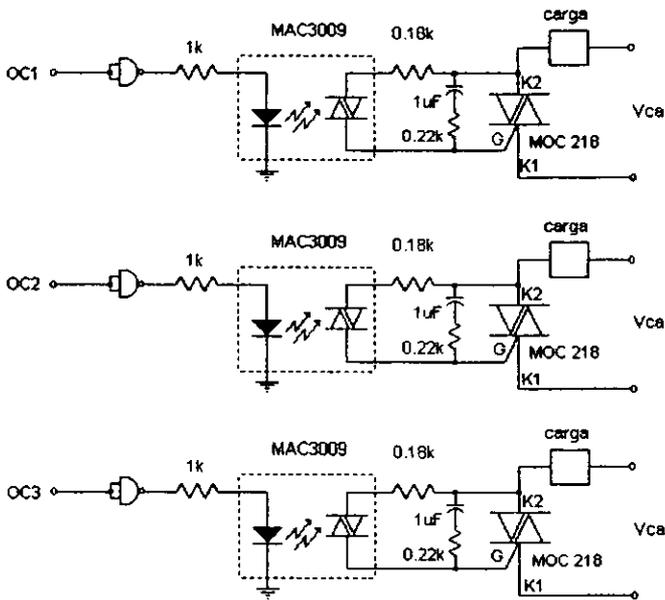


Fig. 6.14 Circuito de potencia

## 6.4 Acondicionamiento de las señales de los sensores.

Cuando se obtiene una señal directamente de un sensor, ésta por lo regular posee algunas características que hacen necesario un tratamiento o acondicionamiento previo antes de ser introducidas a la etapa de digitalización. Por lo regular se requiere amplificarlas, o atenuarlas, aplicarles una componente de corriente directa (offset), filtrarlas, rectificarlas, etc. En nuestro caso, afortunadamente nada de esto es necesario, debido a que las señales por un lado son de corriente directa y se presentan en un rango que cabe perfectamente bien en el que exige el convertidor A/D.

Como se describe en el capítulo 4, el convertidor A/D que contiene el microcontrolador, requiere una diferencia de potencial de referencia, la cual no rebase los voltajes de alimentación, pudiendo ser igual a estos o incluso menor. Esta referencia de voltaje es proporcionada por un circuito integrado regulador de voltaje (LM336), quien entrega un valor constante de 2.5 volts. Este dispositivo aparte de proporcionar el voltaje de referencia al convertidor, también sirvió para alimentar a los sensores tal y como se muestra en el siguiente diagrama.

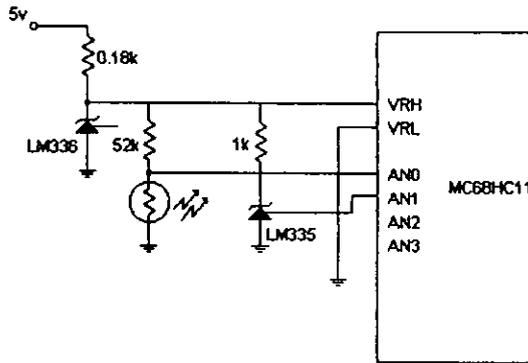


Fig. 6.15 Configuración de la etapa de sensores

## 6.5 Diseño del programa para el Controlador Difuso.

Ahora se presenta una descripción general del desarrollo del programa para el control difuso, apoyándonos totalmente en el algoritmo implementado en el capítulo 5, y en el conjunto de instrucciones de la familia HC11 para su codificación. Viendo el programa desde un punto de vista general, este se encuentra formado por siete bloques, cada uno de los cuales cumple una función bien definida. La estructura del programa es entonces como sigue:

- *Base de datos.* Cada uno de los conjuntos difusos para la difusificación y la desdifusificación, al igual que la base de conocimientos, se encuentran codificados en forma de bytes y almacenados en un segmento de la memoria EPROM. Esta área de almacenamiento se conoce como 'base de datos'.
- *Etapa de Digitalización y Normalización.* En esta parte, inicialmente son digitalizadas las señales de los sensores, para posteriormente adaptar convenientemente el dato digitalizado en el rango de \$00 a \$FF (función de normalización).
- *Etapa de Difusificación.* El dato una vez normalizado, es utilizado por este bloque para ubicarlo en el conjunto difuso que le corresponde, y así asignarle un grado de membresía.
- *Etapa de Evaluación de Reglas.* En esta etapa, se realiza un mecanismo de inferencia, basado en el método max-min comentado en el capítulo 2. De aquí se originan los conjuntos de desdifusificación recortados.

- *Etapa de Desdifusificación.* Aplicando el método del centroide a los conjuntos de la etapa anterior, se obtiene finalmente el valor de salida del controlador difuso.
- *Etapa de Desnormalización.* El dato arrojado por la etapa de desdifusificación es trasladado a otro intervalo, en el cual se ubican las unidades reales de los dispositivos a controlar.

### Variables de trabajo

Debido al número limitado de registros internos con los que cuenta el microcontrolador, y a la gran cantidad de resultados parciales y datos temporales que se generan durante la ejecución del programa, es indispensable contar con localidades de memoria libres, que permitan un intercambio de datos en cualquier momento. En seguida se presenta la lista de las variables utilizadas en el desarrollo del controlador, algunas de las cuales están dedicadas a un proceso específico y otras sirven como variables de propósito general.

Nombre	Dirección	Función
switchs	\$4000	Almacena los estados de distintos procesos del programa.
Vi	\$4001	Almacena el valor digitalizado proveniente de los sensores.
CONT	\$4002	Variable contador de propósito general.
CONT1	\$4003	Variable contador de propósito general.
X_BKUP	\$4004	Variable de respaldo para el registro IX.
GRAD_MEM	\$4006	Guarda el valor difusificado de una entrada digital.
VAR1	\$4007	Variable de propósito general.
FUNCION	\$4009	Vector que almacena los conjuntos difusos aplicados y los grados de membresía para una entrada digital dada.
VI_BKUP	\$4011	Almacena un respaldo de la señal de entrada.
RESULTADO	\$4012	Almacena el resultado que entrega el controlador difuso.
SETPPOINT	\$4013	Almacena el SETPOINT para el ciclo de control actual.
U1	\$4014	Variable de propósito general
U2	\$4015	Variable de propósito general
XX	\$4016	Variable de propósito general
SUMPRODS	\$4017	Almacena la suma de productos en la desdifusificación
SUMGRADS	\$4019	Almacena la suma de grados de membresía en la desdifusificación.
SALIDA	\$401B	Vector que almacena los grados de membresía para los conjuntos de salida.
SETPTEM	\$4028	Almacena el setpoint para el control de temperatura.
SETPILU	\$4029	Almacena el setpoint para el control de iluminación.
TCNT2	\$4040	Almacena el valor calculado para el OC2.
TCNT3	\$4042	Almacena el valor calculado para el OC3.
TCNT1	\$4044	Almacena el valor calculado para el OC1.
U3	\$4046	Variable de propósito general
swittemp	\$4047	Respaldo de la variable switch en el ciclo de temperatura.
Reg1_Tx	\$4049	Variable para rutina de comunicación serial.
NUMCIC	\$404A	Contador de ciclos de la señal de alterna para el control de la iluminación.
U4	\$404B	Variable de propósito general.

## Configuración inicial de registros del microcontrolador.

Las siguientes líneas de código inicializan algunos registros internos del micro, mediante los cuales se activan funciones de temporización, conversión analógica/digital y comunicación serial.

Idaa	#\$80	<i>Activa al convertidor analógico / digital al poner en alto el byte más alto del registro OPTION.</i>
staa	OPTION	
Idaa	#\$10	<i>Programa al Input Capture 1 para que se active por una transición de bajo a alto.</i>
staa	\$1021	
lds	#\$00ff	<i>Ubica al Stack Pointer en la dirección \$00ff</i>
Idaa	#\$00	<i>Fija la frecuencia del pre-escalador del temporizador a un valor de E/1.</i>
staa	\$1024	
Idaa	#\$30	<i>Programa al puerto asíncrono para una velocidad de 9600 baudios.</i>
staa	\$102b	
Idaa	#\$00	<i>Configura la trama para la comunicación serial: 8 bits, 1 bit de paro, 1 bit de inicio.</i>
staa	\$102c	
Idaa	#\$0c	<i>Activa el transmisor y el receptor del puerto asíncrono.</i>
staa	\$102d	
Idaa	#\$04	<i>Activa la interrupción del Input Capture 1.</i>
staa	\$1022	

## Rutina Principal

Independiente de los seis bloques descritos anteriormente, existe una rutina principal la cual distribuye los procesos que se realizan en el programa. En este segmento se atienden principalmente dos tareas: una es el control de un teclado por medio del cual se pueden variar cualquiera de los setpoints, y por otro lado también se supervisa el conteo de ciclos de corriente alterna. Concretamente, mediante esta última tarea se determina cuando debe comenzar un ciclo para control de iluminación y cuando comienza uno de temperatura. Una vez que se acumulan 30 pulsos en la variable NUMCIC, esta se limpia y se activa un ciclo de control para iluminación, activando al mismo tiempo una bandera que indique este evento (se pone en cero el bit menos significativo de la variable switch). Después de que tienen lugar un número determinado de estos ciclos, se activa uno nuevo pero esta vez para la temperatura, e igualmente se levanta una bandera (para este caso se pone en uno el bit menos significativo de la variable switch). En el Anexo 1 se muestra el listado completo del programa, identificando esta rutina principal mediante la etiqueta SHOOT.

Debido a que la implementación de los bloques principales del controlador difuso (difusificación, evaluación de reglas y desdifusificación) no significa otra cosa que codificar los algoritmos del capítulo 5, lo que a continuación se presenta es una breve descripción del desarrollo de las etapas de

normalización, desnormalización, así como las rutinas encargadas de manejar las interrupciones del temporizador.

### ***Etapa de digitalización y normalización.***

Debido a que se manejan únicamente dos variables de control, se utilizan 2 de los 4 canales con los que cuenta el convertidor A/D. De esta forma, la señal analógica de la temperatura es aplicada al canal AN0 del micro, mientras que en el AN1 se introduce la señal de iluminación proveniente de la fotoresistencia.

Para comenzar una digitalización, es necesario saber en que tipo de ciclo se está trabajando (temperatura o iluminación). Esta información se conoce verificando el bit menos significativo (bms) de la variable switch. En base a esto, el siguiente bloque de código realiza la conversión y la normalización almacenando el valor digital en la variable Vi.

#### ***conversión y normalización de temperatura***

```

cnv4  ldaa #$80
      staa ADCTL
cnv2  bita ADCTL
      beq cnv2
      ldab ADR1           En ADR1 se carga el valor de la temp. digitalizada.
      tab
      staa Vi
      ldaa #$05
      mul
      stab var1
      ldab Vi
      idx #$000a
      idiv
      xgdx
      addb var1           El registro b contiene el valor [ADR1] * 5.1

```

#### ***conversión y normalización para iluminación.***

```

cnv1  ldaa #$81
      staa ADCTL
      ldaa #$80
cnv3  bita ADCTL
      beq cnv3
      ldab #$ff
      subb ADR2
      cmpb #$ff
      bne difusi
      ldab #$fc

```

Mediante el registro ADCTL se selecciona el canal a digitalizar, de esta forma para la temperatura se carga \$80, y \$81 para la iluminación. Una vez leído el dato del registro ADR1 (registro de resultado para el canal AN0), se multiplica por 5.1 para normalizarlo, en tanto que el almacenado en ADR2 (registro de resultado para el canal AN1), es restado de la cantidad \$FF. Ya normalizado el dato, este se almacena en la variable Vi y comienza la rutina de difusificación.

### **Etapa de desnormalización.**

De acuerdo al diseño del controlador difuso, este puede entregar los resultados en la forma que se requiera, es decir, 1 byte ó más, con o sin decimales, cantidades con signo, etc. En nuestro caso, los resultados son números enteros representados en 1 byte sin signo, que pueden tener cualquier valor en el rango de 30 a 240, debido a la distribución de los singletons o conjuntos difusos de salida. Pero como algunas veces necesitaremos contar con cantidades más pequeñas,

incluyendo al cero, una vez obtenido el resultado se le resta el número 30, con lo cual se tiene un nuevo rango que va de 0 a 210.

Para el caso de la temperatura, cada semiciclo de la onda senoidal fue dividido en 18 partes, de tal forma que los incrementos o decrementos mínimos de ángulo serán de 10 grados (180/18). Pero el caso ahora, es encontrar un factor que nos permita escalar el resultado que arroja el controlador difuso, y se convierta en una cantidad que se mueva en el rango de 0 a 18. Este factor entonces se obtiene mediante el cociente de dos cantidades: el número de divisiones del ángulo (18) y el resultado máximo del controlador (210), por lo tanto,

$$\text{factor} = 18 / 210 = 0.085$$

Introduciendo este factor en la siguiente expresión, se obtiene la cantidad de decenas de grados en la que se dispararán los triacs.

$$\text{Temperatura desnormalizada} = 18 - ( 1 + [(\text{RESULTADO} - 30) * 0.085] )$$

Pero lo que de aquí surja sigue siendo un resultado parcial, ya que es necesario traducir esta cantidad en tiempo. Para esto nos basamos en el sistema de temporización del microcontrolador, el cual maneja una resolución de 0.5  $\mu$ S por cuenta. Si consideramos que hemos dividido al semiciclo en 18 partes, cada una de las cuales tiene una magnitud en tiempo de 0.4629 mS, entonces tenemos,

$$0.4629 / 0.5e-6 = 925 \text{ cuentas o ciclos de reloj}$$

lo cual significa que cada una de las divisiones del ángulo requieren de 925 (039D en hexadecimal) cuentas del sistema de temporización. Finalmente, el resultado expresado en ciclos del temporizador, es almacenado en la variable TCNT2. De esta forma, todas las operaciones mencionadas, quedan plasmadas en el siguiente bloque de código.

```

Idaa RESULTADO
suba #\$1E      ; Se le resta 30 (p \$1E) al resultado
ldab #\$55
mul            ; multiplicación por 85 (p \$55)
ldx #\$03E8
idiv          ; se realiza una división entre 1000 para
inx           ; formar la multiplicación por 0.085
xgdx
stab CONT
lqab #\$12
subb CONT
cmpb #\$00
bne Desnorm2
ldab #\$01
Desnorm2 stab CONT
ldaa #\$9d
mul           ; se realiza la multiplicación por 925
std VAR1
ldab CONT
ldaa #\$03
mul
tba
ldab #\$00
addd VAR1
xgdx
stx TCNT2    ; se almacena el resultado final en la
ldaa switchs ; variable TCNT2.

```

Para el caso de la iluminación, el proceso de desnormalización es muy similar, sólo que ahora el semiciclo se divide aún más, concretamente en 210 partes. Para conocer el número de ciclos del temporizador que forman cada división se realizan las siguientes operaciones:

un semiciclo tiene una duración de 0.008333 segundos

por lo tanto  $0.008333 / 210 = 39.68 \mu\text{S}$

que expresado en ciclos de temporizador

$$(39.68e-6) / (0.5e-6) = 79.36$$

Este factor es el utilizado para la desnormalización de la iluminación en lo que a escala amplia se refiere, ya que por otra parte, se utiliza un factor de 4 para lograr un ajuste fino, el cual es muy útil en este caso específicamente. Para determinar cual factor emplear, el sistema toma como umbral un error de 48 (30 en hexadecimal), e inmediatamente activa la octava bandera de la variable switchs, es decir, cuando el error es menor o igual a 48, el programa utiliza el factor de 4 y activa la bandera. Independientemente del factor utilizado, una vez realizado el cálculo correspondiente, el resultado es restado o adicionado al valor actual que contiene la variable TCNT3, asignando este último resultado a la misma variable. El siguiente, es el listado correspondiente a este proceso.

```

Desnorm13 ldaa RESULTADO
          suba #$1E
          ldab #$80
          bitb switchs
          beq Desnorm23
          ldab #$04
          bra Desnorm24
Desnorm23 ldab #$4F      ; Multiplicación por 79 (= $4F)
Desnorm24 mul
          std VAR1
          ldaa RESULTADO
          suba #$1E
          ldab #$80
          bitb switchs
          beq Desnorm25
          ldab #$00
          bra Desnorm26
Desnorm25 ldab #$24      ; Multiplicación por 36 (= $24)
Desnorm26 mul
          ldx #$64
          idiv
          xgdx
          addd VAR1
dedifusi21 std VAR1
          ldaa #$04
          bita switchs
          beq Desnorm4
          ldd #$4114
          subd TCNT3
          addd VAR1
          std VAR1
          ldd #$4114
          subd VAR1
          cpd #$03E8
          bmi Desnorm14
          bra Desnorm15
Desnorm14 ldd #$03E8
Desnorm15 std TCNT3
          bra Desnorm3
Desnorm4  ldd #$4114      ;***** Resta último resultado al contador TCNT3
          subd TCNT3
          subd VAR1
          std VAR1
          ldd #$4114
          subd VAR1
          cpd #$4114
          bhi Desnorm16
          bra Desnorm17
Desnorm16 ldd #$4100
Desnorm17 std TCNT3

```

**Etapa de atención a interrupciones.**

Físicamente, los actuadores son 3, 2 para el control de la temperatura y 1 para el control de la iluminación. Gracias a esta situación, se emplean únicamente 2 temporizadores y no 3. El temporizador dedicado a la temperatura, sirve para controlar tanto a los calentadores eléctricos como al extractor de aire caliente, esto dependiendo del ciclo del que se trate, es decir, "calentar" o "enfriar".

Aparte de las interrupciones de los temporizadores de salida (OC1), se cuenta adicionalmente con otra, la cual pertenece al capturador de entradas 1 (IC1). En la rutina de esta interrupción como ya se mencionó anteriormente, se reciben y manejan los pulsos de sincronía provenientes del circuito de detección de cruce por cero. Una vez que se detecta la llegada de uno de estos pulsos, la rutina toma el instante de tiempo en el que se dió el suceso, para posteriormente sumarlo a los valores contenidos en las variables TCNT2 y TCNT3. Estos últimos resultados son asignados a los registros de control de los temporizadores de salida 3 y 4, con lo cual quedan programados para operar un tiempo después, exactamente cuando se presenta el ángulo requerido. El listado de esta rutina es como sigue:

```

IC1  ldaa #$04
      staa $1023
      ldd $1010
      std TCNT1
      ldaa #$30
      oraa $1000
      staa $1000
      ldaa #$60
      oraa switches
      staa switches
      ldd TCNT1
      addd TCNT2
      std $101A
      ldd TCNT1
      addd TCNT3
      std $101C
      inc NUMCIC
      rti

```

En lo que se refiere a las rutinas de atención de los temporizadores de salida (OC), estas realizan una labor muy sencilla, la cual consiste en generar un pulso de unos cuantos microsegundos de duración, pero suficiente para disparar en cada semiciclo al triac. Básicamente ésto es lo que se realiza en cada una de las dos rutinas, salvo en la de la temperatura, ya que aquí se debe tomar la decisión de a donde dirigir el pulso, al triac del calentador o al del extractor. Pero independientemente de esta diferencia, las rutinas resultan bastante similares como se muestra a continuación:

**Rutina de atención para el OC3**

```

OC3   ldaa #$04
      bita switemp
      beq OC3_1
      ldaa #$df
      staa U2
      bra OC3_2
OC3_1 ldaa #$bf
      staa U2
OC3_2 ldaa U2
      anda $1000 ;
      staa $1000 ; Activa pulso en OC1 u OC3

OC3_4 ldaa #$50 ;{
      deca
      bne OC3_4 ;}

      ldaa #$20
      staa $1023 ; Limpia banderas de OC1 u OC3

      ldaa U2
      coma
      oraa $1000 ;{ Desactiva pulso de OC1 u OC3
      staa $1000 ;}

      ldaa #$20
      bita switchs
      beq OC3_3
      ldd $101A ;{
      addd #$4394 ; Programa temporización para próximo semi-ciclo
      std $101A ;}
      ldaa #$df
      anda switchs
      staa switchs
OC3_3 rti

```

**Rutina de atención para el OC4**

```

OC4   ldaa #$10 ;{
      staa $1023

      ldaa #$ef
      anda $1000 ; Activa pulso en OC4
      staa $1000 ;}

      ldaa #$50
OC4_2 deca

```

```
    bne OC4_2

    ldaa #$10
    oraa $1000
    staa $1000    ;}

    ldaa #$40
    bita switchs
    beq OC4_1
    ldd $101C    ;{
    addd #$4394 ; Programa temporización para próximo semi-ciclo
    std $101C    ;}
    ldaa #$bf
    anda switchs
    staa switchs
OC4_1 rti
```

## CAPITULO 7:

### *Resultados y Conclusiones*

---

En el capítulo anterior se mostraron algunas pruebas sencillas efectuadas sobre todo al sistema mínimo, esto como parte de la serie de verificaciones que se realizaron al sistema en su totalidad. En este capítulo se presentan las aplicadas al programa, así como los resultados que se obtuvieron. Inmediatamente después de este capítulo, se reportan las conclusiones finales de este trabajo de tesis.

#### 7.1 Resultados

De manera general, se realizaron tres sesiones de pruebas al programa, la primera de ellas fue dedicada a cada una de las partes medulares del controlador difuso, es decir, al módulo de difusificación, al módulo de evaluación de reglas, así como al de desdifusificación. Esto se hizo así debido a la necesidad de evaluar el desempeño de estas rutinas básicas dentro del microcontrolador, y verificar entre otras cosas la diferencia entre los resultados obtenidos con el micro y los obtenidos analíticamente con apoyo de una calculadora. Con la certeza de que el hardware funcionaba adecuadamente, la segunda y tercera sesión de pruebas involucraron al sistema completo, es decir, se establecieron una serie de set-points o puntos de ajuste, y mediante diferentes bases de datos se realizaron las primeras acciones de control de temperatura e iluminación, de las cuales se obtuvieron distintas gráficas que muestran el comportamiento del controlador bajo diferentes condiciones.

**7.1.1 Resultados de las pruebas al controlador difuso básico.** Para llevar a cabo las verificaciones de las rutinas antes citadas, se echó mano de una herramienta muy útil en el desarrollo de software a nivel de lenguaje ensamblador. Esta herramienta no es otra cosa que un simulador de microcontrolador, el cual es un programa para PC que fue desarrollado por Motorola, como un apoyo adicional para el usuario de la línea de microcontroladores 68HC11. En la pantalla de este programa, se muestra la representación del interior del dispositivo, es decir, se pueden observar a cada uno de los registros, tanto los de propósito general, como también los de configuración y el de banderas, se pueden visualizar igualmente distintos bloques de memoria tanto de RAM como de ROM, lo cual es muy útil ya que se puede así dar seguimiento a los resultados parciales y totales durante la ejecución del programa. En otro segmento de la pantalla se puede observar el listado completo del código en lenguaje ensamblador tal cual fue creado en el editor. El simulador cuenta

por otra parte, con un conjunto de comandos propios que facilitan su uso y que ayudan en el análisis del programa. La siguiente figura muestra la pantalla del simulador, en la cual se puede observar directamente la representación de los componentes más importantes del micro. Posterior a esta figura se muestran las pruebas correspondientes a la primera sesión.

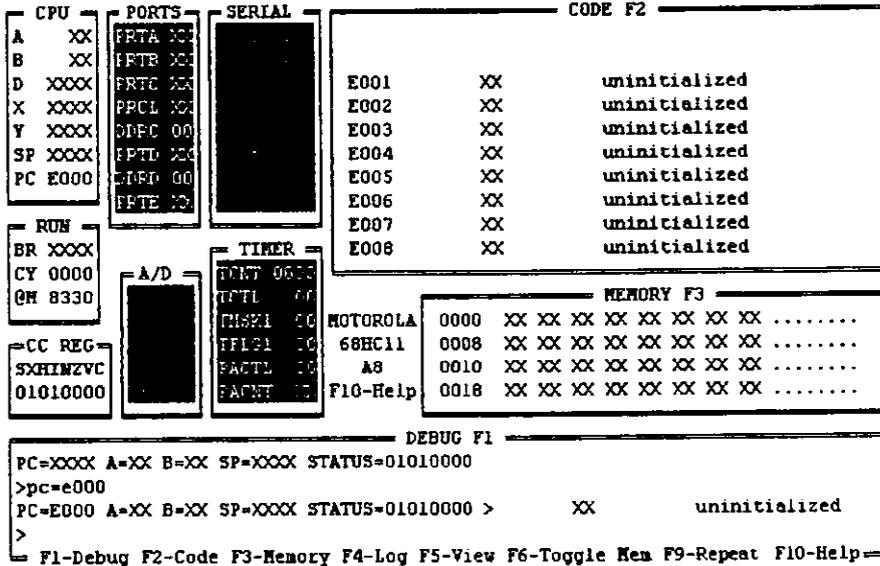


Fig. 7.1 Pantalla del simulador del 68HC11

**Ciclo Difuso para iluminación.** Para esta parte se propusieron 2 set-points diferentes y 2 valores de entrada tal cual se muestra.

Set-Point 1 = \$C0                      Vi1 = \$18  
 Set-Point 2 = \$80                      Vi2 = \$55

**Prueba 1.** Set-Point = \$C0 y Vi = \$C8

• Difusificación:

**Resultados HC11      Resultados Analíticos**

μ Iluminado	\$1E	\$1E
μ Brillante	\$46	\$46
μ Pequeño	\$64	\$64

Si  $y = mx + b$ , con  $m = -2$  y  $b = 430$ , entonces

$$\begin{aligned}\mu_{\text{Iluminado}} &= (m)(V_i) + b \\ \mu_{\text{Iluminado}} &= (-2)(200) + 430 \\ &= 30 = \$1E\end{aligned}$$

con  $m = 2$  y  $b = -330$ , entonces

$$\begin{aligned}\mu_{\text{Brillante}} &= (m)(V_i) + b \\ \mu_{\text{Brillante}} &= (2)(200) - 330 \\ &= 70 = \$46\end{aligned}$$

- Evaluación de Reglas:**

**Resultados HC11:**

$$\begin{aligned}\text{CUT}_{\text{Semi-Abierto}}(\$1E) \\ \text{CUT}_{\text{Regular}}(\$46)\end{aligned}$$

donde CUT se refiere a un conjunto de desdifusificación recortado a la altura de la cantidad contenida en los paréntesis.

**Resultados Analíticos:**

Regla No. 1 Si  $V_i = \text{Iluminado}$  y  $E = \text{Pequeño}$ , entonces Angulo = Semi-Abierto

$$\min(\mu_{\text{Iluminado}}, \mu_{\text{Pequeño}}) = \min(\$1E, \$64) = \$1E$$

por lo tanto,  $\text{CUT}_{\text{Semi-Abierto}}(\$1E)$

Regla No. 2 Si  $V_i = \text{Brillante}$  y  $E = \text{Pequeño}$ , entonces Angulo = Regular

$$\min(\mu_{\text{Brillante}}, \mu_{\text{Pequeño}}) = \min(\$46, \$64) = \$46$$

por lo tanto,  $\text{CUT}_{\text{Regular}}(\$46)$

- Desdifusificación:**

**Resultados HC11:**

$$\text{Centroide} = \$2B$$

**Resultados Analíticos:**

De acuerdo con la fórmula del centroide,

$$\text{Centroide} = \frac{\sum (\mu \cdot x)}{\sum \mu} = \frac{[(\$1E \cdot \$28) + (\$46 \cdot \$2D)]}{\$1E + \$46}$$

$$= \frac{\$4B0 + \$C4E}{\$64} = \frac{\$10FE}{\$64} = \$2B$$

por lo tanto, Centroide = \$2B

Prueba 2. Set-Point = \$B0 y Vi = \$55

• Difusificación:

Resultados HC11 Resultados Analíticos

$\mu$ Penumbra	\$3C	\$3C
$\mu$ Semi-iluminado	\$28	\$28
$\mu$ Pequeño	\$2B	\$2A
$\mu$ Regular	\$39	\$39

Si  $y = mx + b$ , con  $m = -2$  y  $b = 430$ , entonces

$$\begin{aligned} \mu_{\text{Penumbra}} &= (m)(V_i) + b \\ \mu_{\text{Penumbra}} &= (-2)(85) + 230 \\ &= 60 = \$3C \end{aligned}$$

con  $m = 2$  y  $b = -330$ , entonces

$$\begin{aligned} \mu_{\text{Semi-iluminado}} &= (m)(V_i) + b \\ \mu_{\text{Semi-iluminado}} &= (2)(85) - 130 \\ &= 40 = \$28 \end{aligned}$$

con  $m = -1.25$  y  $b = 156$ , entonces

$$\begin{aligned} \mu_{\text{Pequeño}} &= (m)(V_i) + b \\ \mu_{\text{Pequeño}} &= (-1.25)(91) + 156 \\ &= 42.25 = \$2A \end{aligned}$$

con  $m = 1.25$  y  $b = -56$ , entonces

$$\begin{aligned} \mu_{\text{Regular}} &= (m)(V_i) + b \\ \mu_{\text{Regular}} &= (1.25)(91) - 56 \\ &= 57.75 = \$39 \end{aligned}$$

• Evaluación de Reglas:

Resultados HC11:

CUT<sub>Abierto</sub>(\$2B)

$CUT_{Semi-Abierto}(\$39)$

donde CUT se refiere a un conjunto de desdifusificación recortado a la altura de la cantidad contenida en los paréntesis.

**Resultados Analíticos:**

Regla No. 1 Si Vi = Penumbra y E = Pequeño, entonces Angulo = Abierto

$$\min(\mu_{Penumbra}, \mu_{Pequeño}) = \min(\$3C, \$2B) = \$2B$$

Regla No. 2 Si Vi = Penumbra y E = Regular, entonces Angulo = Semi-Abierto

$$\min(\mu_{Penumbra}, \mu_{Regular}) = \min(\$3C, \$39) = \$39$$

Regla No. 3 Si Vi = Semi-Iluminado y E = Pequeño, entonces Angulo = Semi-Abierto

$$\min(\mu_{Semi-Iluminado}, \mu_{Pequeño}) = \min(\$28, \$2B) = \$28$$

Regla No. 4 Si Vi = Semi-Iluminado y E = Regular, entonces Angulo = Semi-Abierto

$$\min(\mu_{Semi-Iluminado}, \mu_{Regular}) = \min(\$28, \$39) = \$28$$

$$\max(\$28, \$28, \$39) = \$39$$

Por lo tanto  $CUT_{Semi-Abierto}(\$39)$   
 $CUT_{Abierto}(\$2B)$

• **Desdifusificación:**

**Resultados HC11:**

Centroide = \$4D

**Resultados Analíticos:**

De acuerdo con la fórmula del centroide,

$$\text{Centroide} = \frac{\sum (\mu \cdot x)}{\sum \mu} = \frac{[(\$2B \cdot \$3C) + (\$39 \cdot \$5A)]}{\$2B + \$39} =$$

$$= \frac{\$A14 + \$140A}{\$64} = \frac{\$1E1E}{\$64} = \$4D$$

por lo tanto, Centroide = **\$4D**

**Ciclo Difuso para Temperatura.** Nuevamente con 2 set-points y 2 valores de entrada se tiene lo siguiente:

Set-Point 1 = \$E5  
Set-Point 2 = \$70

Vi1 = \$33  
Vi2 = \$92

**Prueba 1.** Set-Point = \$E5 y Vi = \$33

• **Difusificación:**

#### Resultados HC11 Resultados Analíticos

$\mu_{\text{Fresco}}$	\$59	\$58
$\mu_{\text{Normal}}$	\$0B	\$0B
$\mu_{\text{Regular}}$	\$22	\$21
$\mu_{\text{Grande}}$	\$42	\$42

Si  $y = mx + b$ , con  $m = -1.64$  y  $b = 172$ , entonces

$$\begin{aligned}\mu_{\text{Fresco}} &= (m)(Vi) + b \\ \mu_{\text{Fresco}} &= (-1.64)(51) + 172 \\ &= 88.36 = \$58\end{aligned}$$

con  $m = 1.64$  y  $b = 72$ , entonces

$$\begin{aligned}\mu_{\text{Normal}} &= (m)(Vi) + b \\ \mu_{\text{Normal}} &= (1.64)(51) - 72 \\ &= 11.64 = \$0B\end{aligned}$$

con  $m = -1.25$  y  $b = 256$ , entonces

$$\begin{aligned}\mu_{\text{Regular}} &= (m)(Vi) + b \\ \mu_{\text{Regular}} &= (-1.25)(178) + 256 \\ &= 33.5 = \$21\end{aligned}$$

con  $m = 1.25$  y  $b = -156$ , entonces

$$\begin{aligned}\mu_{\text{Grande}} &= (m)(Vi) + b \\ \mu_{\text{Grande}} &= (1.25)(178) - 156 \\ &= 66.5 = \$42\end{aligned}$$

- Evaluación de Reglas:

**Resultados HC11:**

CUT<sub>Medio</sub>(\$0B)  
 CUT<sub>Semi-Cerrado</sub>(\$0B)  
 CUT<sub>Cerrado</sub>(\$22)  
 CUT<sub>Completo</sub>(\$42)

donde CUT se refiere a un conjunto de desdifusificación recortado a la altura de la cantidad contenida en los paréntesis.

**Resultados Analíticos:**

Regla No. 1 Si Vi =Fresco y E = Regular, entonces Angulo = Cerrado

$$\min(\mu_{\text{Fresco}}, \mu_{\text{Regular}}) = \min(\$59, \$22) = \$22$$

por lo tanto, CUT<sub>Cerrado</sub>(\$22)

Regla No. 2 Si Vi =Fresco y E = Grande, entonces Angulo =Completo

$$\min(\mu_{\text{Fresco}}, \mu_{\text{Grande}}) = \min(\$59, \$42) = \$42$$

por lo tanto, CUT<sub>Completo</sub>(\$42)

Regla No. 3 Si Vi =Normal y E = Regular, entonces Angulo =Medio

$$\min(\mu_{\text{Normal}}, \mu_{\text{Regular}}) = \min(\$0B, \$22) = \$0B$$

por lo tanto, CUT<sub>Medio</sub>(\$0B)

Regla No. 4 Si Vi =Normal y E = Grande, entonces Angulo = Semi-Cerrado

$$\min(\mu_{\text{Normal}}, \mu_{\text{Grande}}) = \min(\$0B, \$42) = \$0B$$

por lo tanto, CUT<sub>Semi-Cerrado</sub>(\$0B)

- Desdifusificación:

**Resultados HC11:**

Centroide = \$CC

**Resultados Analíticos:**

De acuerdo con la fórmula del centroide,

$$\begin{aligned} \text{Centroide} &= \frac{\sum (\mu \cdot x)}{\sum \mu} = \frac{[(\$0B \cdot \$78) + (\$0B \cdot \$96) + (\$22 \cdot \$B4) + (\$42 \cdot \$F0)]}{\$0B + \$0B + \$22 + \$42} \\ &= \frac{\$528 + \$672 + \$17E8 + \$3DE0}{\$7A} = \frac{\$615A}{\$7A} = \$CC \end{aligned}$$

por lo tanto, Centroide = \$CC

**Prueba 2.** Set-Point = \$70 y  $V_i = \$92$

- Difusificación:**

**Resultados HC11    Resultados Analíticos**

$\mu$ Normal	\$21	\$20
$\mu$ Cálido	\$45	\$45
$\mu$ Pequeño	\$64	\$64

Si  $y = mx + b$ , con  $m = -1.64$  y  $b = 272$ , entonces

$$\begin{aligned} \mu_{\text{Normal}} &= (m)(V_i) + b \\ \mu_{\text{Normal}} &= (-1.64)(146) + 272 \\ &= 32.56 = \$20 \end{aligned}$$

con  $m = 1.64$  y  $b = -170$ , entonces

$$\begin{aligned} \mu_{\text{Cálido}} &= (m)(V_i) + b \\ \mu_{\text{Cálido}} &= (1.64)(146) - 170 \\ &= 69.44 = \$45 \end{aligned}$$

- Evaluación de Reglas:**

**Resultados HC11:**

CUT<sub>Semi-Cerrado</sub>(\$45)  
CUT<sub>Cerrado</sub>(\$21)

donde CUT se refiere a un conjunto de desfusificación recortado a la altura de la cantidad contenida en los paréntesis.

### Resultados Analíticos:

Regla No. 1 Si Vi =Normal y E = Pequeño, entonces Angulo = Semi-Abierto

$$\min(\mu_{\text{Normal}}, \mu_{\text{Pequeño}}) = \min(\$21, \$64) = \$21$$

por lo tanto,  $CUT_{\text{Semi-Abierto}}(\$21)$

Regla No. 2 Si Vi =Cálido y E = Pequeño, entonces Angulo = Abierto

$$\min(\mu_{\text{Cálido}}, \mu_{\text{Pequeño}}) = \min(\$45, \$64) = \$45$$

por lo tanto,  $CUT_{\text{Abierto}}(\$45)$

### • Desdifusificación:

#### Resultados HC11:

Centroide = \$9F

#### Resultados Analíticos:

De acuerdo con la fórmula del centroide,

$$\begin{aligned} \text{Centroide} &= \frac{\sum (\mu \cdot x)}{\sum \mu} = \frac{[(\$45 \cdot \$96) + (\$21 \cdot \$B4)]}{\$45 + \$21} = \\ &= \frac{\$286E + \$1734}{\$66} = \frac{\$3FA2}{\$66} = \$9F \end{aligned}$$

por lo tanto, Centroide = \$9F

**7.1.2 Resultados de las pruebas físicas hechas al controlador a nivel de base de datos.** En el capítulo 3 se proporciona información útil para el diseño de un controlador difuso, se presentan por ejemplo las distintas opciones que se tienen para seleccionar la forma de los conjuntos difusos de entrada y de salida, así como algunos aspectos adicionales referentes a estos, también se explican 2 métodos diferentes para realizar el proceso de inferencia (evaluación de reglas), culminando con 6 métodos opcionales para realizar la tarea de desfusificación. Teniendo esta variedad de recursos, caven las preguntas: ¿ qué diferencia hay en utilizar una u otra opción ? ,

¿ existe una combinación óptima para realizar un mejor control ? Sería magnífico poder contestar estas preguntas en base a resultados obtenidos en forma práctica, pero debido a las limitantes del proyecto, nuestro controlador utiliza un solo tipo de conjuntos difusos y trabaja métodos fijos de inferencia y desdifusificación, motivo por el cual sólo será posible realizar algunas pruebas que nos muestren la influencia del número de conjuntos difusos, el efecto de distintas combinaciones de reglas, y cómo varían los resultados del controlador cuando se modifican tanto el número como la posición de los conjuntos de salida.

• **Pruebas sobre los conjuntos difusos.**

Las siguientes gráficas, tienen como objetivo mostrar el comportamiento de las variables de control modificando el número de conjuntos difusos de entrada. Junto con la gráfica, se presenta la matriz difusa que la genera, en la que sus renglones y columnas representan los distintos valores lingüísticos de tales variables. Las tres primeras gráficas, fueron generadas variando únicamente la cantidad de conjuntos difusos o valores de la variable que forma las columnas, manteniendo constante el número de renglones (ó conjuntos difusos de la otra variable), mientras que las siguientes tres presentan el caso contrario.

MA	MA	AB	SA	MA
MA	MA	AB	SA	MA
MA	MA	AB	SA	MA

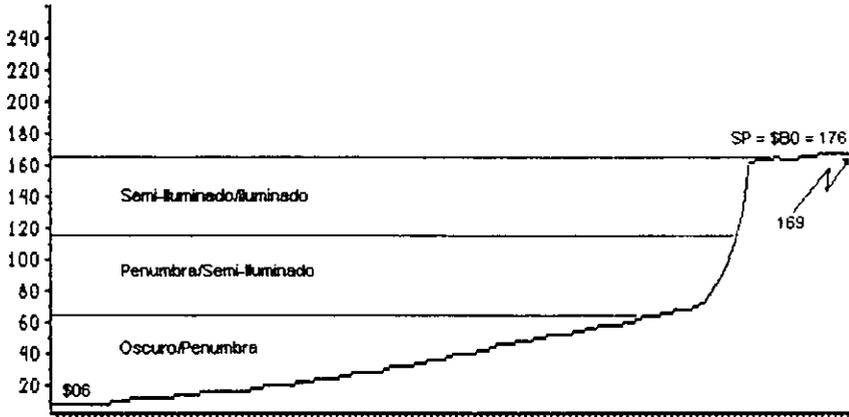


Fig. 7.2

MA	MA	AB	SA	MA	MA	MA
MA	MA	AB	SA	MA	MA	MA
MA	MA	AB	SA	MA	MA	MA

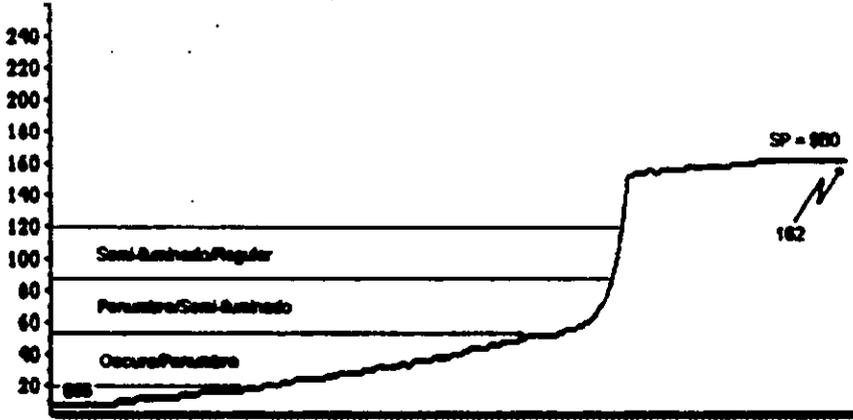


Fig. 7.3

MA	MA	AB	SA	MA	MA	MA	MA	MA
MA	MA	AB	SA	MA	MA	MA	MA	MA
MA	MA	AB	SA	MA	MA	MA	MA	MA

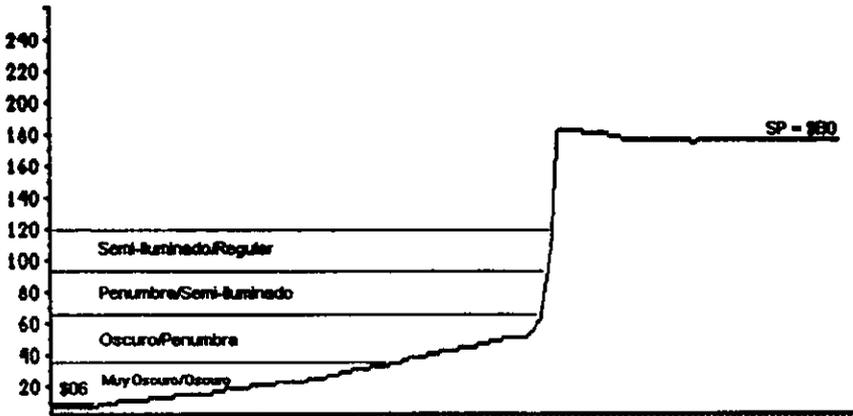


Fig. 7.4

MA MA MA MA MA  
 AB AB AB AB AB  
 SA SA SA SA SA

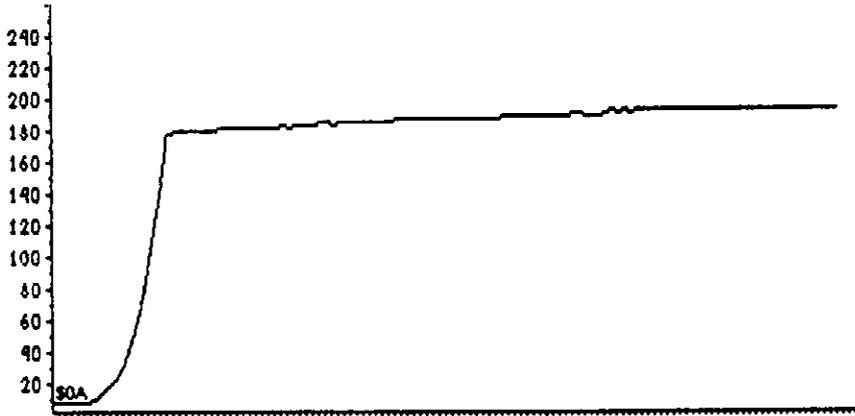


Fig. 7.5

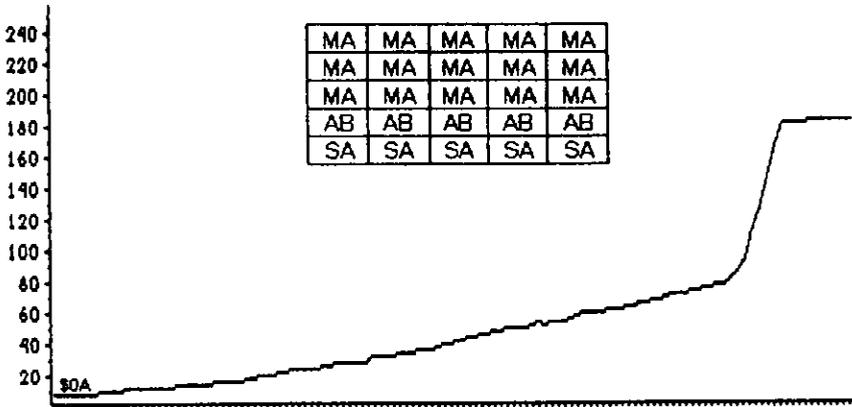


Fig. 7.6

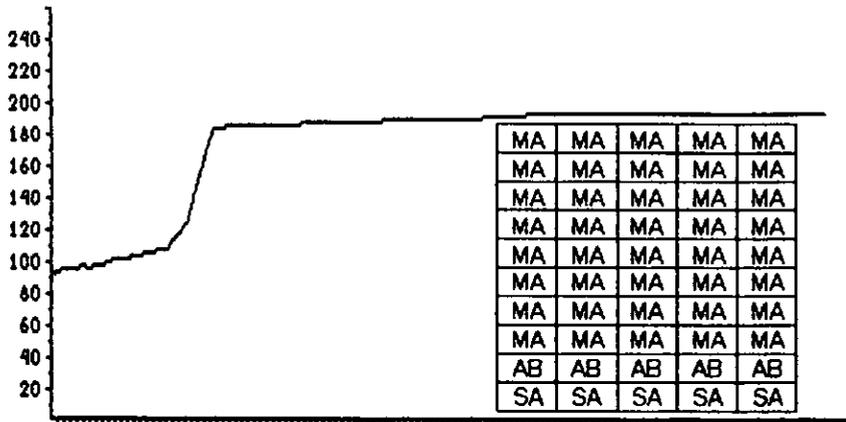


Fig. 7.7

Al observar cada de las respuestas anteriores, no podemos decir que el efecto que presentan será similar para cualquier caso. En las tres primeras pareciera como si la gráfica se desplazara a la izquierda, alcanzando por lo tanto el set-point en un intervalo de tiempo más corto, mientras que en las tres últimas, sucede todo lo contrario. El punto es que ésta no es una consecuencia natural del aumentar el número de conjuntos difusos, más bien este efecto varía dependiendo del grado de acción de las reglas que actúan. Lo que si podemos afirmar, es que el aumentar el número de conjuntos difusos nos permite tener un mejor control en intervalos más reducidos de la variable que se maneja, pudiéndose entonces obtener respuestas con intensidades más remarcadas.

A manera de complemento se presentan las dos siguientes gráficas, las cuales nuevamente muestran como varía la respuesta al disminuir el intervalo de definición de los conjuntos.

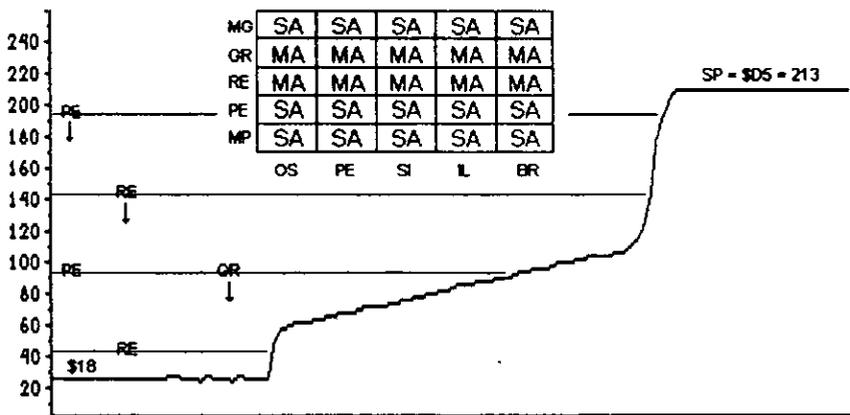


Fig. 7.8

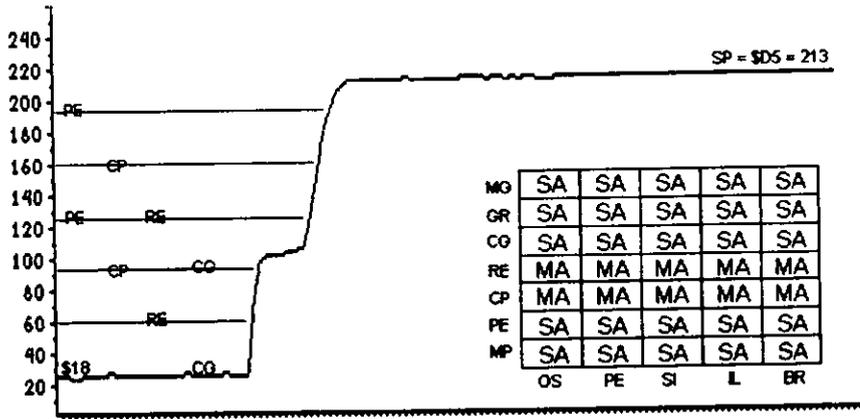


Fig. 7.9

• **Pruebas sobre la base de reglas.**

Para esta parte, se generaron algunas gráficas que muestran la participación que tienen las reglas en el desempeño del sistema. Como se podrá ver, mediante una adecuada combinación de ellas, es posible obtener diferentes características en estas gráficas, lo cual obviamente se traduce en una mayor diversidad de tipos de respuestas.

La figura 7.9 muestra una gráfica que no presenta ninguna transición brusca, el setpoint es alcanzado a través de ligeros incrementos, producto de la acción más débil que se puede lograr con el controlador. Como se puede observar en la FAM correspondiente, cada una de las reglas presentan la misma conclusión, es decir "MA", la cual de acuerdo con lo mencionado en el capítulo 5, es la que en menor grado contribuye en la intensidad de la respuesta.

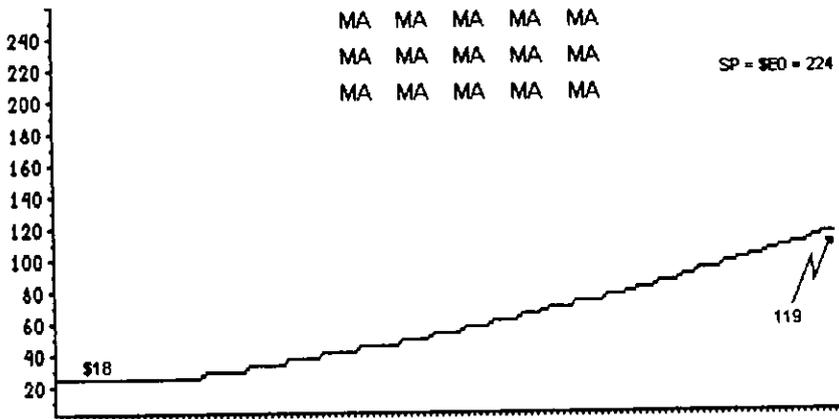


Fig. 7.10

La gráfica siguiente, es obtenida nuevamente a partir de un conjunto de reglas que presentan la misma conclusión, en este caso, "AB", la cual por así decirlo, es la que cuantitativamente continua después de "MA". Ya en esta gráfica se observa una respuesta mucho más rápida, la parte lineal presenta una pendiente visiblemente superior, de tal forma que el set-point es alcanzado casi en forma inmediata.

AB AB AB AB AB  
 AB AB AB AB AB  
 AB AB AB AB AB

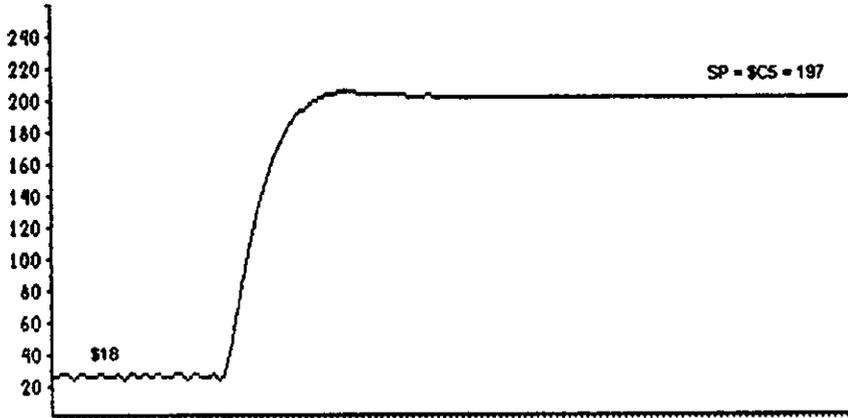


Fig. 7.11

La contribución de una regla puede ser de distinta intensidad dependiendo del lugar que ocupe en la FAM. En el caso de la figura 7.12, se observa un pequeño escalón producido por la regla :

**SI ILUM = "Penumbra" Y ERROR = "Pequeño" ENTONCES ANGULO = "Semi-cerrado"**

este efecto se pudiera haber visto más pronunciado si la regla hubiera sido,

**SI ILUM = "Oscuro" Y ERROR = "Pequeño" ENTONCES ANGULO = "Semi-cerrado"**

MA MA MA MA MA  
 MA MA MA MA MA  
 MA SC MA MA MA

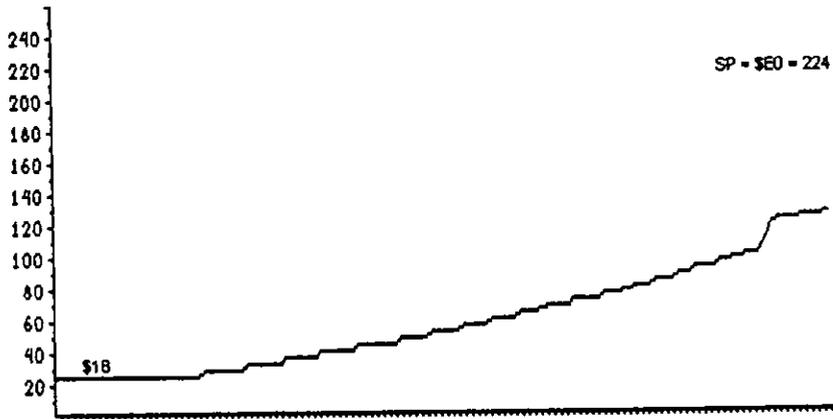


Fig. 7.12

Las figuras 7.13 y 7.14, muestran el efecto que se obtiene al intensificar las reglas correspondientes a todo un nivel de iluminación. En el primer caso, el escalón producido marca en forma bastante aproximada el rango que abarca el nivel "Penumbra", mientras que en la gráfica 7.14, el escalón se ubica el rango de "Iluminado".

MA	SC	MA	MA	MA
MA	SC	MA	MA	MA
MA	SC	MA	MA	MA

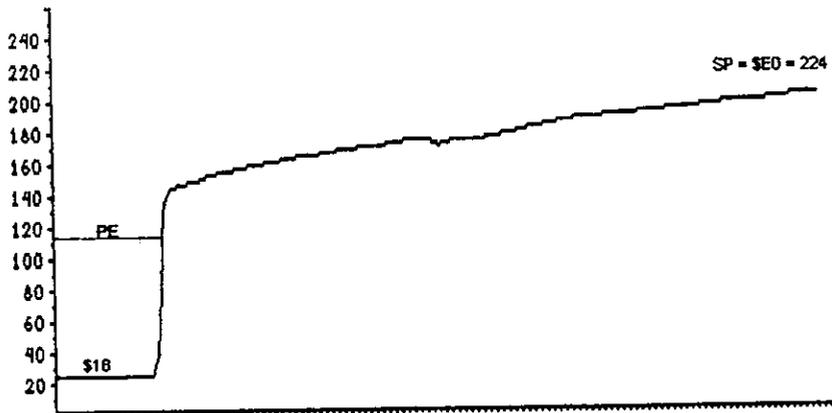


Fig. 7.13

MA	MA	MA	SC	MA
MA	MA	MA	SC	MA
MA	MA	MA	SC	MA

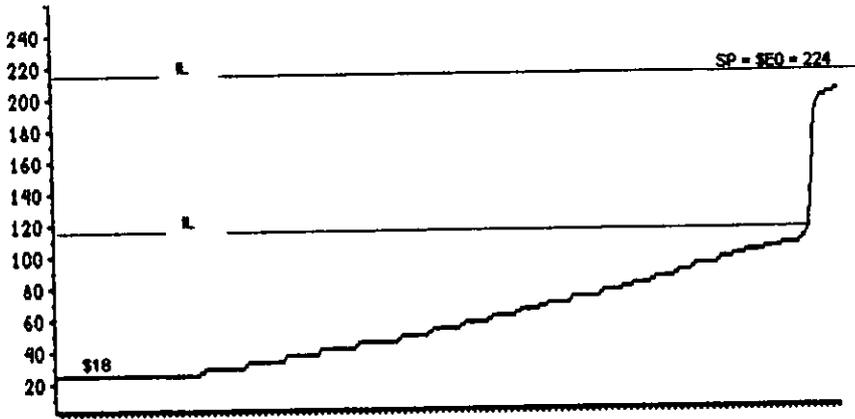


Fig. 7.14

Para poder hacer una comparación, la gráfica 7.15 al igual que la 7.13 se genera al intensificar las reglas correspondientes al rango "Penumbra", solo que las intensifica en menor grado al presentar como conclusión "AB". Por lo cual se observa un ligero decremento en la altura y la pendiente del escalón, así como un achatamiento en sus extremos.

MA	AB	MA	MA	MA
MA	AB	MA	MA	MA
MA	AB	MA	MA	MA



Fig. 7.15

La última de esta serie de gráficas muestra una respuesta escalonada. Como se puede ver en su FAM, cuatro de las nueve zonas están bajo el dominio de la conclusión "MA". La participación de este singleton es el que genera el aplanamiento de la respuesta en estas zonas. Para una mayor comprensión de esta gráfica, se marcan en ella las áreas de dominio de cada conjunto difuso, de forma que se pueda ir siguiendo mediante la ayuda de la FAM la forma de la gráfica.

SA MA MA SA MA MA SA SA SA  
 SA MA MA SA MA MA SA SA SA  
 SA MA MA SA MA MA SA SA SA

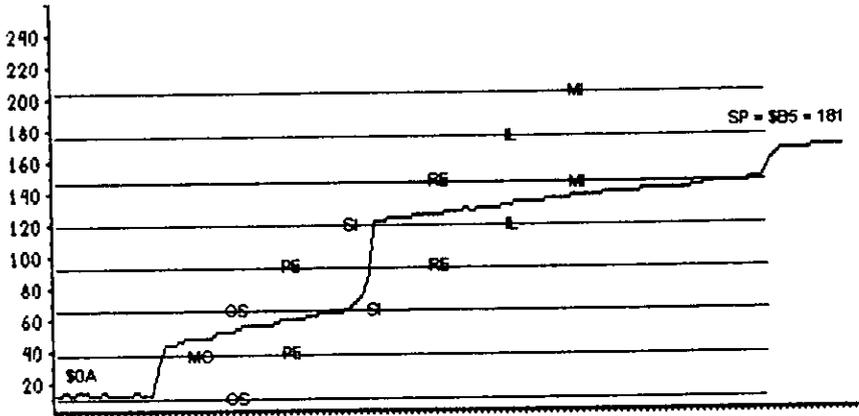


Fig. 7.16

• Pruebas sobre los conjuntos difusos de salida (singletons).

Cada una de las gráficas anteriores, se generaron utilizando los conjuntos difusos de salida (singletons) propuestos desde el capítulo 5. Al cambiar la distribución de estos conjuntos, se obtuvieron las siguientes dos respuestas.

AB AB AB AB AB  
 AB AB AB AB AB  
 AB AB AB AB AB



Fig. 7.17

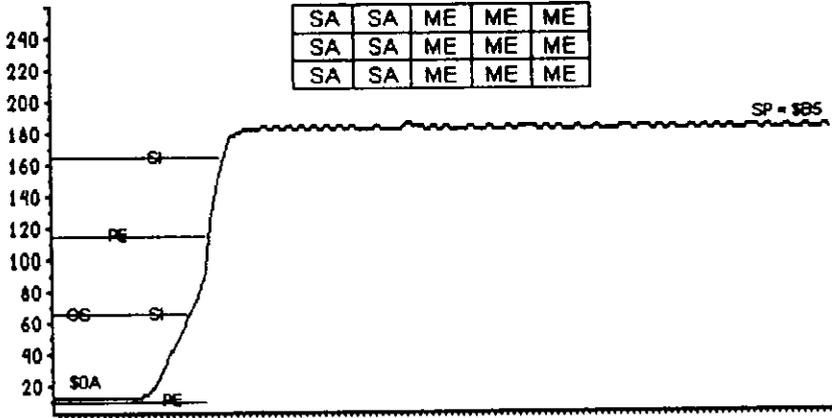


Fig. 7.18

La base de reglas utilizada para la primera de las dos pruebas, es la misma que generó la gráfica de la figura 7.11, y como se puede ver la pendiente ahora es menor, de forma que la transición de "MA" a "AB" se lleva a cabo de una manera más suave. El mismo efecto se puede observar en la figura 7.18, en la que se alcanza a observar el punto en donde comienza a perder fuerza la acción de "SA" y comienza la de "ME". Como parámetro de comparación se pueden observar los segmentos verticales de la figura 7.16, generados también a partir del sigleton "SA", los cuales presentan una pendiente más pronunciada. En términos generales, el efecto de este cambio, fue atenuar las magnitudes de los niveles de salida del sistema, aunque bien con una distribución adecuada puede darse el efecto contrario. La siguiente tabla muestra los valores que definen a los singletons utilizados en estas pruebas y en las anteriores.

Sinleton	valor anterior	valor actual
MA	\$1E	\$1E
AB	\$3C	\$23
SA	\$5A	\$28
ME	\$78	\$3C
SC	\$96	\$46
CE	\$B4	\$50
CO	\$F0	\$96

**7.1.3 Pruebas al controlador de iluminación.** Tal y como se menciona en capítulos anteriores, el controlador de iluminación actuará modificando la magnitud del voltaje que se le aplica a las lámparas, modificando así su intensidad luminosa. En el ININ, prácticamente la totalidad de las áreas de trabajo son iluminadas mediante lámparas fluorescentes, las cuales para su correcto funcionamiento, requieren un nivel de voltaje ubicado en un rango muy estrecho; aún así, estas pequeñas variaciones de voltaje no producen cambios en la iluminación. Debido a este problema, las siguientes pruebas fueron realizadas en un pequeño cuarto que contaba con una lámpara incandescente.

Con la base de reglas planteada originalmente en el capítulo 5 para este controlador, se obtuvieron tres gráficas que muestran su comportamiento bajo diferentes situaciones. En la primera de ellas, se tiene inicialmente un nivel de iluminación de 10 (sin unidades), y se requiere elevarlo a 119, punto en el que la intensidad se clasifica difusamente como "Semi-iluminado" en mayor grado, e "iluminado" en grado menor.

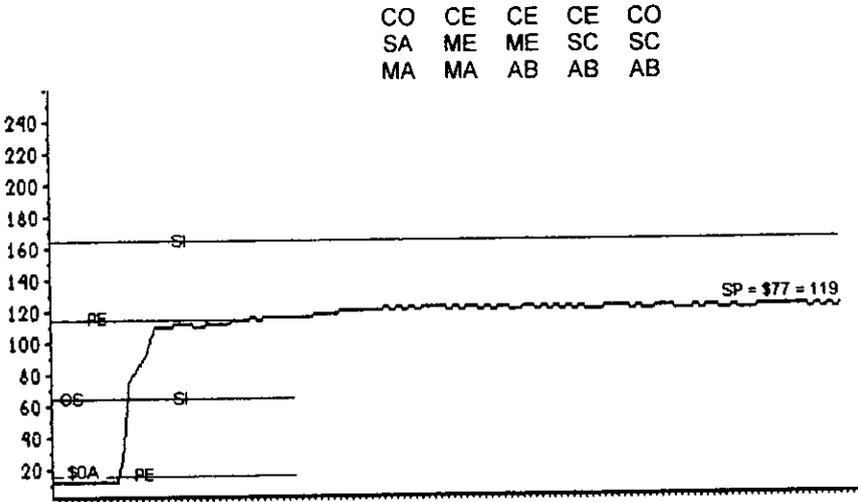


Fig. 7.19

Para la siguiente gráfica, el ejercicio es parecido al anterior, solo que el set-point se ubica en el valor de 197, por lo cual, se le puede definir difusamente como "Iluminado" y "Brillante". Observando con detenimiento las dos respuestas, surge inmediatamente un detalle muy importante, y es que mientras en la figura 7.19 el set-point es alcanzado en un lapso de tiempo muy corto, en la segunda jamás es alcanzado.

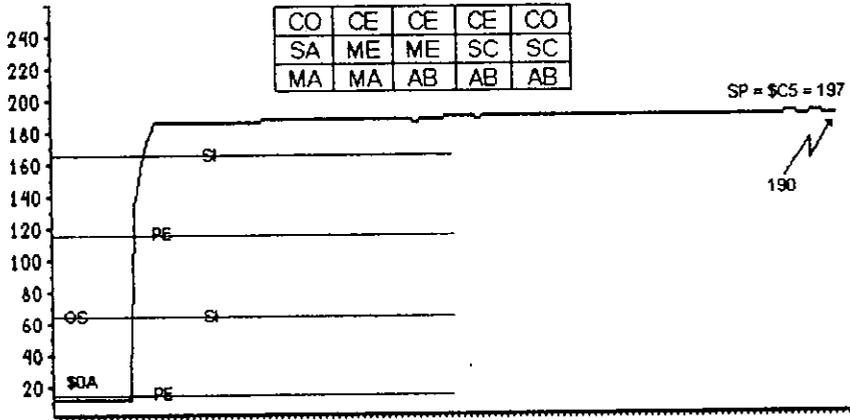


Fig. 7.20

Todavía con la misma base de reglas, se invierten los setpoints, con la finalidad de observar que es lo que pasa cuando el nivel de iluminación se encuentra en descenso. En esta situación, el error es "Grande" y el nivel de iluminación se clasifica como "Oscuro". Al igual que en el caso anterior, el set-point no se alcanza mientras transcurre la gráfica.

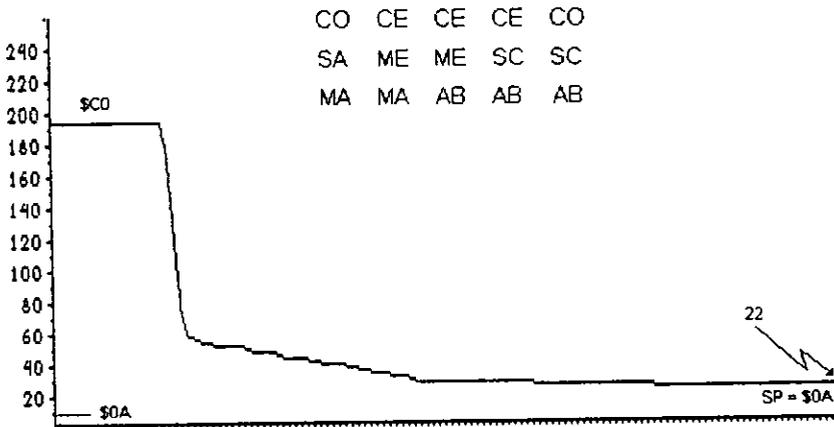


Fig. 7.21

Observando las gráficas 7.20 y 7.21, se puede afirmar sin ninguna dificultad que el problema se encuentra en la zona en donde el error es "Pequeño", así que ahora se propone un cambio en esta parte de la base de reglas. Las gráficas 7.22 y 7.23 muestran el resultado obtenido a partir de esta modificación.

CO	CE	CE	CE	CO
SA	ME	ME	SC	CE
SA	SA	SA	SA	SA

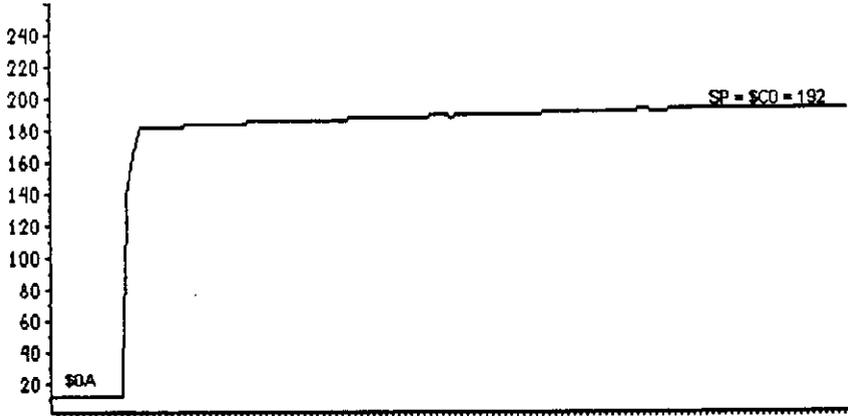


Fig. 7.22

CO	CE	CE	CE	CO
SA	ME	ME	SC	CE
SA	SA	SA	SA	SA

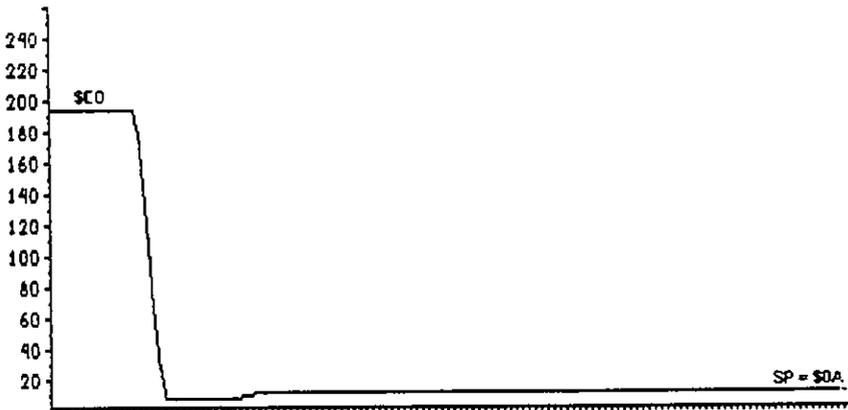


Fig. 7.23

El cambio anterior consistió en modificar las conclusiones de las reglas que actúan en la zona donde el error es "Pequeño". Con la finalidad de mejorar aún más las respuestas hasta ahora obtenidas, nuevamente se proponen otros cambios en la base de reglas, derivándose la siguiente gráfica.

SC	SC	CE	CE	CO
ME	ME	SC	SC	SC
ME	ME	ME	SC	SC

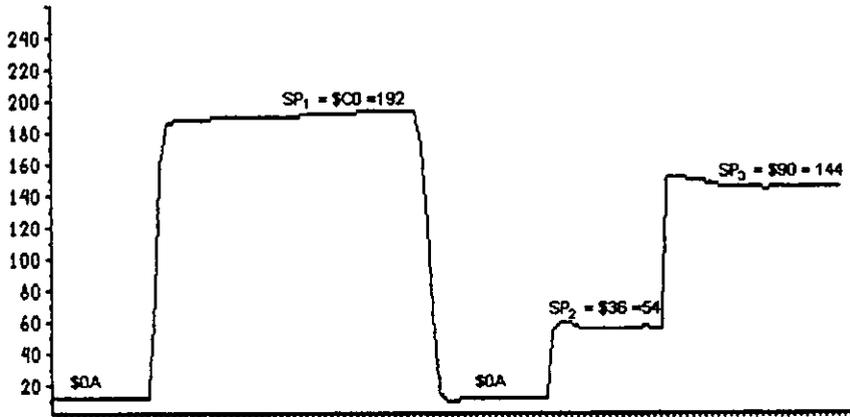


Fig. 7.24

Estos últimos cambios realizados, afectan varias zonas de las gráficas anteriores. En términos generales, ahora se tiene una mejor respuesta en la zona de error "Pequeño" independientemente del nivel de iluminación. También se modificaron las reglas de los errores "Regular" y "Grande", y debido en parte a esta acción, se observa un ligero sobre tiro en las dos últimas respuestas de la gráfica.

**7.1.4 Pruebas al controlador de Temperatura.** El lugar en donde se llevó a cabo el control de temperatura, es el mismo que se menciona en las pruebas de iluminación. Esto porque se requería una área reducida debido a las limitaciones de los actuadores, los cuales eran 2 calentadores eléctricos de 800 watts para el proceso de calentamiento, y un extractor de aire para enfriar el cuarto.

Para esta parte se llevaron a cabo únicamente 4 pruebas, 2 de calentamiento y 2 de enfriamiento. La gráfica 7.25 corresponde a la base de reglas original para calentamiento, y como se puede ver, al cabo de 10 minutos que duró la elaboración de la gráfica no se alcanzó el set-point. Después de este tiempo, la graficación presentaba la misma forma sin que se registrara ningún cambio, por lo que se determinó que con tal base de reglas jamás se alcanzaría el set-point.

Calentar				Enfriar			
CO	SC	ME	AB	SC	CE	CE	CO
CE	ME	SA	AB	SA	ME	ME	SC
SC	SA	AB	MA	MA	AB	AB	SA

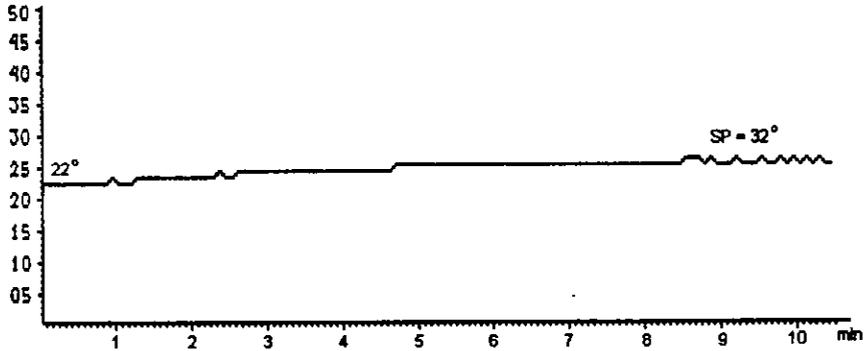


Fig. 7.25

Después de realizar varias pruebas adicionales de calentamiento, se optó por reportar la siguiente gráfica, en la cual se observa como el set-point ahora si se alcanza y en un tiempo realmente corto.

Calentar				Enfriar			
CO	CE	CE	CE	SC	CE	CE	CO
CE	SC	SC	SC	SA	ME	ME	SC
SC	ME	ME	ME	MA	AB	AB	SA

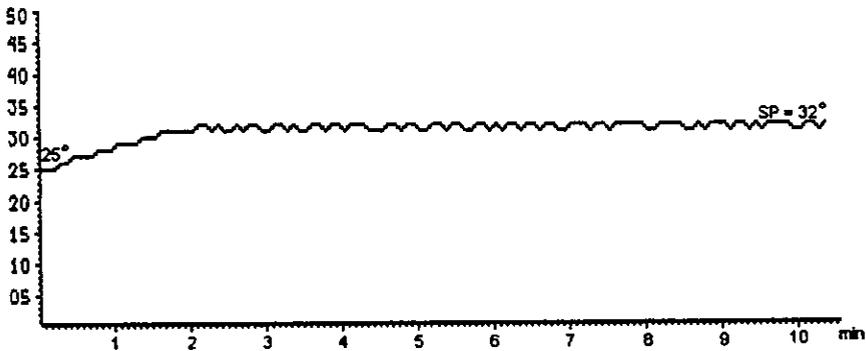


Fig. 7.26

Las gráficas 7.27 y 7.28 corresponden al proceso de enfriamiento, observándose en la primera un descenso de la temperatura más rápido en los primeros dos minutos; pasado este tiempo, el enfriamiento prácticamente se lleva a cabo por ventilación natural.

Calentar				Enfriar			
CO	CE	CE	CE	SC	CE	CE	CO
CE	SC	SC	SC	SA	ME	ME	SC
SC	ME	ME	ME	MA	AB	AB	SA

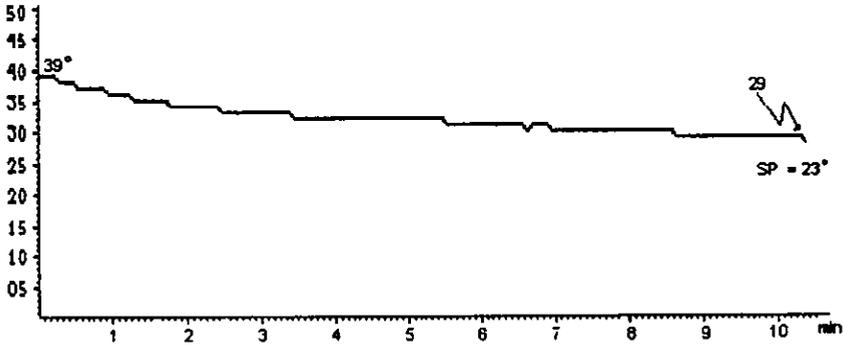


Fig. 7.27

Al modificar las reglas se tiene un proceso de enfriamiento mucho más rápido, de tal forma que el setpoint es alcanzado al transcurrir 2.5 minutos.

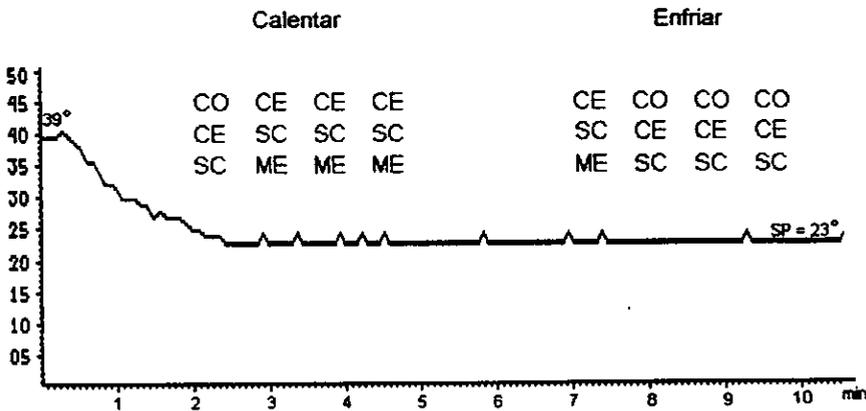


Fig. 7.28

## 7.2 Conclusiones.

La lógica difusa representa un método alternativo para el desarrollo de sistemas de control, comparable en eficiencia y rendimiento con las técnicas tradicionales. El poder representar artificialmente experiencias y conocimientos adquiridos por un experto, nos permite utilizar esta información en diversas tareas de automatización, sobre todo en aquellas en las que por su complejidad se requiere la acción directa de un humano experto.

La lógica difusa nos provee de un conjunto de herramientas matemáticas que nos permiten manejar conceptos y cantidades que no estén perfectamente definidas, y que por el contrario presenten altos grados de imprecisión. Mediante las matemáticas tradicionales, es imposible tratar con expresiones lingüísticas, las cuales por no poder ser exactamente cuantificadas, quedan fuera de toda teoría matemática clásica. En el desarrollo de un controlador difuso, se requiere manipular matemáticamente el conocimiento, el cual es proporcionado por un experto en su lenguaje natural, de ahí que un modelo matemático implementado en base a cantidades tales como "caliente", "bajo", "rápido", "muy lento", etc. requiere de una base teórica diferente, siendo ésta la lógica difusa.

El diseño formal de cualquier sistema de control no difuso, exige un fuerte conocimiento de la teoría del control clásico y moderno, además de la obtención de un modelo matemático que describa el comportamiento del sistema. Bajo estas condiciones, el diseño puede llegar a ser una tarea que requiera un alto grado de especialización. Por el contrario, el control difuso posee una metodología de diseño mucho menos complicada, y en la etapa de implementación muchas veces la parte más laboriosa es la que tiene que ver con la obtención del conocimiento, sin embargo esto se resuelve la gran mayoría de las veces mediante la simple observación del proceso o con la aplicación de cuestionario a operadores.

Por otra parte, cada una de las aplicaciones basadas en lógica difusa, requieren procesar la información contenida en las bases de conocimiento para así entregar resultados. Dependiendo del tipo de aplicación, la velocidad para el tratamiento de datos puede llegar a ser un aspecto importante; en el caso de los sistemas expertos por ejemplo, se requiere no solo de alta velocidad de procesamiento, si no también del almacenamiento de grandes volúmenes de datos. Debido a esto, los sistemas expertos son implementados en computadoras que cuenten con las capacidades que la aplicación requiere. En el caso del control, las aplicaciones por lo general son menos exigentes en cuanto a recursos de hardware, tal es así que en la actualidad la mayoría de los diseños son desarrolladas a base de microcontroladores y circuitos integrados especializados como el AL220 (antes NLX220) y varios mas.

En nuestro caso, se buscó un microcontrolador que contara con los recursos suficientes para implementar un sistema de control para aplicaciones no muy demandantes de memoria y altas velocidades de procesamiento. Los microcontroladores MC68HC11 resultaron ser los más adecuados, sobre todo porque integran en la misma pastilla un convertidor A/D de 8 bits, pieza fundamental en la estructura del controlador difuso, así también por la facilidad de conseguirlo comercialmente. El dispositivo cuenta además entre otras características con un sistema de temporización, así como con interfase de comunicación serial, la cual en este proyecto resultó de gran utilidad, específicamente en la etapa de obtención de resultados, los mismos que fueron graficados y analizados mediante una computadora personal. Cabe mencionar también que existen muchos sistemas cuyos modelos matemáticos involucran cálculos que combinan operaciones aritméticas, funciones trigonométricas, exponentes, logaritmos, etc. de tal forma que la obtención de un resultado consume una cantidad de tiempo que puede llegar a ser significativa. El hecho de que en un controlador difuso los resultados se logren a través de operaciones aritméticas básicas y simples como son sumas, restas, multiplicaciones y divisiones, confirma el por qué un microcontrolador resulta más que adecuado para implementar este tipo de sistemas.

De acuerdo a lo mencionado en los primeros capítulos, la estructura de un controlador difuso contempla tres etapas básicas: difusificación, evaluación de reglas y desdifusificación; en caso de ser necesario se integran otras dos: normalización y desnormalización. Los principios con los que operan cada una de las tres etapas básicas, están apoyados en conceptos de lógica de conjuntos difusos, por lo que la programación de las rutinas respectivas se facilita en gran manera, sin que sea necesario un programador experto para la elaboración del programa. Una vez que se cuenta con la programación de las rutinas básicas, estas pueden ser utilizadas para múltiples aplicaciones, ya que independientemente de la tarea que tenga que realizar el controlador, el tratamiento difuso de los datos siempre será el mismo, es decir, difusificar, inferir y desdifusificar. La única diferencia estará en el contenido de las bases de conocimiento y en la adaptación de la información proveniente tanto de los sensores, como de la etapa de desdifusificación.

En estas condiciones se tiene entonces un sistema de control realmente adaptable para diversas aplicaciones. En nuestro proyecto por ejemplo, se realiza el control sobre dos variables independientes, las cuales tienen definidos sus propios conjuntos difusos de entrada y de salida, así como bases de reglas distintas. En los dos casos no se siguió ningún criterio para la selección de parámetros tales como el número de conjuntos, su forma, ubicación, número de reglas, método de inferencia, método de desdifusificación, etc. El controlador trabajó con todos estos valores obtenidos en base a suposiciones, pero durante el desarrollo de las pruebas fueron modificados gradualmente hasta que se logró el control deseado.

Durante la etapa de pruebas las primeras modificaciones fueron hechas sobre los conjuntos difusos, para ver la respuesta del sistema al incrementar o disminuir el número de estos. De aquí se observó que mientras más conjuntos sean, se tiene control sobre rangos más estrechos de la señal de entrada, o lo que es lo mismo se tiene un controlador con mayor resolución. Igualmente fue modificada la base de reglas original, para observar las distintas respuestas logradas al intensificar o atenuar las conclusiones de las reglas. Durante estas pruebas se generaron también algunas gráficas que muestran claramente los efectos de los cambios, en base a los cuales se logró por ejemplo acelerar o atrasar respuestas, y controlar el sobre tiro (over shot). Esto último es de particular interés en muchos sistemas, en los que un incremento importante en la variable a controlar por arriba del set-point puede repercutir negativamente. Finalmente se reportaron las gráficas definitivas de las respuestas correspondientes a la temperatura y la iluminación, las cuales se comparan con gráficas obtenidas a partir de las bases de conocimientos propuestas inicialmente, de manera que se puede ver claramente el efecto de los cambios realizados.

# ANEXO 1

(Listado del programa de control difuso)

```

org $e000

OPTION:          equ    $1039
ADCTL:          equ    $1030
ADR1:           equ    $1031
ADR2:           equ    $1032
ADR3:           equ    $1033

                ldaa   #$80
                staa   OPTION
                ldaa   #$10
                staa   $1021
                lds    #$00ff
                ldaa   #$00
                staa   $1024
                ldaa   #$30
                staa   $102b
                ldaa   #$00
                staa   $102c
                ldaa   #$0c
                staa   $102d

clr U4

switchs        equ    $0000
Vi              equ    $0001
CONT           equ    $0002
CONT1          equ    $0003
X_BKUP         equ    $0004
GRAD MEM       equ    $0006
VARI           equ    $0007
FUNCION        equ    $0009
Vi_BKUP        equ    $0011
RESULTADO      equ    $0012
SETPOINT       equ    $0013
U1             equ    $0014
U2             equ    $0015
XX             equ    $0016
SUMPRODS       equ    $0017
SUMGRADS       equ    $0019
SALIDA         equ    $001B
TEMPERR        equ    $01

TIEMPO1        equ    $001C
TIEMPO2        equ    $001E
TIEMPO3        equ    $0020
RESCAL         equ    $0022
RESENF         equ    $0024
RESILU         equ    $0026
SETPTM         equ    $0028
SETPILU        equ    $0029
BUFFER         equ    $002A
TCNT2          equ    $0040
TCNT3          equ    $0042
TCNT1          equ    $0044
U3             equ    $0046
switemp        equ    $0047
swillum        equ    $0048
Reg1_Tx        equ    $0049
NUMCIC         equ    $004A
U4             equ    $004B
U_ANT          equ    $004C
BUFF           equ    $004E

;.....
;* switches = 01 -----> 1=temp 0=ilum
;*          = 02 -----> 1=err temp. 0=err ilum
;*          = 04 -----> 1= SP>Vi, 0= SP<Vi
;.....

```

```

;*      = 08 -----> bandera de #=1 en zonas de no intersección *
;*      = 10 -----> one shoot para activación de interrupciones *
;*      = 20 -----> inicio de ciclo senoidal para OC3 *
;*      = 40 -----> inicio de ciclo senoidal para OC4 *
;*      = 80 -----> singletons recortados *
;.....

```

```

ini      ldx #$ffff
dex
bne ini
clr NUMCIC
cli
ldaa #$04
staa $1023
staa $1022
ldaa #$0f
jsr comando
ldaa #$0f
jsr comando
ldaa #$3C      ;**** CONFIGURA 8 BITS Y 1 LINEA
jsr comando
jsr dly2mS
ldaa #$3C
jsr comando
jsr dly2mS
ldaa #$3C
jsr comando
jsr dly2mS
ldaa #$3C
jsr comando
jsr dly2mS
ldaa #$01      ;**** LIMPIA PANTALLA
jsr comando
jsr dly2mS
ldaa #$02      ;**** HOME
jsr comando
jsr dly2mS
ldaa #$a3      ;.....
staa SETPTEM   ;.....
ldaa #$55      ;.....
staa SETPILU   ;.....
clr switchs
ldaa #$ff
staa $1000
ldd #$0050     ;.....??????
std BUFF      ;.....??????

SHOOT     ldy #$000A      ;***** temporizacion para 5 mS.
SHOOT28   psha
           ldab #$04
           stab $0102
           ldaa $0103
           cmpa #$08
           bne SHOOT12
           ldx #$01ff
SHOOT24   dex
           bne SHOOT24
SHOOT25   ldaa #$04
           staa $0102
           ldaa $0103
           cmpa #$00
           bne SHOOT25
           ldd #MESS2
           jsr DISP
           bra SHOOT13
SHOOT12   jmp SHOOT4
SHOOT13   jsr TECLA      ;Checa si hay tecla presionada
;         ldd #MESS2
;         jsr DISP
           ldaa #SC5
           jsr comando
           ldaa SETPTEM
           jsr NUM
           ldaa #$01

```

Programa de control difuso

```

SHOOT8      staa U1
             jsr TECLA
             cmpb #508      ; Salir de modo de modificaci3n de SP
             beq SHOOT14
             bra SHOOT15
SHOOT14     jmp SHOOT5
SHOOT15     cmpb #580      ; Conmuta SP para modificaci3n
             bne SHOOT6
             ldab #501
             cmpb U1
             bne SHOOT7
             idaa #501     ; Cambia a SP de Iluminaci3n
             jsr comando
             ldd #MESS3
             jsr DISP
             ldaa #5C5
             jsr comando
             ldaa SETPILU
             jsr NUM
             ldaa #500
             staa U1
             bra SHOOT8
SHOOT7     idaa #501     ; Cambia a SP de Temperatura
             jsr comando
             ldd #MESS2
             jsr DISP
             ldaa #5C5
             jsr comando
             ldaa SETPTM
             jsr NUM
             ldaa #501
             staa U1
SHOOT6     bra SHOOT8
             cmpb #520      ; Incrementa SP
             bne SHOOT9
             ldaa U1
             cmpa #500
             beq SHOOT10
             inc SETPTM
             ldaa #5C5
             jsr comando
             ldaa SETPTM
             jsr NUM
             ldab #500
             bra SHOOT8
SHOOT10    inc SETPILU
             ldaa #5C5
             jsr comando
             ldaa SETPILU
             jsr NUM
             ldab #500
             bra SHOOT8
SHOOT9     cmpb #540      ; Decrementa SP
             bne SHOOT26
             bra SHOOT27
SHOOT26    jmp SHOOT8
SHOOT27    ldaa U1
             cmpa #500
             beq SHOOT11
             dec SETPTM
             ldaa #5C5
             jsr comando
             ldaa SETPTM
             jsr NUM
             ldab #500
             jmp SHOOT8
SHOOT11    dec SETPILU
             ldaa #5C5
             jsr comando
             ldaa SETPILU
             jsr NUM
             ldab #500
             jmp SHOOT8
SHOOT5     ldaa #501

```

Programa de control difuso

```

        jsr comando
SHOOT4   pula
SHOOT1   ldaa NUMCIC          ;-----
        cpa $S1E ;-----> | Número de ciclos para muestreo |
        bmi SHOOT22        ;|         de iluminación         |
        bra SHOOT23        ;-----
SHOOT22  jmp SHOOT28
SHOOT23  clr NUMCIC
        ldaa SETPILU
        staa SETPOINT
        ldaa $S70
        andaa switchs
        staa switchs
        jsr cnv
        dey
        bne SHOOT18
        bra SHOOT19
SHOOT18  jmp SHOOT1
SHOOT19  ldaa SETPTM
        staa SETPOINT
        ldaa $S01
        oraa switchs
        staa switchs
        jsr cnv
        jmp SHOOT

```

```

;-----
;                                         Comienza Programa                                     |
;-----

```

```

MESS1   fcb 'Transfiriendo', $fe
MESS2   fcb $80, 'SP de Te', $A0, 'mp.', $fe
MESS3   fcb $80, 'SP de Il', $A0, 'um.', $fe
MESS4   fcb $80, 'Salir', $fe

cnv     psha
        pshb
        pshx
        pshy
        ldaa $S00
        staa ADCTL
cnv0    ldaa $S01
        bita switchs ; **** 1= Temp.  0= Ilum.
        beq cnv1
        pshx
        ldd $S0000
        std VAR1
        ldx $S000a
cnv4    ldaa $S80
        staa ADCTL
cnv2    bita ADCTL
        beq cnv2
        ldaa $S00
        ldab ADRI ; **** carga temperatura digitalizada
        addb #TEMPERR
        addd VAR1
        std VAR1
        dex
        bne cnv4
        ldd VAR1
        ldx $S000a
        idiv
        xgdx
        pulx
        tba
        staa Vi
        ldaa $S05
        mul
        stab VAR1

```

Programa de control difuso

```

ldab Vi
ldx #S000a
idiv
xgdx
addb VAR1
jmp difusi
cnv1    pshx
        ldd #S0000
        std VAR1
        ldx #S000a
cnv5    ldaa #S82
        staa ADCTL
        ldaa #S80
cnv3    bita ADCTL
        beq cnv3
        ldaa #S00
        ldab ADR3
        addd VAR1
        std VAR1
        dex
        bne cnv5
        ldd VAR1
        ldx #S000a
        idiv
        xgdx
        stab VAR1
        pulx
        ldab #Sff
        subb VAR1 ; **** carga iluminación digitalizada
        cmpb #Sff
        bne difusi
        ldab #Sfc

difusi  stab Vi ; **** carga de Vi
        stab Vi_BKUP
        ldd #S0000
        std FUNCION
        std FUNCION+2
        std FUNCION+4
        std FUNCION+6
        clr CONT1
        ldab Vi
difusi_14 ldaa #S01
        bita switches
        beq difusi_1
        ldx #BC_TEMP
        ldaa #S02
        bita switches
        beq difusi_2
        ldx #BC_ERTEMP
        bra difusi_2
difusi_1  ldx #BC_ILUM
        ldaa #S02
        bita switches
        beq difusi_2
        ldx #BC_ERILUM
difusi_2  ldaa #S01
        staa CONT ; CONT = 1
        stx X_BKUP ; X_BKUP = X
difusi_10 pshx
        ldab #S03
        abx ; [X + 3] = 3er. punto
        ldab 0,x
        ldaa #S00
        std BUFFER
        ldab Vi
        cpd BUFFER
        bpl difusi_11
        bra difusi_12
difusi_11 jmp difusi_9
difusi_12 pulx
        pshx
        ldab #S01

```

```

abx          : [X + 1] = 1er. punto
ldab 0,x
ldaa #$00
std BUFFER
ldab Vi
cpd BUFFER
bmi difusi_11
pulx
pshx
ldab #$02
abx          : [X + 2] = 2o. punto
ldab 0,x
ldaa #$00
std BUFFER
ldab Vi
cpd BUFFER
bpl difusi_5
beq difusi_3
pulx
pshx
ldab #$04
abx
ldd #$0000
cpd 0,x
bne difusi_4
ldab #$64
stab GRAD_MEM
ldaa #$08
oraa switchs
staa switchs
jmp difusi_8
difusi_4    ldaa Vi
pulx
pshx
ldab #$04
abx
ldab 0,x
mul
std VAR1
ldaa Vi
pulx
pshx
ldab #$05
abx
ldab 0,x
mul
ldx #$0064
idiv
xgdx
addd VAR1
std VAR1
pulx
pshx
ldab #$06
abx
ldd VAR1
subd 0,x
stab GRAD_MEM
bra difusi_8
difusi_3    ldab #$63
stab GRAD_MEM
bra difusi_8
difusi_5    pulx
pshx
ldab #$08
abx
ldd #$0000
cpd 0,x
bne difusi_6
ldab #$64
stab GRAD_MEM
bra difusi_8
difusi_6    pulx
pshx

```

```

ldab #S08
abx
ldaa Vi
ldab 0,x
mul
std VAR1
pulx
pshx
ldaa Vi
ldab #S09
abx
ldab 0,x
mul
ldx #S0064
idiv
xgdx
addd VAR1
std VAR1
pulx
pshx
ldab #S0a
abx
ldd 0,x
subd VAR1
stab GRAD_MEM
difusi_8   ldaa CONT
          ldy #FUNCION
          ldab CONT1
          aby
          staa 0,y
          ldaa #S08
          bita switchs
          beq difusi_16
          inc CONT1
          ldaa #Sf7
          anda switchs
          staa switchs
difusi_16  ldaa GRAD_MEM
          staa 4,y

zzz       inc CONT1
difusi_9   ldab #S0b
          pulx
          abx
          inc CONT      ; CONT = CONT + 1
          ldaa CONT
          ldy X_BKUP
          cmpa 0,y
          bis difusi_13
zzz2      ldaa #S02
          bita switchs
          bne infer
          oraa switchs
          staa switchs      ; Pone en '1' el segundo bit de switchs
          ldaa Vi
          staa VAR1
          ldaa #S00
          ldab Vi
          std BUFFER
          ldab SETPOINT
          subd BUFFER      ; [A]=SETPOINT - Vi
          bmi difusi_15
          tba
          psha
          ldaa #S04      ;**** pone en 1 el 3er bit de switchs
          oraa switchs
          staa switchs
          pula
          staa Vi
difusi_15  jmp difusi_14
          ldab SETPOINT
          ldaa #S00
          std BUFFER
          ldab VAR1

```

```

        subd BUFFER
        tba
        staa Vi
        ldaa #9FB
        anda switchs
        staa switchs
        jmp difusi_14
difusi_13 jmp difusi_10
infer    ldd #S0000
        std SALIDA
        std SALIDA+2
        std SALIDA+4
        std SALIDA+6
        ldaa #Sf5
        anda switchs ; ***** LIMPIA BANDERA DE CICLO DE
        staa switchs ; FUZIFICACION DE ERROR
        ldaa #S01
        bita switchs
        beq infer1 ; ***** SI ES CICLO DE ILUMINACION ENTONCES ....
        ldaa #S04
        bita switchs
        beq infer2 ; ***** SI ES CICLO DE ENFRIAR ENTONCES .....
        ldx #BR CALENTAR
        bra infer3
infer2   ldx #BR ENFRIAR
        bra infer3
infer1   ldx #BR ILUM
infer3   clr CONT ; CONT = 0
infer11  pshx ; GUARDA BR.....
        ldx #FUNCION ; ***** calculo de la columna de la FAM *****
        ldab CONT ; IX = IX + [FUNCION + CONT]
        abx ; IX = IX + [B]
        ldab 0,x ; [B] = [IX + B] = Primer antecedente
        cmpb #S00
        beq infer14
        bra infer13
infer14  jmp infer7
infer13  ldaa 4,x ; [A] = Grado de membres;a correspondiente
        staa U1 ; U1 = [A] = 'a' del primer antecedente
        ldaa #S02 ; b = (b*2) - 1
        mul
        decb
        decb
pulkx   ; carga en X el valor de BR.....
pshx
abx
        ldx 0,x ; IX = vector columna
        stx X_BKUP
        ldy #FUNCION+2
        ldab CONT
        aby ; IY apunta al segundo antecedente
        ldab 0,y
        cmpb #S00
        beq infer6
        decb
        abx ; IX apunta a la 1a. conclusión
        ldab 0,x ; Carga en B de la primera conclusión
        ldx #SALIDA-1
        abx ; IX apunta al número de salida correspondiente a la conclusión disparada
        ldaa 4,y ; Carga del grado de mem. del segundo antecedente
        cmpa U1 ; aerror - avar_cont
        bhs infer4
        cmpa 0,x ; Si avar_cont es menor Entonces Obtiene el máximo
        bls infer5
        staa 0,x ; Guarda el mínimo
        bra infer5
infer4   ldaa U1
        cmpa 0,x ; Obtiene el máximo
        bls infer5
        staa 0,x ; Guarda el mínimo
infer5   ldaa CONT
        cmpa #S00
        bhi infer6
        ldy #FUNCION+3

```

```

ldab 0,y      ; Carga en B del 2o. antecedente de la 2a. regla
cmpb #$00
beq infer7
decb
ldx X_BKUP
abx           ; IX apunta a la segunda conclusión
ldab 0,x      ; Carga la segunda conclusión
ldx #SALIDA-1
abx
ldaa 4,y      ; Carga del grado de mem. del segundo antecedente
cmpa U1
bhs infer8
cmpa 0,x      ; Obtiene el máximo
bls infer9
staa 0,x      ; Guarda el mínimo
bra infer9
infer8        ldaa U1
cmpa 0,x      ; Obtiene el máximo
bls infer9
staa 0,x      ; Guarda el mínimo
infer9        bra infer7
infer6        ldy #FUNCION+1
ldab CONT
aby
ldab 0,y      ; Carga en B del 2o. antecedente de la 4a. regla
decb
ldx X_BKUP
abx           ; IX apunta a la cuarta conclusión
ldab 0,x      ; Carga la cuarta conclusión
ldx #SALIDA-1
abx
ldaa 4,y      ; Carga del grado de mem. del segundo antecedente
cmpa U1
bhs infer10
cmpa 0,x      ; Obtiene el máximo
bls infer7
staa 0,x      ; Guarda el mínimo
bra infer7
infer10       ldaa U1
cmpa 0,x      ; Obtiene el máximo
bls infer7
staa 0,x      ; Guarda el mínimo
infer7        pulx
inc CONT
ldaa CONT
cmpa #$02
bmi infer12
bra dedifusi
infer12       jmp infer11
dedifusi      ldd #$0000
std SUMPRODS
std SUMGRADS
ldaa #$01
staa CONT
bita switchs
beq dedifusi2
ldaa #$04
bita switchs
bne dedifusi1
ldx #BCS_FRIO
bra dedifusi3
dedifusi1     ldx #BCS_CALIENTE
bra dedifusi3
dedifusi2     ldaa #$04
bita switchs
beq dedifusi6
ldaa SETPILU
suba Vi_BKUP
cmpa #$10
bhi dedifusi7
ldx #BCS_ILUMI
ldaa #$80
oraa switchs

```

Programa de control difuso

```

    staa switches
    bra dedifusi3
dedifusi7 ldx #BCS_ILUM
    bra dedifusi3
dedifusi6 ldaa Vi_BKUP
    suba SETPILU
    cmpa #510
    bhi dedifusi8
    ldx #BCS_ILUM1
    ldaa #580
    oraa switches
    staa switches
    bra dedifusi3
dedifusi8 ldx #BCS_ILUM
dedifusi3 ldaa 0,x
    staa VAR1           ;Número de funciones de salida
    pshx
    pulx
dedifusi5 pshx
    ldab CONT
    abx
    ldaa 0,x
    staa XX
    ldx #SALIDA-1
    ldab CONT
    abx
    ldab 0,x
    stab GRAD_MEM
    cmpb #500
    beq dedifusi4
    ldaa #500
    ldab GRAD_MEM
    addd SUMGRADS
    std SUMGRADS
    ldab GRAD_MEM
    ldaa XX
    mul
    addd SUMPRODS
    std SUMPRODS
dedifusi4 inc CONT
    pulx
    ldaa VAR1
    cmpa CONT
    bhs dedifusi5
    ldd SUMPRODS
    ldx SUMGRADS
    idiv
    xgdx
    stab RESULTADO

;-----
;                               Fin del controlador difuso                               |
;-----

Desnorm  ldaa #501
    bita switches
    beq Desnorm1
dedifusi9 ldaa RESULTADO           ;***** Desnormalización de la Temperatura *****
    suba #51E           ; {
    ldab #554           ; i + {(RESULTADO - 30) x .084}
    mul
    ldx #503E8
    idiv
    inx           ; }
    xgdx
    stab CONT
    ldab #512
    subb CONT
    cmpb #500
    bne Desnorm2
    ldab #501
Desnorm2 stab CONT

```

```

        ldaa #$9d
        mul
        std VAR1
        ldab CONT
        ldaa #$03
        mul
        tba
        ldab #$00
        addd VAR1
        xgdx
        stx TCNT2
        ldaa switchs
        staa switemp
        jmp Desnorm3
Desnorm1  ldaa #$10           ; 'i' 'i' Desnormalización de la Iluminación 'i' 'i'
        bita $1022
        beq Desnorm13
        inc U4
Desnorm13 ldaa RESULTADO
        suba #$1E
        cmpa #$00
        bne Desnorm27
        ldaa #$01
Desnorm27 ldab #$80
        bitb switchs
        beq Desnorm23

        ldab #$02           ; '± Ajuste fino '±
        bra Desnorm24
Desnorm23 ldab #$12        ; factor
Desnorm24 mul
        std VAR1
        ldaa RESULTADO
        suba #$1E
        cmpa #$00
        bne Desnorm28
        ldaa #$01
Desnorm28 ldab #$80
        bitb switchs
        beq Desnorm25
        ldab #$00           ; '± Ajuste fino '±
        bra Desnorm26
Desnorm25 ldab #$00
Desnorm26 mul
        ldx #$64
        idiv
        xgdx
        addd VAR1
dedifus121 std VAR1
        ldaa #$04
        bita switchs
        beq Desnorm4
        ldd #$4114
        subd TCNT3
        addd VAR1
        std VAR1
        ldd #$4114
        subd VAR1
        cpd #$0800
        bmi Desnorm14
        bra Desnorm15
Desnorm14 ldd #$0800
Desnorm15 std TCNT3
        bra Desnorm3
Desnorm4  ldd #$4114        ; ***** Resta último resultado al contador TCNT3
        subd TCNT3
        subd VAR1
        std VAR1
        ldd #$4114
        subd VAR1
        cpd #$4114
        bhi Desnorm16
        bra Desnorm17
Desnorm16 ldd #$4100

```

```

Desnorm17 std TCNT3

Desnorm3  ldaa #$01
          bita switches
          beq Desnorm5

          ldaa #$10      ;(
          bita switches
          bne Desnorm5
          oraa switches ; Activa las interrupciones de temporización
          staa switches
          ldaa #$34
          staa $1023
          staa $1022   ;)
          ldaa switches
          staa switemp
          clr NUMCIC
          ldd #$0000
          std TCNT2
          ldd #$4114
          std TCNT3
          clr U_ANT
    
```

```

-----
Desnorm5  ldaa $FF
Desnorm22 deca
          bne Desnorm22
          ldaa #$01
          bita switches
          beq Desnorm18
          ldaa #$10
          bita $1022
          beq Desnorm18
          jsr Header
          pshx
          ldx #$0003
Desnorm30 ldaa TCNT2
          jsr Tx          ; ***** Tx FUNCION
          dex
          bne Desnorm30
          ldx #$0003
Desnorm31 tba
          jsr Tx          ; ***** Tx TCNT3 low
          dex
          bne Desnorm31
          ldx #$0003
Desnorm32 ldaa switches
          jsr Tx          ; ***** Tx switches
          dex
          bne Desnorm32
          ldx #$0003
Desnorm29 ldaa Vi_BKUP
          jsr Tx          ; ***** Tx Vi
          dex
          bne Desnorm29
          pulx
          ldaa $c5
          jsr comando
          ldaa Vi_BKUP
          jsr NUM
          ldaa RESULTADO
          jsr Tx          ; ***** Tx Resultado
          bra Desnorm18
Desnorm19 jmp rola
Desnorm18 puly
          pulx
          pulb
          pula
          rts
    
```

```

;-----
-
dly2mS    pshx      ;**** Retardo de 2mSegs.
          ldx #S02BC
dly2mS_1  dex
          bne dly2mS_1
          pulx
          rts

```

```

;-----
-
dly2.5    pshx      ;**** Retardo de 5 mS
          pshy
          ldx #S0007 ;32
dly2.5_2  ldy #S82DC
dly2.5_1  dey
          bne dly2.5_1
          dex
          bne dly2.5_2
          puly
          pulx
          rts

```

```

;-----
TECLA    pshx
          ldaa #S08
          staa $0102
          ldaa $0103
          cmpa #S00
          beq TECLA2
          ldx #S03ff
TECLA7   dex
          bne TECLA7
TECLA3   ldab #S08
          stab $0102
          ldab $0103
TECLA5   ldaa #S08
          staa $0102
          ldaa $0103
          cmpa #S00
          bne TECLA5
          bra TECLA1
TECLA2   ldaa #S04
          staa $0102
          ldaa $0103
          cmpa #S00
          beq TECLA1
          ldx #S03ff
TECLA8   dex
          bne TECLA8
TECLA4   ldaa #S04
          staa $0102
          ldab $0103
TECLA6   ldaa #S04
          staa $0102
          ldaa $0103
          cmpa #S00
          bne TECLA6
TECLA1   pulx
          rts

OC3      ldaa #S04
          bita switemp
          beq OC3_1

          ldaa #Sbf
          staa U2
          bra OC3_2

```

```

OC3_1  ldaa #$df
       staa U2

OC3_2  ldaa U2
       anda $1000      ;
       staa $1000      ;   Activa pulso en OC2 u OC3

OC3_4  ldaa #$ff      ; {
       deca
       bne OC3_4      ; }

       ldaa #$20
       staa $1023      ;   Limpia banderas de OC2 u OC3

       ldaa U2
       coma
       oraa $1000      ;{   Desactiva pulso de OC2 u OC3
       staa $1000      ;}

       ldaa #$20
       bita switches
       beq OC3_3

       ldd $101A      ;{
       addd #$4394    ;   Programa temporizaci3n para pr3ximo semi-ciclo
       std $101A      ;}
       ldaa #$df
       anda switches
       staa switches

OC3_3  rti

OC4    ldaa #$10      ; {
       staa $1023

       ldaa #$5f
       anda $1000      ;   Activa pulso en OC4
       staa $1000      ; }

OC4_2  ldaa #$50
       deca
       bne OC4_2
       ldaa #$10
       oraa $1000
       staa $1000      ; }
       ldaa #$40
       bita switches
       beq OC4_1

       ldd $101C      ;{
       addd #$4394    ;   Programa temporizaci3n para pr3ximo semi-ciclo
       std $101C      ;}
       ldaa #$bf
       anda switches
       staa switches

OC4_1  rti

IC1    ldaa #$04
       staa $1023
       ldd $1010
       std TCNT1
       ldaa #$30
       oraa $1000
       staa $1000
       ldaa #$60
       oraa switches
       staa switches
       ldd TCNT1
       addd TCNT2
       std $101A
       ldd TCNT1
       addd TCNT3
       std $101C
       inc NUMCIC
       rti

```

## ANEXO 2

(Conjunto de instrucciones del microcontrolador MC68HC11)

Mnemónico	Códigos de operación	Número de ciclos	Mnemónico	Códigos de operación	Número de ciclos
ABA	1B	2	ASR	77	6
ABX	3A	3		57	6
ABY	18 3A	4		18 57	7
ADCA (opr)	89	2	ASRA	47	2
	99	3	ASRB	57	2
	B9	4	BCC (rel)	24	3
	A9	4	BCLR (opr)	15	6
	18 A9	5	(msk)	1D	7
ADCB (opr)	C9	2		18 1D	8
	D9	3	BCS (rel)	25	3
	F9	4	BEQ (rel)	27	3
	E9	4	BGE (rel)	2C	3
	18 E9	5	BGT (rel)	2E	3
ADDA (opr)	8B	2	BHI (rel)	22	3
	9B	3	BHS (rel)	24	3
	BB	4	BITA (opr)	85	2
	AB	4		95	3
	18 AB	5		B5	4
ADDB (opr)	CB	2		A5	4
	DB	3		18 A5	5
	FB	4	BITB (opr)	C5	2
	EB	4		D5	3
	18 EB	5		F5	4
ADDD (opr)	C3	4		E5	4
	D3	5		18 E5	5
	F3	6	BLE (rel)	2F	3
	E3	6	BLO (rel)	25	3
	18 E3	7	BLS (rel)	23	3
ANDA (opr)	84	2	BLT (rel)	2D	3
	94	3	BMI (rel)	2B	3
	B4	4	BNE (rel)	26	3
	A4	4	BPL (rel)	2A	3
	18 A4	5	BRA (rel)	20	3
ANDB (opr)	C4	2	BRCL (opr)	13	6
	D4	3	(msk)	1F	7
	F4	4	(rel)	18 1F	8
	E4	4	BRN (rel)	21	3
	18 E4	5	BRSET (opr)	12	6
ASL (opr)	78	6	(msk)	1E	7
	68	6	(rel)	18 1E	8
	18 68	7	BSET (opr)	14	6
ASLA	48	2	(msk)	1C	7
ASLB	58	2		18 1C	8
ASLD	05	3	BSR (rel)	8D	6

Mnemónico	Códigos de operación	Número de ciclos
BVC (rel)	26	3
BVS (rel)	29	3
CBA	11	2
CLC	0C	2
CLI	0E	2
CLR (opr)	7F	6
	6F	6
	18 6F	7
CLRA	4F	2
CLRB	5F	2
CLV	0A	2
CMPA (opr)	81	2
	91	3
	B1	4
	18 B1	4
CMPB (opr)	C1	2
	D1	3
	F1	4
	18 F1	4
COM (opr)	73	6
	63	6
	18 63	7
COMA	43	2
COMB	53	2
CPD (opr)	1A 83	5
	1A 93	6
	1A B3	7
	1A A3	7
	CD A3	7
CPX (opr)	8C	4
	9C	5
	BC	6
	AC	6
	CD AC	7
CPY (opr)	18 8C	5
	18 9C	6
	18 BC	7
	1A AC	7
	18 AC	7
DAA	19	2
DEC (opr)	7A	6
	6A	6
	18 6A	7
DECA	4A	2
DECB	5A	2
DES	34	3
DEX	09	3
DEY	18 09	4

Mnemónico	Códigos de operación	Número de ciclos
EORA (opr)	88	2
	98	3
	B8	4
	A8	4
	18 A8	5
EORB (opr)	C8	2
	D8	3
	F8	4
	E8	4
	18 E8	5
FDIV	03	41
IDIV	02	41
INC (opr)	7C	6
	6C	6
	18 6C	7
INCA	4C	2
INCB	5C	2
INS	31	3
INX	08	3
INY	18 08	4
JMP (opr)	7E	3
	6E	3
	18 6E	4
JSR (opr)	9D	5
	BD	6
	AD	6
	18 AD	7
LDAA (opr)	86	2
	96	3
	B6	4
	A6	4
	18 A6	5
LDAB (opr)	C6	2
	D6	3
	F6	4
	E6	4
	18 E6	5
LDD (opr)	CC	3
	DC	4
	FC	5
	EC	5
	18 EC	6
LDS (opr)	8E	3
	9E	4
	BE	5
	AE	5
	18 AE	6
LDX (opr)	CE	3
	DE	4
	FE	5

Mnemónico	Códigos de operación	Número de ciclos	Mnemónico	Códigos de operación	Número de ciclos
LDY (opr)	18 CE	4	RORA	46	2
	18 DE	5	RORB	56	2
	18 FE	6	RTI	3B	12
	1A EE	6	RTS	39	5
	18 EE	6	SBA	10	2
LSL (opr)	78	6	SBCA (opr)	82	2
	68	6		92	3
	18 68	7		B2	4
LSLA	48	2		A2	4
LSLB	58	2		18 A2	5
LSLD	05	3	SBCB (opr)	C2	2
LSR (opr)	74	6		D2	3
	64	6		F2	4
	18 64	7		E2	4
LSRA	44	2		18 E2	5
LSRB	54	2	SEC	0D	2
LSRD	04	3	SEI	0F	2
MUL	3D	10	SEV	0B	2
NEG (opr)	70	6	STAA (opr)	97	3
	60	6		B7	4
	18 60	7		A7	4
NEGA	40	2		18 A7	5
NEGB	50	2	STAB (opr)	D7	3
NOP	01	2		F7	4
ORAA (opr)	8A	2		E7	4
	9A	3		18 E7	5
	BA	4	STD (opr)	DD	4
	AA	4		FD	5
	18 AA	5		ED	5
ORAB (opr)	CA	2		18 ED	6
	DA	3	STOP	CF	2
	FA	4	STS (opr)	9F	4
	EA	4		BF	5
	18 EA	5		AF	5
				18 AF	6
PSHA	36	3	STX (opr)	DF	4
PSHB	37	3		FF	5
PSHX	3C	4		EF	5
PSHY	18 3C	5		CD EF	6
PULA	32	4	STY (opr)	18 DF	5
PULB	33	4		18 FF	6
PULX	38	5		1A EF	6
PULY	18 38	6		18 EF	6
ROL (opr)	79	6	SUBA (opr)	80	2
	69	6		90	3
	18 69	7		B0	4
ROLA	49	2		A0	4
ROLB	59	2		18 A0	5
ROR (opr)	76	6	SUBB (opr)	C0	2
	66	6		D0	3
	18 66	7			

Mnemónico	Códigos de operación	Número de ciclos
SUBD (opr)	83	4
	93	5
	B3	6
	A3	6
	18 A3	7
SWI	3F	14
TAB	16	2
TAP	06	2
TBA	17	2
TEST	00	∞
TPA	07	2
TST (opr)	7D	6
	6D	6
	18 6D	7
TSTA	4D	2
TSTB	5D	2
TSX	30	3
TSY	18 30	4
TXS	35	3
TYS	18 35	4
WAI	3E	
XGDX	8F	3
XGDY	18 8F	4

## BIBLIOGRAFIA

---

1. *Electrical And Electronic Instrumentation* ; Hai Hung Chiang; John Willey & Sons; U.S.A 1984.
2. *GE/RCA Optoelectronics* ; GE / RCA; U.S.A. 1987.
3. *Modern Heating, Ventilation And Air Conditioning* ; George Clifford; Prentice Hall; EngleWood Cliffs, New Jersey 1990.
4. *Handbook of Air Conditioning And Refrigeration* ; Shan K. Wang; Mc Graw-Hill; U.S.A. 1993.
5. *Transductores y Acondicionadores de Señal* ; Ramón Pallas Areny; Marcombo; Barcelona, 1989.
6. *Temperature Control* ; Myer Kutz; Robert E. Krieger Publishing; Huntington, New York, 1975.
7. *Temperature Measurement* ; L. Michalski, K. Eckersdorf; John Willey ; England 1991.
8. *Instrumentation And Control Fundamentals And Applications* ;Chester L. Nachtigal; John Willey & Sons; U.S.A. 1990.
9. *"An Introduction to Fuzzy Control"*, D. Driankov, H. Hellendoorn, Springer, Berlin, 1996.
10. *"Industrial Applications of Fuzzy Technology"*, K. Hirota.
11. *"Fuzzy Reasoning in Information, Decision, and Control System"*, Tzafestas, S.G., Kluwer

Academic Publiser, 1994.

12. **"Fuzzy Logic in Knowledge-based systems Decision and Control"**, Gupta Yamakawa, Elsevier Science Publishing Company.
13. **"Fuzzy Set Theory And Its Applications"**, H.J. Zimmerman, Kluwer Academic Publisher.
14. **"Fuzzy Mathematical Techniques with Applications"**, Abraham Kandel, Addison Wesley Publishing
15. **"Fuzzy Sets Uncertainty and Information"**, George J. Klir, Tina A. Folger.
16. **"Introduction to Fuzzy Arithmetic"**, Arnold Kaufman, Madan M. Gupta, Van Nostrand Reinhold Company.
17. **"MC68HC11E9 Technical Manual"**, Motorola, 1993.
18. **"HC11 Reference Manual"**, Motorola, 1993.