

UNIVERSIDAD NACIONAL AUTONOMA  
DE MÉXICO

47  
2ej

FACULTAD DE INGENIERIA

*P* *Macl impresos en ambas*

"DISEÑO Y CONSTRUCCION DE UN COLORIMETR"

**T E S I S**

QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA  
(AREA ELECTRICA - ELECTRONICA)

**P R E S E N T A :**

**M A R C O S J I M E N E Z D A V I L A**

INGENIERO EN COMPUTACION

**P R E S E N T A N :**

**H U G O C E D I L L O L A G U N A**

**E L I S A C O L U M B A M A L D O N A D O S U B I A S**

DIRECTOR DE TESIS: **M I J U A N C A R L O S R O A B E I Z A**

CIUDAD UNIVERSITARIA, MEXICO, D.F.

1995

TESIS CON  
PALLA DE ORIGEN

274280



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.

PAGINACION

DISCONTINUA

## INDICE

AGRADECIMIENTOS	viii
1. INTRODUCCION	1
1.1. COLORIMETRIA.	1
1.1.1.¿COMO DETECTA EL OJO HUMANO LOS COLORES?	3
1.1.1.A.PRINCIPIO DE HUYGENS	6
1.1.1.B.INTERFERENCIAS	24
1.1.1.C.INTENSIDAD EN EL EXPERIMENTO DE YOUNG.	28
1.1.1.D.INTENSIDAD DE LA INTERFERENCIA EN EL EXPERIMENTO DE YOUNG	31
1.1.1.E.OBTENCION DE MAXIMOS Y MÍNIMOS.	31
1.1.1.F.COHERENCIA	36
1.1.1.F.1.COHERENCIA TEMPORAL.	41
1.1.1.F.2.COHERENCIA ESPACIAL.	41
1.1.2.SUMA DE PERTURBACIONES ONDULATORIAS	41
1.1.2.A.CAMBIO DE FASE EN LA REFLEXION	44
1.1.2.B.EL INTERFEROMETRO DE MICHELSON	44
1.2.DIFRACCION	51
1.2.A.DIFRACCION EN UNA SOLA RANURA.	54
1.3. CARACTERISTICAS DEL PROBLEMA	62
III. INTRODUCCION AL DISEÑO DEL HARDWARE	67
III.1.HARDWARE. ARQUITECTURA DEL MICROCONTROLADOR PIC17C44	71
III.1.1.PIC17C44	71
III.1.1.B.ARQUITECTURA DEL PIC17C44	71
III.1.1.3.ESCHEMA DE FLOW/ CICLO DE INSTRUCCIONES	81
III.1.1.4.INSTRUCCIONES DE FLUJO/ENTUBAMIENTO (FLOW/PIPELINE)	84
III.1.1.5.RESTABLECIMIENTO (RESET)	85
III.1.1.5.1.RESTABLECIMIENTO (RESET)	85
III.1.1.5.2.POWER UP TIMER (PWRT)	85
III.1.1.5.3.OSCILLATOR START UP TIMER (OST)	85
III.1.1.5.4.SECUENCIA DE TIME OUT.	88
III.1.1.6.INTERRUCCIONES	94
III.1.1.6.1.REGISTRO DE ESTADO DE INTERRUCCION (INTERRUPT STATUS REGISTER)	94
INTSTA	94
III.1.1.6.2. INTERRUCCIONES PERIFERICAS, HABILITACION DEL REGISTRO	94
PERIPHERAL INTERRUPT ENABLE REGISTER (PIE)	94

III.1.F.3.PERIPHERAL INTERRUPT REQUEST REGISTER (PIR).	100
III.1.F.4.OPERACIÓN DE INTERRUPCIÓN.	101
III.1.F.5.INTERRUPCIÓN RAO/INT.	103
III.1.F.6.INTERRUPCIÓN TMRO.	103
III.1.F.7.INTERRUPCIÓN TOCKI	103
III.1.F.8.INTERRUPCIONES PERIFÉRICAS	104
III.1.F.9. CONTEXTOS QUE SE GUARDAN DURANTE LAS INTERRUPTIONES (CONTEXT SAVING DURING INTERRUPTS).	104
III.1.G.ORGANIZACIÓN DE LA MEMORIA.	105
III.1.G.1.ORGANIZACIÓN DE LA MEMORIA DEL PROGRAMA.	106
III.1.G.2.OPERACIÓN DE LA MEMORIA DEL PROGRAMA.	106
III.1.G.3.ORGANIZACIÓN DE LOS DATOS DE MEMORIA.	112
III.1.G.4.REGISTRO DE PROPOSITO GENERAL (GPR).	114
III.1.G.5.REGISTROS DE FUNCIONES ESPECIALES (SFR).	114
III.1.G.6.REGISTROS DE ESTADOS ALU	115
III.1.G.7.REGISTRO DE ESTADO DEL CPU (CPUSTA).	117
III.1.G.8.REGISTRO TMRO DE ESTADO Y CONTROL (TOSTA).	118
III.1.H.OPERACIÓN DEL STACK.	119
III.1.I.DIRECCIONAMIENTO INDIRECTO.	121
III.1.J.REGISTROS DE DIRECCIONAMIENTO INDIRECTO	122
III.1.K.OPREACIÓN DE DIRECCIONAMIENTO INDIRECTO	123
III.1.L.TABLA DEL PUNTERO (TBLPTRL Y TBLLLPTRH)	124
III.1.M.MODULO DEL CONTADOR DEL PROGRAMA.	125
III.1.N.REGISTRO DE SELECCION DE BANCOS	126
III.1.O.TABLA DE LECTURA Y ESCRITURA	128
III.1.N.MULTIPLICADOR DE HARDWARE.	133
III.1.O.PUERTOS DE ENTRADA/SALIDA.	133
III.1.O.1.REGISTRO DEL PORTA.	135
III.1.O.1.1.UTILIZANDO RA2, RA3 COMO SALIDAS.	135
III.1.O.2.PORTB Y REGISTRO DDRB.	138
III.1.O.3.PORTC Y REGISTROS DEL DDRC.	142
III.1.O.4.PORTD Y REGISTROS DDRD.	144
III.1.O.5.PORTE Y EL REGISTRO DDRE	146
III.1.O.6.CONSIDERACIONES DE PROGRAMACIÓN DE I/O.	149
III.1.P.MODULO DE TRANSMISIÓN RECEPCIÓN UNIVERSAL ASÍNCRONO SÍNCRONO (USART).	150

III.1.F.1.USAPT GENERADOR DE VELOCIDAD DE BAUDS (BRG).	154
III.1.F.2.MODO ASÍNCRONO DEL USART.	158
III.2. ADS901E	161
III.2.A. DESCRIPCIÓN	161
III.2.B. CARACTERÍSTICAS.	162
III.2.C. TEORÍA DE FUNCIONAMIENTO.	163
III.2.D. SALIDA DE DATOS DIGITALES.	165
III.2.E. SEÑAL BALANCEADA Y MODO COMÚN.	166
III.2.E.1.1. CONSIDERACIONES.	166
III.2.F. MANEJO DE LAS ENTRADAS ANALÓGICAS.	167
III.2.F.1.1. MANEJO DE ACOPLAMIENTO DE AC.	167
III.2.F.1.2. CIRCUITO DE INTERFACE DE ACOPLAMIENTO DE DC.	169
III.2.F.1.3. REFERENCIA EXTERNA	170
III.2.F.1.4. SOLUCIÓN DE LA REFERENCIA DE MANERA ECONÓMICA.	171
III.2.F.1.5. SOLUCIÓN A LA REFERENCIA EXACTA	172
III.2.A. ENTRADA DE RELOJ	173
III.2.B. SEÑALES DIGITALES.	174
III.2.C. MODO DE BAJO CONSUMO.	175
III.2.D. DESACOPLOAMIENTO Y CONEXIONES A TIERRA	176
III.2.D.1.1. CONSIDERACIONES.	176
III.2.E. DIVISOR DE VOLTAJE.	178
III.3. CONTADORES	182
III.3.A. OPERACIÓN DE CONTADORES ASÍNCRONOS.	182
III.4 MEMORIAS	187
III.4.A. ARREGLO BÁSICO DE MEMORIA SEMICONDUCTORA.	187
III.4.B. DIRECCIÓN Y CAPACIDAD DE MEMORIA.	189
III.4.C. OPERACIONES BÁSICAS DE LA MEMORIA.	190
III.4.C. OPERACIÓN ESCRITURA.	191
III.4.E. OPERACIÓN DE LECTURA.	192
III.4.F. RAM	193
III.5. COMPARADOR.	198
III.5.A. COMPARADOR CON HISTÉRESIS.	199
III.6 REGIONES DE CORTE Y SATURACIÓN DEL TRANSISTOR.	206
III.7. CIRCUITO INTEGRADO 74F126.	212
III.8.LA WWW	214
III.9.HTML Y EL AMPLIO MUNDO WEB.	215

III.10.INTERNET.	215
III.10.A.DIBUJOS DE BROWSER Y EL LENGUAJE HTML.	217
III.10.B.CÓMO SE CONECTA INTERNET.	219
III.10.C.CLIENTES, SERVIDORES Y BROWSERS.	221
III.10.D.EL FLUJO DE INFORMACIÓN.	222
III.11.BENEFICIOS DE LA WORLD WIDE WEB.	223
III.12.¿QUE ES HTML?	224
III.12.A.ESTÁNDARES DE HTML Y EXTENSIONES.	224
III.13.EL CONSORCIO DE WWW.	225
III.14.LAS FUERZAS DE INGENIERÍA EN INTERNET.	226
III.15.CARACTERÍSTICAS DE HTML.	226
III.15.A.LIMITACIONES ESPECÍFICAS DE HTML.	229
III.15.B.PRO Y CONTRA DE LAS VERSIONES.	231
III.15.C.EVITANDO LAS EXTENSIONES.	231
III.15.D.HERRAMIENTAS PARA DISEÑAR FORMAS HTML.	231
III.15.E.PROCESADOR DE PALABRAS O EDITOR HTML.	232
III.16.¿QUÉ ES UNA BASE DE DATOS?	233
III.16.A.STRUCTURED QUERY LANGUAGE (SQL).	234
III.16.A.1.TRAER DATOS (SELECT).	234
III.16.A.2.SELECCIONAR REGISTROS ESPECÍFICOS (WHERE).	235
III.16.B.ORDENAMIENTO DE LAS CONSULTAS (ORDER BY).	236
III.17. PLATAFORMA DE NETSCAPE Y JAVASCRIPT.	237
III.18. EL CLIENT-SIDE DE JAVASCRIPT.	239
III.19. JAVASCRIPT EN EL LADO DEL SERVIDOR.	241
III.20. ARQUITECTURA DE APLICACIONES JAVASCRIPT.	243
III.20.A. SERVIDORES DE BASE DE DATOS	245
III.21. CONSTRUCCION DE UNA APLICACIÓN JAVASCRIPT.	248
III.22. RESUMEN DE RUNTIME PROCESSING.	251
IV RESUMEN Y CONCLUSIONES.	253
IV.1.A.ESPECIFICACIONES	253
IV.1.A.1.CAMBIO DE NIBBLES	255
IV.1.A.2.CICLO DE CONTROL	256
IV.1.B.INTERPRETACIÓN DE LOS DATOS A TRAVES DEL SOFTWARE.	267
IV.1.B.1.CARACTERIZACION.	269
IV.1.B.2.MUESTREO.	271
IV.1 CONCLUSIONES	273

## BIBLIOGRAFIA

APENDICE A COMO UTILIZAR HTML.

APENDICE B MANUAL DE USUARIO PARA EL COLORIMETRO

APENDICE C FUENTES DE VOLTAJE

APENDICE D CARACTERISTICAS Y DETALLES DE LA TARJETA TAD Y LA CAMARA DE VIDEO

APENDICE E LISTADO DE PROGRAMAS QUE UTILIZA EL COLORIMETRO

## **AGRADECIMIENTOS**

En este apartado mencionaré a varias instituciones y personas que hicieron posible la realización del proyecto de investigación presentado en la tesis.

Las ideas de las cuales partió el proyecto fueron propuestas por los M.I. Juan Carlos Roa y Luis E. Bustamante Murillo. La síntesis de sus ideas surgió en el Centro de Instrumentos de la UNAM, específicamente, dentro de lo que fue el Proyecto Universitario de Investigación y Desarrollo Espacial (PUIDE). Mi participación en el PUIDE como becario me dio la oportunidad de aprender y disfrutar de la investigación en tecnología de punta.

El MI Juan Carlos Roa, mi asesor de tesis, revisó y corrigió con todo cuidado cada capítulo. La tesis también fue leída por los ingenieros Lucila Patricia Arellano, Ernesto Suarez Sport, Manuel Manriquez Miranda y, así como el M.I. Adolfo Millán Nájera, quienes hicieron comentarios de gran utilidad en la fase final del proyecto.

Asimismo, quisiera mencionar a las siguientes personas. Al M.I. Luis E. Bustamante Murillo y al Ing. Saúl De la Rosa Nieves, por compartir conmigo sus conocimientos de electrónica digital y analógica. A Juan Carlos Vidal Serrato por haber realizado el esquemático de la tarjeta de adquisición de datos. Al señor José Luis Montoya, por su colaboración en el desarrollo del proyecto. Un agradecimiento muy especial a la Lic. Patricia Díaz Herrera, por su valiosa colaboración en la redacción y corrección de estilo en el texto de esta tesis. Por su apoyo y cariño, a, Rufina Dávila Ramirez, Honorio Jiménez Villalobos, Raúl Álvarez, Georgina y Olivia Jimenez Dávila. Y finalmente por su confianza, a Francisco Díaz B., Amparo Herrera E. y Angélica Díaz H.

**Marcos Jiménez Dávila**

## **A LA UNAM**

Por habernos dado la oportunidad de estudiar gratis  
Por dar la oportunidad a todos los estudiantes como nosotros de estudiar  
una carrera universitaria que no podríamos haber pagado en una institución  
privada, dado que en el tiempo que estudiamos no teníamos recursos. Así  
como a gran número de instalaciones culturales, deportivas, etc., de las  
cuales hicimos uso al cien por ciento.

## **A LOS ALUMNOS**

Terminar una carrera no es todo, se tiene que seguir de cerca los  
progresos tecnológicos, dado que la educación que se da en la UNAM es  
formal. Solo así tendremos la capacidad de enfrentarnos a los avances en  
tecnología. Además DEBEMOS difundir nuestras ideas.

## **A NUESTROS PADRES**

Por habernos dado su apoyo para estudiar, pues ellos no tuvieron esa  
oportunidad y, sin embargo, no nos la negaron. Esto fue posible gracias a  
la gratitud de la UNAM, ya que, de otra forma, no habríamos estudiado.

## **A LOS MAESTROS**

Por su ardua labor, ya que no se hacen ricos por dar clases, están  
contentos y motivados por los alumnos que salen adelante. Dar clases es  
difícil, porque no solo es dar el tiempo de clase, sino la preparación de  
la misma.

## **A LOS ALUMNOS FÓSILES**

Si llegan a leer esto, les recomendamos que estudien y se pongan las pilas,  
porque la competencia y competitividad son palabras muy conocidas por  
todos. En los concursos se queda el mejor, además de que atrás de ustedes  
viene más gente que quiere estudiar y esta es una cadena que viene, se va y

no se queda. Así como hay tiempo para nacer, morir, trabajar y divertirse, también hay tiempo para estudiar, no dejen pasar esta oportunidad.

Un ejemplo es la hierba que crece en cualquier lugar en forma desmesurada, pero cuando llega un ventarrón se la lleva y se muere porque no tiene raíces bien fundamentadas. Tu puedes, compañero, si así lo quieres.

**Elisa Columba Maldonado Subías**

**Hugo Cadillo Laguna**

## I INTRODUCCION

### I.1.A. COLORIMETRIA

La colorimetría está resolviendo en la actualidad, adhiriendo grado relativo de actividad, tanto en el campo de la investigación científica como en el de las aplicaciones industriales. Durante el periodo de los últimos diez años, los sistemas de especificaciones del color, desarrollados por distintos autores y adoptados por la Comisión Internacional de Iluminación (CIE), están capaces de describir detalladamente, desde distintos aspectos, de la Colorimetría, los problemas que el problema principal de la industria de diseño de un sistema de color, consecuencia de los cambios extendidos en el campo de actividad, se ha abierto una nueva etapa caracterizada por la aparición de nuevos problemas y temas de investigación que se pueden agrupar en los apartados siguientes:

#### • METODOS DE MEDIDA

El método de medida del color existe en el grado de una línea de acción, en la que se trata de correlacionar los datos de color con el problema de la reproducibilidad de los resultados de distintos instrumentos de medida, a caracterizar, en sí misma, que se ha en la misma condición de un tipo de color con un tipo de relación de color medida por distintas condiciones observadas.

- **MODELOS DE APARIENCIA DEL COLOR**

Los sistemas de especificación del color adoptados por la C.I.E. presentan fallos en relación con la Apariencia del color. Hay casos en que las muestras que tienen las mismas coordenadas cromáticas no aparecen como iguales a un observador, sin que ello esté relacionado con las diferencias propias entre el observador concreto escogido y el observador patrón definido por la C.I.E.. Al problema que presentan estas experiencias se han ido desarrollando distintos sistemas de apariencia del color, que han de ser comprobados en la práctica.

- **REDEFINICIÓN DE LOS OBSERVADORES PATRONES**

Los observadores definidos por C.I.E. como patrones tienen algunas deficiencias en algunas zonas espectrales y están siendo estudiados detenidamente y con detenimiento hoy en día por distintos grupos de investigación.

- **APLICACIONES INDUSTRIALES**

La medida del color está siendo cada vez más usada en la industria como herramienta de control de calidad, puesto que la medida del color se hace con métodos no invasivos. Por citar algunos ejemplos, el color se usa como control de maduración de alimentos, como característica de definición para "denominaciones de origen" de distintos tipos de alimentos, o como definitorio de la calidad de un producto.

## 1.2. ¿CÓMO DETECTA EL OJO HUMANO LOS COLORES?

Aunque el medio ambiente se nos presenta en color, en realidad el mundo es incoloro. El mundo visible se compone de materia incolora y de ondas electromagnéticas igualmente incoloras que únicamente se distinguen por su longitud de onda.

Generalmente la materia no es visible directamente. Solamente la vemos directamente en casos excepcionales, cuando emite radiaciones visibles, es decir, cuando esta incandescente o por sí misma irradia de otra manera. De otra forma, solamente la percibimos cuando refleja en nuestra retina rayos electromagnéticos visibles.

Esta luz puede llegar a nuestros ojos desde una fuente de radiación directa, pero también indirectamente y esto es lo que ocurre al ser reflejada o traspasada por una materia.

Como ya se mencionó las ondas electromagnéticas se distinguen solamente por su longitud de onda y por su energía, pero no por su color. La percepción del color se produce solamente en el cerebro del individuo.

Para el observador la materia se aparece en sus variadas formas de colores diferentes, porque según sea su constitución molecular, absorbe o refleja respectivamente de forma distinta la luz incidente.

Una parte de los rayos de luz blanca que caen sobre un objeto multicolor opaco son absorbidos por dicho objeto mientras el resto es reflejado y puede caer en el ojo del observador. Si se

trata de materia transparente, el proceso es análogo. La parte de luz no absorbida atraviesa la materia produciéndose el efecto de colorido.

La rosa roja y el tallo verde de una rosa, por ejemplo, se diferencian entre sí por la diferente constitución molecular de la estructura de los tejidos. Así sucede que la flor absorbe la zona verde de la luz blanca del sol, sustrae de la luz una parte de la energía y remite solamente la parte de la radiación utilizada que aparece roja al observador. Con el tallo sucede lo inverso, absorbe de la luz la parte roja y remite la parte visible de la radiación que da la impresión de color verde al observador.

Cuando no existen radiaciones electromagnéticas visibles, no puede haber colorido. En una habitación completamente oscura, una alfombra azul es tan invisible como un tomate rojo o un ratón blanco. Únicamente tiene la facultad de parecer azul con una determinada luz. Naturalmente esta facultad también la tiene cuando no existe la luz, pero entonces no se percibe. La luz que hace óptimamente visible esta facultad de la materia, la que parece color, debe tener igual intensidad que todas las ondas electromagnéticas visibles.

La materia de idéntica estructura atómica se muestra idéntica con la misma iluminación. Si cambia la composición espectral de la luz, también cambia entonces el aspecto de la materia, por lo que una tela de abrigo parece diferente a la luz de una lámpara que a la luz del día.

Por consiguiente el aspecto colorido es siempre algo relativo. No solamente depende de la propiedad que tenga la materia de

absorber la luz que cae, de remitirla o de traspasarla, sino que también es decisiva la caída, la composición espectral de la luz existente.

El proceso visual tiene lugar en un transcurso de tiempo. En el proceso visual el órgano recode también las ondas que ha emitido una determinada fuente. Estas ondas se transforman en una percepción. El transcurso de tiempo durante el proceso visual puede ser percibido por el hombre debido a la velocidad tan rápida de la luz. En espacio sin aire la luz se mueve a una velocidad de 300000 km por segundo. Por consiguiente, la luz podría girar alrededor del globo terráqueo siete veces y media en un solo segundo.

El ojo es un sistema óptico. Tiene la misión de dirigir radiaciones visibles a la membrana reticular. Esta recode el estímulo físico y lo transforma en una excitación fisiológica. La excitación es dirigida por cada uno de los receptores de la membrana reticular por medio de fibras nerviosas al medio óptico y por este al cerebro. En la región cerebral calcarina que está conectada a la corteza cerebral, esta excitación se transforma en una sensación, esta a su vez es la visión consciente.

La figura 1.1, muestra un corte esquemático del ojo humano. La esclerótica (A) envuelve el globo ocular. En la esclerótica, delante del sistema óptico del ojo, se encuentra una córnea transparente (B), que elimina una parte de las ondas electromagnéticas invisibles, dejando pasar solamente ondas de una magnitud que oscila aproximadamente de 380 a 750 nanómetros. La córnea además actúa por su curvatura como lente convexa. Seguidamente el rayo de luz tiene que atravesar una capa de líquido llamado humor acuoso (C), que se encuentra entre la

cornea y el cristalino (D). Finalmente, los rayos visibles llegan al (humor vítreo) (E) del ojo y caen sobre la retina (F). El iris y la pupila, colocados delante del cristalino reaccionan con la intensidad de luz, es decir, regula la cantidad de luz que penetra al ojo. El cristalino del ojo deja pasar las ondas electromagnéticas de longitud entre 380 nm y 760 nm. Gracias a su curvatura variable es capaz de proyectar en la retina con toda exactitud objetos que estén a diferentes distancias uno detrás del otro. La retina está formada por una serie de capas. La luz atraviesa primeramente una capa de fibras nerviosas y unas cuantas capas de células. Pero solamente cuando, debajo de aquellas, llega a la capa de bastones y conos, es cuando la energía física de las ondas electromagnéticas se transforma en un estímulo fisiológico.

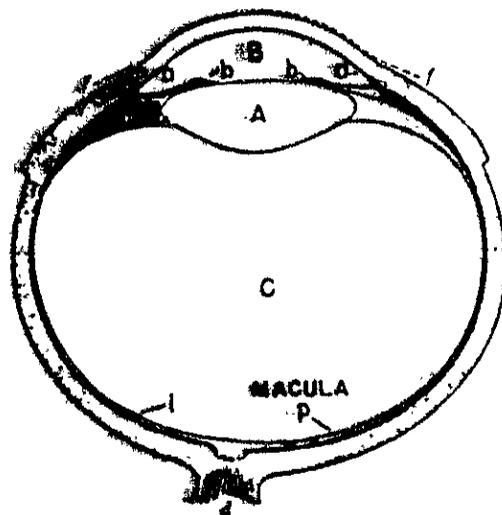


Figura 1. 1 Corte esquemático del ojo.

Los bastones y conos están repartidos de forma irregular sobre la retina. En la región central de la retina, en la zona que está frente al cristalino, dominan los conos. En el resto de la retina los conos y bastones están colocados uno al lado del otro. Por último, en los bordes de la retina dominan los bastones.

Mientras que los bastones están unidos por células dardadas en grupos o fibras nerviosas, cada cono tiene su propia unión con una fibra nerviosa. Todas las fibras nerviosas se reúnen en la papila del nervio óptico, a la que también se le llama (punto ciego). De allí parten en haz hacia el cerebro, estando cada fibra nerviosa al lado de la demás. En la zona cega cada una puede transmitir información al cerebro por separado.

El exacto funcionamiento de los conos y bastones no ha sido aún explicado científicamente. No puede establecerse con seguridad cómo se produce realmente la visión en color.

La manera en que funcionan los bastones se conoce con un poco más exactitud. Estos perciben únicamente la diferencia de claridad, tienen su mayor importancia cuando existe poca iluminación. La diferencia de colores no puede percibirse con ayuda de los bastones. Esto explica el porque en la noche, con la débil luz de la luna, no se perciben los diferentes colores, pero con la escasa luz existente solo funcionan los bastones.

Los bastones contienen un líquido llamado (purpura visual). Esta sustancia consiste en un pigmento de carotina unida con proteínas que ha entrado en combinación con la vitamina A y que con la luz sufre alteraciones químicas. Este cambio se efectúa pasando de naranja y del amarillo óptico al blanco óptico, liberándose durante este proceso energías que producen estímulo.

fisiológicos. En cuanto cesa el estímulo de la luz se regenera la sustancia visual volviendo a su estado primitivo.

Del funcionamiento de los conos, de los que se encuentran hasta 15000 en un milímetro cúbico, solo puede hablarse con poco fundamento.

La teoría más extendida se basa sobre el hecho de que existen tres tipos de conos diferentes con sensibilidades diferentes para tres zonas espectrales (zona azul - violeta, verde y rojo - naranja).

Por lo tanto el proceso de la visión del color se supone del siguiente modo: Al ojo llega un estímulo de color que estimula a los distintos tipos de conos de manera diferente según sea su construcción espectral. Así pues las células nerviosas transmiten al cerebro valores diferentes. La sensación del color de los diferentes matices se produciría por consiguiente de manera muy similar a como sucede en la mezcla aditiva de colores primarios y secundarios. Así, pues, la visión se fundamentaría en el mismo principio que en la televisión en color hace posible la producción de colores compuestos. Estas semejanzas son claras y lógicas.

Existe otra teoría que supone un solo tipo de conos que funcionarían todos por igual, es decir, de forma similar a la que se describe en la (visión - bastones). La diferencia residiría en que todos los conos son sensibles a variaciones de longitudes de onda y de intensidad, mientras los bastones solo pueden registrar las diferencias de intensidad.

Aunque no se haya logrado una prueba científica contundente de la teoría de los tres tipos diferentes de conos, es la teoría más aceptada del funcionamiento del ojo humano.

### I.2.A. PRINCIPIO DE HUYGENS.

Importantes dificultades aparecen al intentar calcular rigurosamente la propagación de la luz en medios no homogéneos, en un medio que es parcialmente obstruido por cuerpos opacos. Sin embargo, en la mayoría de los casos de importancia práctica es posible encontrar una respuesta aproximada mediante el uso de métodos que requieren solamente algunas suposiciones de carácter general acerca de la naturaleza de las ondas luminosas. Tales métodos están basados en el principio de Huygens.

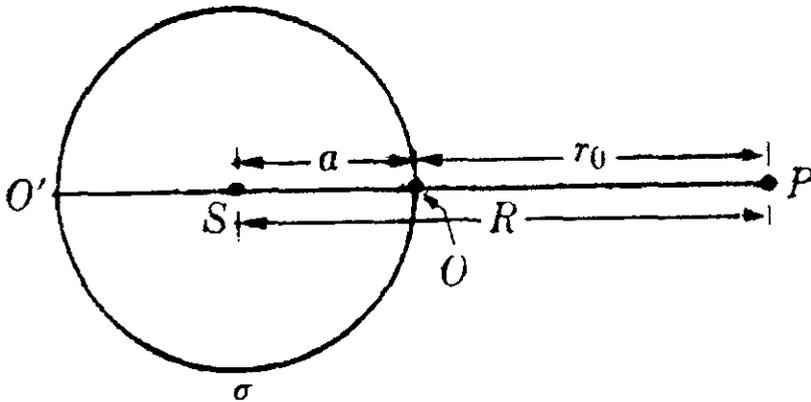


Figura 1. 2 Superficie auxiliar para la aplicación del principio de Huygens.

El principio de Huygens tuvo origen en el conocimiento general de que las ondas se propagan gradualmente, de punto a punto de un medio. Por consiguiente, si una fuente  $S$  es rodeada por una superficie cerrada  $\sigma$  (Fig. 1.2), la perturbación originada en  $S$  podrá alcanzar la región del espacio exterior a  $\sigma$  solamente atravesando dicha superficie. Es entonces natural considerar a la perturbación en la región exterior como causada por la perturbación en la superficie  $\sigma$ ; esto es, suponer que los diferentes puntos  $\sigma$ , cuando son alcanzados por la onda, se convierten en el origen de ondas secundarias y que la perturbación observada más allá de la superficie  $\sigma$  resulta de la superposición de estas ondas secundarias. Este es el enunciado del principio de Huygens en su forma más general.

Resta por demostrar que podemos reconstruir la onda conocida más allá de  $\sigma$ , combinando los efectos de un conjunto apropiado de ondas secundarias que emanan de los distintos elementos de  $\sigma$ . La prueba general del principio de Huygens fue dada por Kirchhoff, es conocida como el teorema de Kirchhoff.

Para tal análisis, utilizaremos las ecuaciones de ondas esféricas que dan  $s$  (desplazamiento) y  $p$  (cambio de presión) en función de  $r$  y  $t$ :

$$p = \frac{F(t-r/v)}{r} \dots\dots\dots \text{Ec. 1.1}$$

$$s = \frac{f(t-r/v)}{r} \dots\dots\dots \text{Ec. 1.2}$$

donde  $r$  es la distancia a la superficie  $\sigma$  a la velocidad de propagación.

Considerando ahora una fuente asimétrica,  $p$  y  $\rho$  serán funciones no solamente de  $r$ , sino también de los ángulos que determinan la orientación de la recta que une a  $O$  con el punto de emisión. Puede demostrarse que si estas funciones varían poco con dicha orientación, ellas pueden escribirse como productos de un factor  $g$  que depende solamente de la orientación. Por ejemplo tendríamos una expresión de la forma

$$p = \frac{f(\theta - r/v)}{r} g \dots \dots \dots \text{Ec. 1.3}$$

Como superficie auxiliar  $\sigma$  tomamos una esfera de radio  $a$  y centro en la fuente  $S$  (Figura 1.1.). Sea un punto  $P$  a una distancia arbitraria  $R > a$  de  $S$ ,  $O$  y  $O'$  los puntos de intersección de la recta  $SP$  con la superficie  $\sigma$  y  $r = O'P = R - a$  la distancia mínima de  $P$  a  $\sigma$ . Suponemos que  $S$  es la fuente de una onda esférica, a lo que consideraremos específicamente por una onda luminosa.

Sea  $E, F, t$  la perturbación óptica al tiempo  $t$  en el punto  $P$ . En un instante dado,  $F$  tiene el mismo valor en todos los puntos de la esfera  $\sigma$ , de manera que podemos escribir:

$$E(\mathbf{a}, t) = f(t) \dots \dots \dots \text{Ec. 1.4}$$

donde  $f$  es una función dada, solamente del tiempo. La expresión general para  $E(\mathbf{a}, t)$  en una onda esférica es de la forma

$$E(R,t) = \frac{\varphi(t - R/v)}{R} \quad \text{Ec. 1. 4}$$

.....

Cuando  $R = \alpha$ , resulta

$$E(\alpha,t) = \frac{\varphi(t - \alpha/v)}{\alpha} \quad \text{.....Ec. 1. 6}$$

y con la ecuación 1.1 obtenemos:

$$\varphi\left(t - \frac{\alpha}{v}\right) = \alpha f(t) \quad \text{.....Ec. 1. 7}$$

La ecuación anterior vale para todo tiempo  $t$ . Podemos entonces reemplazar en ambos lados  $t$  por  $(t - r_0/v)$  y resulta ahora

$$\varphi\left(t - \frac{r_0}{v} - \frac{\alpha}{v}\right) = \alpha f\left(t - \frac{r_0}{v}\right) \quad \text{.....Ec. 1. 8}$$

o (como  $r_0 + \alpha = R$ )

$$\varphi\left(t - \frac{R}{v}\right) = \alpha f\left(t - \frac{r_0}{v}\right) \quad \text{.....Ec. 1. 9}$$

La ecuación 1.4 queda entonces:

$$E(r,t) = E(a+r,t) = \frac{a}{a+r} f\left(t - \frac{r}{v}\right) \dots \dots \dots \text{Eq. 1.1}$$

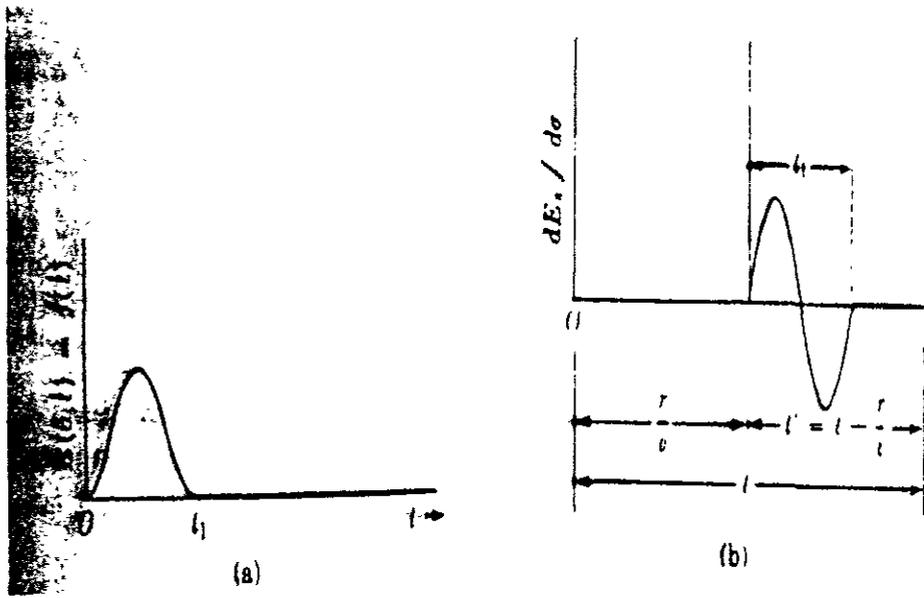


Figura 1.1. Perturbacion primaria en el tiempo y perturbacion secundaria desde una zona infinitesimal.

De acuerdo al principio de Huygen, suponemos que cada elemento  $d\sigma$  de la superficie  $\sigma$  emite una onda secundaria durante el intervalo de tiempo que transcurre entre  $t$  y  $t+dt$ . Sea  $dE_s$  la perturbacion producida por esta onda secundaria en el punto  $P$ . Es natural suponer que  $dE_s$  es proporcional a  $d\sigma$ . Como razon para que la perturbacion que emana del area (en contraste con la que ocurre en un punto) tenga la misma intensidad en todas direcciones, consideraremos a  $dE_s$  como una funcion del tiempo.

las coordenadas, con una ecuación similar a la ecuación I.4, que describe una onda esférica producida por una fuente asimétrica. Sin embargo, supondremos que el factor direccional  $q$  que aparece en esta ecuación, depende solamente del ángulo  $\theta$  entre la perpendicular al elemento de área  $d\sigma$  y la recta que une a este elemento con  $P$  (Fig. I.5). Entonces, siendo  $r$  la distancia de  $d\sigma$  a  $P$ , probamos para  $dE$ , una expresión de la siguiente forma:

$$dE_s = \frac{\psi(t-r/v)}{r} q(\theta) d\sigma \quad \dots\dots\text{Ec. I. 11}$$

Nos referimos a  $q(\theta)$  como el factor de oblicuidad. Sin ninguna pérdida de generalidad, podemos definir a  $q$  de manera que sea igual a la unidad en la dirección de  $S$  a  $P$ ; o sea,

$$q(\theta) = 1 \quad \dots\dots\dots\text{Ec. I. 12}$$

Debemos probar ahora que es realmente posible determinar las funciones  $\psi$  y  $q$  de forma tal que la superposición de las perturbaciones secundarias descritas por la ecuación I.11 reproduzca la perturbación representada por la ecuación I.10, esto es,

$$\int_{\sigma} dE_s = E(\alpha + r_0, t) \quad \dots\dots\dots\text{Ec. I. 13}$$

Con este fin, comenzamos señalando que si  $r$  es la distancia de  $P$  a un punto  $Q$  de la superficie  $\sigma$ , y  $\beta$  es el ángulo entre  $SQ$  y  $SP$  (Fig. I.3), se cumple la siguiente relación:

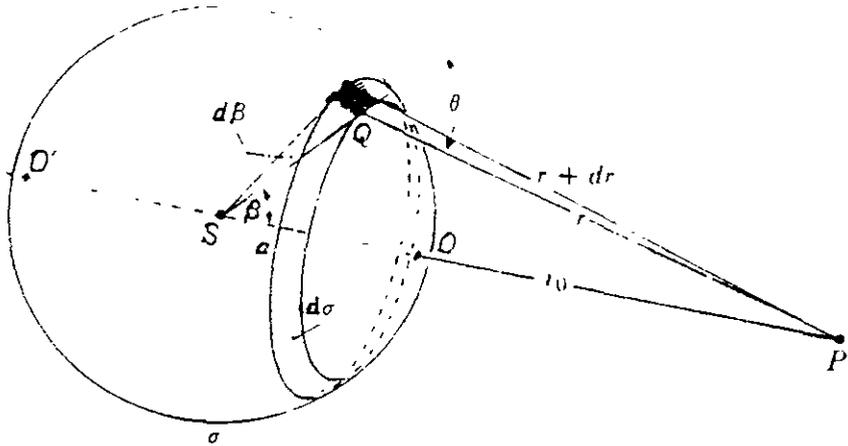


Figura 1.4 Construcción de una zona elemental en la superficie de Huygens.

$$r^2 = a^2 + (\alpha + r_0)^2 - 2a(\alpha + r_0)\cos\beta \dots\dots\dots \text{Eq. 1.14}$$

La diferenciación de esta ecuación nos da:

$$rdr = a(\alpha + r_0)\text{sen}\beta d\beta \dots\dots\dots \text{Eq. 1.15}$$

Consideremos ahora la zona circular determinada por la intersección de la superficie  $\sigma$  con dos conos de semiapertura  $\beta$  y  $\beta + d\beta$ ; sean además  $r$  y  $r + dr$  las distancias de  $S$  a los dos círculos determinados por las intersecciones (fig. 1.4). El área de esta zona es

$$d\sigma = (2\pi\alpha \sin\beta) \cdot (\alpha d\beta) = 2\pi\alpha^2 \sin\beta d\beta = 2\pi \frac{\alpha}{\alpha + r_0} r dr \dots\dots\dots \text{Ec. 1. 16}$$

Desechando cantidades infinitesimales, la distancia  $r$  y el ángulo  $\theta$  son constantes en la zona infinitesimal anteriormente delimitada. Ahora estamos en condiciones de calcular la contribución de esta zona a la perturbación observada en  $P$  simplemente sustituyendo el área de la zona (como fue dada por la ec. 1.16) por la cantidad  $d\sigma$  que aparece en la ecuación 1.12. Obtenemos

$$dE_1 = 2\pi \frac{\alpha}{\alpha + r_0} \psi\left(t - \frac{r}{v}\right) q(\theta) dr \dots\dots\dots \text{Ec. 1. 17}$$

La siguiente etapa es sumar las contribuciones de las diferentes zonas infinitesimales tales como la considerada anteriormente. Como ya se mencionó, la emisión de ondas secundarias comienza en  $t=0$  y termina en  $t=t_1$ ; matemáticamente, esto significa que  $\psi$  es cero, excepto para  $0 < t - r/v < t_1$ . Como las ondas viajan a una velocidad finita  $v$ , y además las distintas regiones de  $\sigma$  están a diferentes distancias de  $P$ , este punto, en un instante dado, recibirá ondas secundarias solamente de una porción limitada de  $\sigma$ , a la cual denominaremos región activa. En la suma de contribuciones de las diferentes zonas que forman una región activa, consideraremos cinco periodos de tiempo:

• **Primer periodo.** Consideremos un instante de tiempo  $t < r/v$ . En este instante, ni la onda primaria ni las ondas secundarias originadas en  $\sigma$  han alcanzado al punto  $P$ . La ecuación (1.11) representa entonces una identidad trivial, porque ambos miembros son idénticamente nulos.

• **Segundo periodo.** Consideremos el intervalo de tiempo  $r/v < t < r/v + 1$ , durante el cual la onda primaria pasa por el punto  $P$ . En este intervalo,  $P$  recibe ondas secundarias de la porción de esfera que se halla en la vecindad de la región activa definida anteriormente; vease Fig. 1.1. Las ondas emitidas en un círculo a una distancia  $r$  determinadas por la ecuación

$$r = v(t - r/v) \quad \text{Ec. 1.12}$$

Expresando la perturbación original como la suma de perturbaciones producidas en las zonas infinitesimales incluidas en la región activa, obtenemos

$$k(a+r, t) = 2\pi \frac{a}{a+r} \int_0^{\theta} q(\theta) \psi \left( t - \frac{r}{v} \right) dr \quad \text{Ec. 1.13}$$

Debido a que el valor de  $r$  es pequeño (vease las desigualdades 1.1), la región activa es una porción muy pequeña de la superficie esférica  $\sigma$ . En consecuencia, el ángulo  $\theta$  de emisión de las ondas secundarias es próximo a cero para todos los puntos de la región activa. Como el factor de oblicuidad es una función que varía poco con  $\theta$  y vale uno en  $\theta = 0$ , podemos poner  $q(\theta) = q(0)$  en la ecuación 1.13.

Cambiamos ahora la variable de integración de  $r$  a  $t'$  poniendo

$$t' = t - \frac{r}{v} \dots \dots \dots \text{Ec. 1. 20}$$

Notemos que aquí  $t$  es el tiempo (fijado) en que las ondas secundarias alcanzan el punto  $P$ , y  $t'$  es el tiempo (variable) en que esta onda deja el punto de la superficie  $\sigma$  que se halla a una distancia  $r$  de  $P$  (véase Fig. 1.4). Sabiendo que  $dr = -v dt'$ , que  $r' = 0$  para  $r = r$  y que  $t' = t - r_0/v$  para  $r = r_0$ , obtenemos de la ec. 1.18 y la ec. 1.19:

$$\frac{\alpha}{\alpha + r_0} f\left(t - \frac{r}{v}\right) = E(\alpha + r_0, t) \dots \dots \dots \text{Ec. 1. 21}$$

$$\begin{aligned} &= 2\pi \frac{\alpha}{\alpha + r_0} \int_{t-r_0/v}^t \psi(t') (-v dt') \\ &= 2\pi \frac{\alpha}{\alpha + r_0} v \int_0^{t-r_0/v} \psi(t') dt' \dots \dots \dots \text{Ec. 1. 22} \end{aligned}$$

o sea,

$$f\left(t - \frac{r}{v}\right) = 2\pi v \int_0^{t-r_0/v} \psi(t') dt' \dots \dots \dots \text{Ec. 1. 23}$$

Como  $f$  es conocida, esta ecuación determina unívocamente la función  $\psi$ . Asimismo, diferenciando con respecto a  $t$  nos da

$$\psi\left(t - \frac{r}{v}\right) = \frac{1}{2\pi v} f'\left(t - \frac{r}{v}\right) \dots \dots \dots \text{Ec. 1. 24}$$

ondas nemos indicado con  $f$  la derivada de una integral con respecto a su argumento al cual nemos escrito con  $t$  en lugar de  $t_1$ .

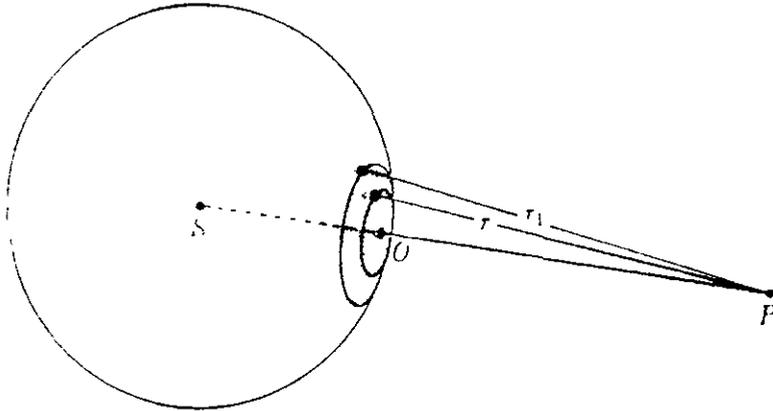


Figura 1. La region activa para  $r/v < r/v_1$

Después de esto, se puede concluir que en el intervalo de tiempo  $r/v < r/v_1$ , es posible ver a la perturbacion original con la resultante de ondas secundaria originadas en los diferentes elementos de la superficie exterior. Esta onda secundaria estan descritas por

$$dE_s = \frac{1}{2\pi v} \int \left(1 - \frac{r}{v}\right) |q(\theta)| d\sigma \dots \dots \dots \text{Eq. 1. 2'}$$

o por la ecuacion equivalente

$$dE_s = \frac{1}{2\pi r} \left( \frac{\partial E(\alpha, t)}{\partial t} \right)_{r=r_0} q(\theta) d\sigma \dots \dots \dots \text{Ec. I. 26}$$

• **Tercer periodo.**

Durante el intervalo de tiempo siguiente

$$r_0/v + t_1 < t < (r_0 + 2\alpha)/v \dots \dots \dots \text{Ec. I. 27}$$

la region activa de la superficie  $\sigma$  es una zona cuyos limites interiores y exteriores son circulos definidos respectivamente por:

$$r_1 = w.$$

$$r_2 = v(t - t_1) \dots \dots \dots \text{Ec. I. 28}$$

De acuerdo al principio de Huygens deberia ser posible representar la perturbacion en P por una ecuacion de la siguiente forma:

$$E(r_0 + \alpha, t) = 2\pi \frac{\alpha}{\alpha + r_0} \int_{r_1}^{r_2} q(\theta) \psi \left( t - \frac{r}{v} \right) dr \dots \dots \dots \text{Ec. I. 29}$$

Por otra parte, sabemos que para todo tiempo mayor que  $r_0/v + t_1$ , la perturbacion en P es nula (o sea,  $E(r_0 + \alpha, t) = 0$  para  $t > r_0/v + t_1$ ). Entonces, si el principio de Huygens es válido, la integral del miembro derecho de la ec. I.29 debe anularse.

- **Cuarto periodo.** La perturbación secundaria proveniente del antipolo  $P'$  llega a  $P$  al tiempo  $(r_0+2a)/v$ , y dura hasta  $(r_0+2a)/v+t$ . Por consiguiente, en el intervalo de tiempo  $(r_0+2a)/v < t < (r_0+2a)/v+t$ , la región activa incluye al punto  $P'$ ; el borde de esta región es el círculo definido por

$$r = v(t - t_0) \dots \dots \dots \text{Eq. 1. 31}$$

El procedimiento similar al utilizado en los casos anteriores aplicando el principio de Huygens, nos da que el miembro derecho de la ecuación resultante debe anularse, esto es

$$E(r+a, t) = 0 \dots \dots \dots \text{Eq. 1. 31}$$

- **Quinto periodo.** Para  $t > (r_0+2a)/v+t$ , el punto  $P$  ya no recibe ondas secundarias de ninguna porción de la esfera  $\sigma$ . La ecuación 1.14 que expresa el principio de Huygens entonces resulta una identidad trivial, porque ambos miembros se idénticamente nulos.

hemos probado así que la ecuación 1.14 se satisface para todo instante  $t$  si tomamos como  $dE$  el valor dado por la ecuación 1.14 y 1.15. Si suponemos que el factor de oblicuidad  $\alpha(\theta)$  que aparece en esas ecuaciones es una función que varía con  $\theta$ , cuyo valor es uno en  $\theta=0$  y cero en  $\theta=\pi$ . Este resultado prueba la validez del principio de Huygens, y determina el carácter de las ondas secundarias.

En resumen, mientras una onda primaria pasa a través de la superficie  $\sigma$ , durante un intervalo de tiempo de cero a  $t_1$ , los distintos elementos de  $\sigma$  emiten ondas esféricas secundarias representadas por las ecuaciones 1.25 o por 1.26 donde el factor de oblicuidad  $q(\theta)$  decrece gradualmente de 1 a 0 a medida que  $\theta$  aumenta de 0 a  $\pi$ .

Las formas de las funciones que describen la dependencia temporal de la perturbación primaria  $E$  en  $\sigma$  y la perturbación secundaria  $dE_s$  en  $P$  se hallan comparadas en la figura 1.3(a) y (b). Como  $dE_s$  es proporcional (ec. 1.25) a la derivada temporal de  $E$ , y  $E(\alpha, t) = 0$  para  $t=0$  y  $t=t_1$ , el área total bajo la curva que representa a  $dE_s$  es cero. Consecuentemente, la perturbación secundaria es a veces positiva y a veces negativa, aunque la perturbación primaria tenga el mismo signo en todo instante.

En un instante dado, un punto  $P$  a una distancia  $r_0 \gg vt_1$  recibe ondas secundarias de una pequeña porción de la superficie a la cual hemos llamado la región activa. La región activa comienza a aparecer en el instante  $t=r_0/v$  cuando las ondas secundarias emitidas por  $C$  alcanzan a  $P$ . Desde este instante hasta  $t=r_0/v+t_1$ , la región activa se expande hasta convertirse en un disco cuyo borde exterior está a una distancia  $r_0+vt_1$  de  $P$ . Durante este intervalo de tiempo la superposición de las ondas secundarias emitidas por los puntos de la región activa da lugar a una perturbación resultante finita. Para  $t > r_0/v+t_1$ , la región activa adquiere la forma de una zona anular que se desplaza gradualmente desde la vecindad del polo  $O$  a la del antipolo  $O'$  a medida que  $t$  aumenta. La perturbación resultante es ahora cero debido a que las perturbaciones negativas recibidas en  $P$  desde una porción de

La region activa cancelan a las perturbaciones positivas  
 provenientes de otra parte de esta region. Al tiempo  $t = r + 2a/v$ ,  
 el punto  $P$  comienza a recibir la onda secundaria emitida por el  
 antipolo  $C'$  lo que significa que en este instante el borde  
 exterior de la region activa ha alcanzado al punto  $C'$ . Para  
 $t > r + 2a/v$ , la region activa es un disco centrado en  $C'$ . El radio  
 de este disco decrece a medida que  $t$  aumenta y se hace nulo en  
 $t = r + 2a/v + t$ , cuando el extremo final de la onda emitida por  $C'$   
 llega a  $P$ . Durante el intervalo de tiempo comprendido entre  
 $t = r - 2a/v$  y  $t = r + 2a/v + t$ , las porciones positivas y negativas de  
 la region activa no se anulan mutuamente. Sin embargo, la  
 perturbacion resultante en  $P$  es cero debido a que la intensidad  
 de las ondas secundarias emitidas hacia atrás es nula.

Como se indicó el principio de Huygens proporciona un poderoso  
 método para el estudio de los fenómenos que ocurren cuando la  
 propagación de las ondas es obstruida parcialmente por obstáculos  
 opacos. Supongamos, por ejemplo, que parte de la superficie  $\sigma$  es  
 cubierta por un obstáculo que no permite la transmisión de la  
 perturbación incidente. Como buena aproximación puede suponerse  
 que las ondas secundarias que llegan a  $P$  provenientes de la parte  
 no obstruida de  $\sigma$  no se modifican por la presencia del obstáculo.  
 Podemos calcular entonces la perturbación óptica en  $P$  sumando las  
 contribuciones de dichas ondas secundarias.

## **I.2.B. INTERFERENCIAS**

Hasta este momento hemos considerado a la luz desde el punto de vista geométrico (óptica geométrica), mas sin embargo debemos tener en cuenta que la luz tiene una forma ondulatoria y una naturaleza tal que puede transmitirse tanto en el espacio, como a través de un medio transparente (vidrio, aire, etc.) que la soporte. A partir del principio de que la luz tiene una forma ondulatoria, se tiene que cuando dos ondas del mismo tipo (de la misma fuente), que se originan desde dos fuentes, se traslapan en un punto en el espacio, la intensidad de la onda resultante en dicho punto puede ser mayor o menor a la intensidad de cualquiera de las dos ondas, a este efecto se le llama **interferencia**. Después de superponerse las ondas individuales se separan y continúan en su trayectoria completamente inafectadas por la interferencia de las mismas.

**INTERFERENCIA CONSTRUCTIVA:** Cuando la intensidad neta resultante es mayor que las intensidades individuales.

**INTERFERENCIA DESTRUCTIVA:** Cuando la intensidad neta es menor que las intensidades individuales.

**COHERENCIA:** Para que ocurra la interferencia las ondas deben ser mutuamente coherentes, esto es que provienen de la misma fuente cuando son monocromáticas tienen exactamente la misma frecuencia (algunos lasers) y no hay variación en la fase entre las ondas a lo largo del tiempo.

**Incoherencia:** en dos fuentes de luz distintas la emisión de luz por los átomos de una fuente es independiente de otra, la

crestas y valles de las ondas que proceden de las dos fuentes y mantienen una relación de fases definida y esto hace que sea incoherente.

### 1.2.C. INTENSIDAD EN EL EXPERIMENTO DE YOUNG.

También conocido como experimento de la interferencia en el cual se ilustra el experimento de Thomas Young: 1774 - 1829, uno de los primeros que se relacionaron directamente con las propiedades ondulatorias de la luz. El arreglo del equipo de laboratorio que se usa para llevar a cabo

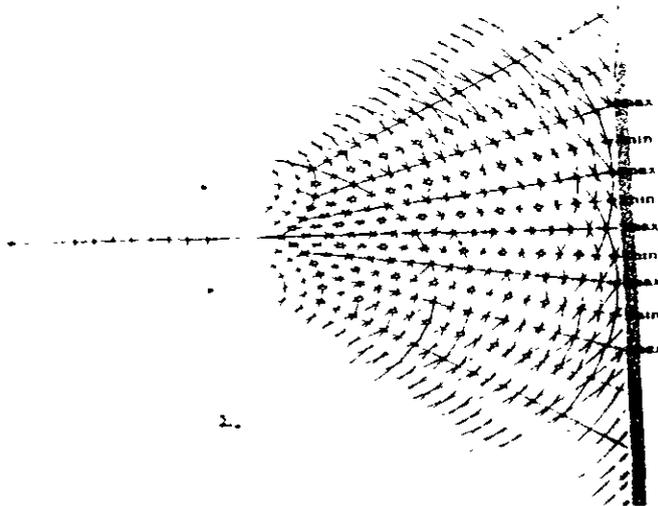


Figura 1.1. Arreglo del equipo de laboratorio para el experimento de Young, indicando las ranuras y las ondas que se propagan hacia la pantalla.

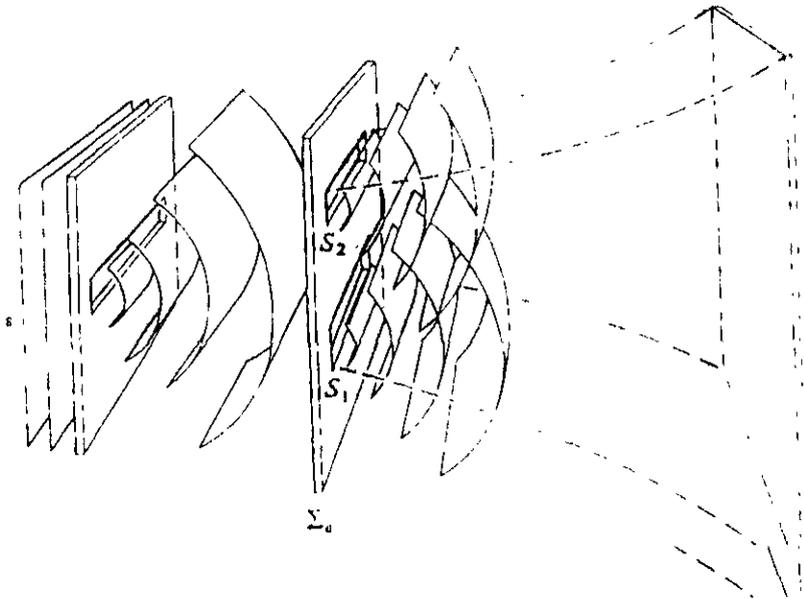


Figura 1.6 Arreglo del equipo de laboratorio para el experimento de Young, indicado las ranuras y las ondas que se propagan hacia la pantalla

Este experimento, que fue ejecutado por primera vez en 1801, se encuentra indicado en la figura 1.6. La luz de una fuente lejana cae sobre una ranura S que por ello actúa como fuente secundaria para un par de ranuras que se encuentran en la segunda pantalla. Estas ranuras son muy angostas y se encuentran separadas por solo una pequeña fracción de milímetros. La luz que pasa por estas ranuras,  $S_1$  y  $S_2$  en la figura 1.6, llega hasta la pantalla de observación, C.

En la figura 1.7 se encontrará la ampliación de una porción de la imagen que se forma en la pantalla de observación.

La imagen presenta líneas brillantes u oscuras ligeramente espaciadas que se conocen como "franjas". De principio de cuentas se deduce que las ranuras  $S_1$  y  $S_2$  actúan como fuentes secundarias de la fuente  $S$ . Las ondas que parten de  $S_1$  y  $S_2$  llegan a la pantalla de observación en fase e interfieren constructivamente para dar una franja brillante; las que llegan a la pantalla con una diferencia de fase de  $180^\circ$  interfieren destructivamente, por lo que la luz se anula por completo. La figura 1.7 muestra la distribución de la intensidad a lo largo de la pantalla de observación sobre la línea media tener una franja de intensidad máxima. Inmediatamente a los lados se presentan franjas oscuras. En seguida se presentan franjas brillantes u oscuras separadas por distancias equidistantes.

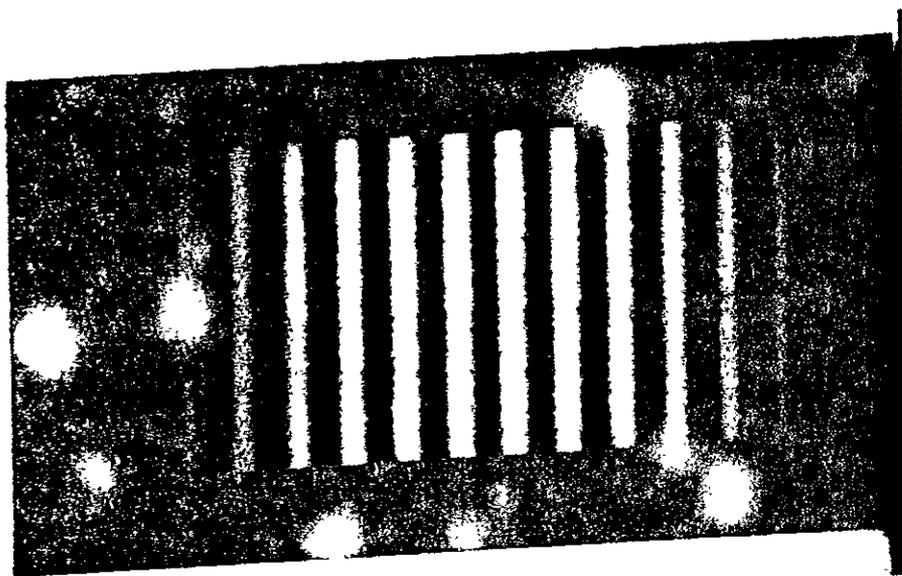


Figura 1.7 Ampliación de una parte de la pantalla de observación.

Como la luz que parte de las dos ranuras  $S_1$  y  $S_2$  presenta la misma fase, aparecerá una franja oscura cada vez que la onda que parte de una ranura recorra media longitud de onda mas que la que proviene de la otra. En la línea central la luz presentara un máximo, porque siempre es equidistante de las dos ranuras. Podemos encontrar la posición de la primera franja oscura determinando la posición en la cual la distancia desde una de las ranuras es mayor en media longitud de onda que la distancia desde la otra.

En la figura 1.8 se indican los parametros que intervienen. La distancia  $l$  entre las ranuras y la pantalla de observación es mucho mayor que la separación que guardan las ranuras entre si y estas mismas deben ser tan angostas que se puedan considerar como fuentes puntuales.

Trazaremos primero  $S_2A$  de tal manera que  $S_2D$  sea igual a  $AD$ . La línea que une la línea central en  $O$  con  $D$  es perpendicular a  $S_2A$  y las ranuras determinan una línea normal a la central. Como consecuencia de esto se tiene que los ángulos marcados con  $\alpha$  son iguales. La distancia  $S_2A$  debe ser igual a la mitad de la longitud de onda,  $\lambda / 2$ , para que se logre la diferencia de fase apropiada. Considerando ahora la igualdad de las tangentes de los dos ángulos  $\alpha$  tendremos:

$$y/l = (\lambda/2)/d \dots \dots \dots \text{Ec. I. 32}$$

Esta expresión relaciona la longitud de onda de la luz con otros parametros cuyo valor se determina en el experimento. Podemos usar los resultados del experimento de Young para determinar la longitud de onda de la luz. El experimento se lleva a cabo con una separación entre las ranuras de 0.1 mm y con la pantalla de

observación a 50 cm de las ranuras. La primera línea oscura queda a una distancia de 1mm de la franja brillante central. Si estos números se substituyen en la ecuación 1.32 (todas las unidades en milímetros) tenemos:

$$1 \times 500 = \lambda \times 1 \times 0.1$$

$$\lambda = 4000 \text{ m}\mu = 400 \text{ m}\mu \dots\dots\dots \text{Eq. 1.32}$$

La longitud de onda de la luz usualmente se da en milimicras o en unidades Angstrom. 1 milimicra (m $\mu$ ) = 10<sup>-4</sup> cm, 1 unidad Angstrom (A), = 10<sup>-8</sup> cm. El resultado anterior es importante, ya que indica el orden de magnitud de las longitudes de onda de la luz. La luz visible tiene longitudes de onda que se encuentran entre 400 m $\mu$  en la region violeta del espectro a 700 m $\mu$  en el extremo rojo del mismo.

No es facil llevar a cabo el experimento de Youngs porque deber satisfacerse muchas condiciones. Se acepta que la luz se encuentre en fase cuando pasa por las ranuras 1 y 2. La mayor parte de las fuentes luminosas son extensas, esto es, sus dimensiones son grandes y la luz la emite por un filamento caliente por una region llena de un gas calentado. En general, en estas fuentes no hay una relacion fija de la fase de la luz que se emite en sus diferentes puntos. Asi, aunque se presentan, generalmente no se ven efectos de interferencia porque la relacion entre las fases continua y rapidamente cambia. Esto se evita en el experimento descrito con el uso de S, que selecciona una pequena porcion de la fuente original.

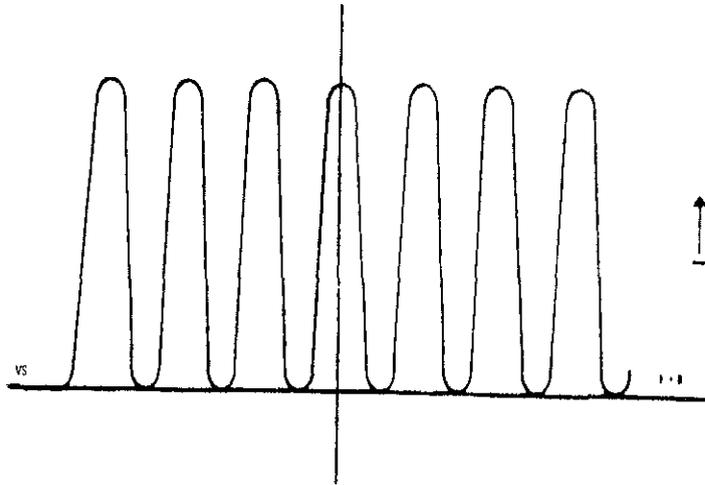


Figura I. 8 Distribucion de la intensidad de la luz que cae sobre la pantalla de observacion.

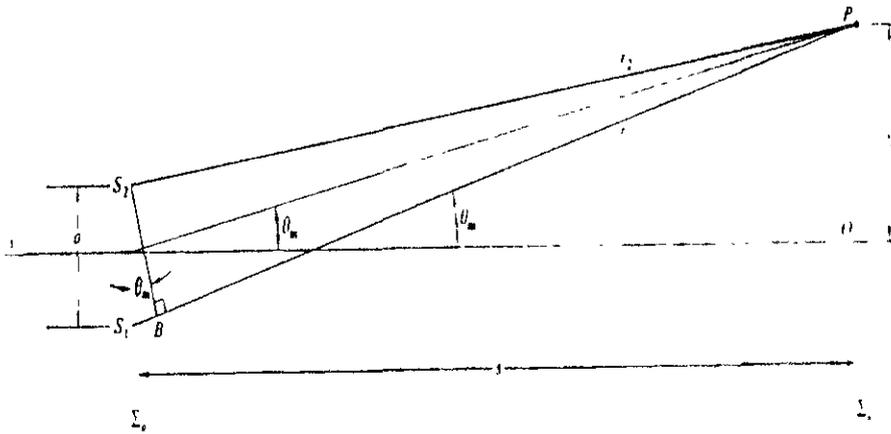


Figura I. 9 El experimento de Young con los parametros que se usan en su análisis.

## I.2.D. INTENSIDAD DE LA INTERFERENCIA EN EL EXPERIMENTO DE YOUNG

### I.2.D.1. OBTENCIÓN DE MÁXIMOS Y MÍNIMOS:

Volviendo a las franjas de interferencia que se observan en la figura 1.10 las cuales corresponden a máximos y mínimos de la intensidad de la interferencia del experimento de Young, tomaremos como base a la figura 1.10 para obtener las ecuaciones correspondientes a dichos máximos y mínimos. Consideremos que  $P$  es un punto cualquiera sobre la pantalla  $C$  de la figura 1.10, y dicho punto  $P$  se encuentra a las distancias  $r_1$  y  $r_2$  de las rejillas  $S_1$  y  $S_2$ , respectivamente. Se traza la línea  $S_1P$  de forma tal que las líneas  $PS_1$  y  $PS_2$  tengan longitudes iguales. Siendo  $d$  la distancia entre las rejillas y a su vez  $d$  es mucho menor a  $f$  (que es la distancia entre las pantallas  $B$  y  $C$ ), se tiene entonces que  $S_1P$  es casi perpendicular a  $r_1$ ,  $r_2$ . Esto significa que el ángulo  $S_1PS_2$  es casi igual al ángulo  $PaC$ , estando ambos ángulos marcados  $\theta$  en la figura 1.10; de forma equivalente, puede considerarse que las líneas  $r_1$  y  $r_2$  son casi paralelas.

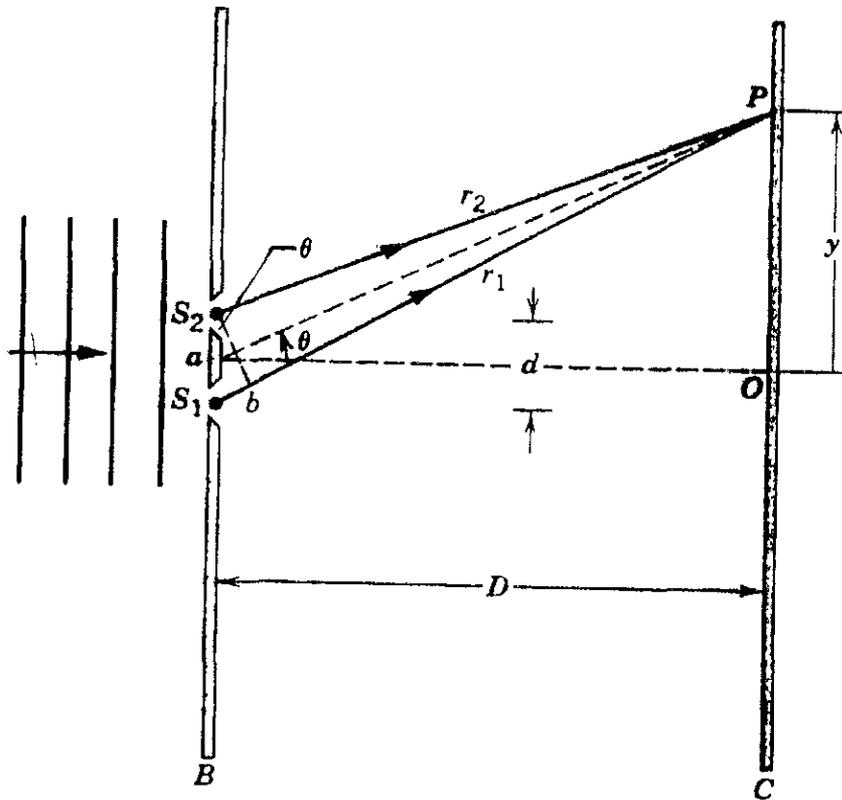


Figura I. 10 Los rayos que proceden de  $S_1$  y  $S_2$  se combinan en  $P$ . En realidad,  $D \gg d$ ; la figura se ha distorsionado por claridad. El punto  $a$  es el punto central entre las rendijas

Ya que las ondas generadas en las rejillas  $S_1$  y  $S_2$  provienen de una misma fuente, la variación que tienen es únicamente que recorren longitudes diferentes, por lo que cuando llega al punto  $P$  lo hacen con una diferencia de fase. El número de longitudes de onda contenido en el tramo  $S_1b$  determina el tipo de interferencia

que se produce en P. Por lo que para poder obtener un máximo en P, las dos ondas deben llegar en fase y así el tramo ST (que es igual a  $d \sin \theta$ ) debe contener un número entero de longitudes de onda  $\lambda$ .

$$S.P = d \sin \theta = m \lambda \quad \text{donde } m = 0, 1, 2, 3, \dots \text{ máximos} \dots \text{Ec. 1. 34}$$

Para un mínimo en P, las dos ondas deben diferir en fase por un múltiplo impar de  $\pi$ , para el cual  $S.P (= d \sin \theta)$  debe contener un número semientero de longitudes de onda, o sea

$$S.P = d \sin \theta = (m + 1/2) \lambda \quad \text{donde } m = 0, 1, 2, 3, \dots \text{ (mínimos)} \dots \text{Ec. 1. 35}$$

o

$$S.P = d \sin \theta = n \lambda \quad \text{donde } n = 1/2, 3/2, 5/2, \dots \text{ (mínimos)} \dots \text{Ec. 1. 36}$$

Notese que para cada máximo o para cada mínimo arriba del punto O se tiene un máximo o mínimo simétrico debajo de O, utilizando  $m = -1, -2, -3, \dots$

Las ecuaciones anteriores 1.34 y 1.35 dan las posiciones de los máximos y mínimos, mas no indican como varía la intensidad entre ellos, por lo que se tiene que obtener una ecuación de intensidad I para cualquier punto P determinado por el ángulo  $\theta$ .

Supongamos que las componentes del campo eléctrico de las dos ondas de la figura 1.10 varía con el tiempo en el punto P como

$$E_1 = E_0 \sin \omega t \dots \dots \dots \text{Ec. 1. 37}$$

$$E_2 = E_0 \sin (\omega t + \phi) \dots \dots \dots \text{Ec. 1. 38}$$

donde  $\omega (=2\pi\nu)$  es la frecuencia angular de las ondas y  $\phi$  es la diferencia de fase entre ellas. Nótese que  $\phi$  depende de la posición del punto P, que está determinado por el ángulo  $\theta$  de la figura 1.10. Suponemos que las rejillas son tan angostas que la luz difractada de cada rejilla ilumina la porción central de la pantalla uniformemente. Esto significa que cerca del centro de la pantalla  $E_0$  es independiente de la posición de P, esto es, del valor de  $\theta$ .

Si la separación  $d$  entre las rejillas es mucho menor que la distancia  $D$  a la pantalla C, los vectores  $E$  de las dos ondas que interfieren son casi paralelos, y podemos reemplazar la suma vectorial de los campos  $E$  con la suma escalar de sus componentes,

$$E = E_1 + E_2 \quad \text{Ec. 1. 39}$$

lo cual, puede escribirse (como se verá en el subcapítulo 1.4.3),

$$E = E_0 \sin(\omega t + \beta) \dots \text{Ec. 1. 40}$$

donde la fase  $\beta$  es

$$\beta = \frac{1}{2}\phi \dots \dots \dots \text{Ec. 1. 41}$$

y la amplitud es

$$E_0 = 2E_1 \cos \beta \dots \dots \dots \text{Ec. 1. 42}$$

La amplitud  $E_0$  de la perturbación de la onda resultante, que determina la intensidad de las franjas de interferencia, depende de  $\beta$ , la que a su vez depende del valor de  $\theta$ , esto es, de la posición del punto P en la figura 1.10. El valor máximo posible

de la amplitud  $E_0$  es  $2E_0$ , igual al doble de la amplitud  $E_0$  de las ondas que se combinan, en correspondencia con un refuerzo completo

La intensidad  $I$  de una onda electromagnética es proporcional al cuadrado de la amplitud  $E_m$  de su campo eléctrico:

$$I = (1/(2\mu_0 c)) E_m^2 \quad \text{Ec. 1. 43}$$

La razón de las intensidades de las dos ondas de luz puede expresarse, por consiguiente, como la razón de los cuadrados de las amplitudes de sus campos eléctricos. Si  $I_0$  es la intensidad de la onda resultante en P, e  $I_1$  es la intensidad que produciría cada una de las ondas actuando por sí misma, entonces

$$I_0 / I_1 = (E_0 / E_1)^2 \quad \text{Ec. 1. 44}$$

Al combinar las ecuaciones 1.42 y 1.44 obtenemos

$$I_0 = I_1 \cos^2 \beta \quad \text{Ec. 1. 45}$$

Notese que la intensidad de la onda resultante en cualquier punto P varía desde cero [por ejemplo, para un punto en el que  $\phi (= 2\beta = \pi)$ ] hasta cuatro veces la intensidad  $I_1$  de cada onda individual [por ejemplo, para un punto en el que  $\theta (= 2\beta = 0)$ ]. Calculemos  $I_0$  en función del ángulo  $\theta$  en la figura 1.10. La diferencia de fase  $\phi$  en la ecuación 5 se asocia con la diferencia de trayectorias  $S_1b$  en la figura 1.10. Si  $S_1b$  es  $1/2 \lambda$ ,  $\phi$  es  $\pi$ , si

Si  $b$  es  $\lambda$ ,  $\phi$  es  $2\pi$ , y así sucesivamente. En general, la diferencia de fases  $\phi/2\pi = \text{diferencia de trayectorias} / \lambda$ .

Haciendo que  $\phi$  sea la diferencia de fases y recordando que la diferencia de trayectorias es  $d \sin \theta$ , podemos escribir esto como

$$\phi = (2\pi / \lambda) (d \sin \theta)$$

o utilizando la ecuación 1.4.8

$$\beta = \phi/2 = (\pi d / \lambda) \sin \theta \dots \dots \dots \text{Ec. 1. 46}$$

por lo tanto, la intensidad en cualquier  $\theta$  puede escribirse como

$$I_{\theta} = 4I_0 \cos^2 \phi/2 \dots \dots \dots \text{Ec. 1. 47}$$

$$I_{\theta} = 4I_0 \cos^2 ((\pi d \sin \theta) / \lambda) \dots \dots \dots \text{Ec. 1. 47}$$

De la ecuación 1.47 vemos que los máximos de intensidad ocurren cuando  $\cos^2 \phi/2 = 1$  o sea

$$\phi = 2m\pi \quad \text{donde } m = 0, \pm 1, \pm 2, \dots$$

Al usar la ecuación 1.46 podemos escribir que

$$d \sin \theta = m\lambda \quad \text{donde } m = 0, \pm 1, \pm 2, \dots \quad (\text{máximos})$$

La cual es la misma que la ecuación 1.34. Los mínimos de intensidad ocurren, de acuerdo a la ecuación 1.47,

cuando

$$\cos^2 \phi/2 = 0, \text{ o sea}$$

$$\phi = (2m + 1)\pi \quad \text{donde } m = 0, \pm 1, \pm 2, \dots$$

lo que podemos escribir usando la ecuación 1.46 como

$$d \sin \theta = (m + 1/2) \lambda \quad \text{donde } m = 0, \pm 1, \pm 2, \dots \quad (\text{mínimos})$$

de acuerdo con la ecuación 1.34.

La figura 1.11 muestra el patrón de intensidades para la interferencia en el experimento de Young. La línea continua horizontal, es  $I_0$ , esta describe el patrón de intensidad (uniforme en la pantalla cuando una de las rejillas se cubre. Si las dos fuentes fueran incoherentes, por lo tanto no existiría la interferencia, pero con la misma amplitud, la intensidad sería uniforme y con un valor de  $2I_0$ , esto se muestra por la línea horizontal discontinua en la figura 1.11. Para fuentes coherentes con la misma amplitud, se espera que la energía se redistribuya en la pantalla, ya que recordemos que habíamos hecho la mención que las ondas después de la interferencia no sufren ningún decremento en su cantidad de energía.

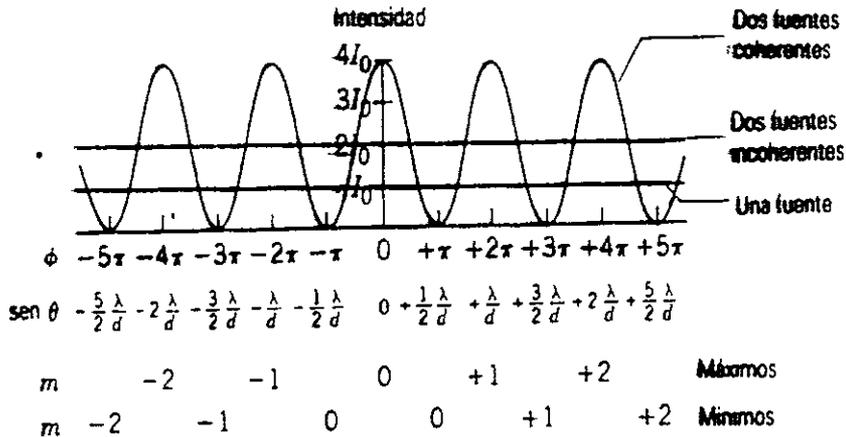


Figura 1. 11 Anteriormente se menciono que ocurre una coherencia cuando las ondas que provienen de la misma fuente o cuando son monocromaticas tienen exactamente la misma longitud y no hay variacion en la fase entre las ondas a lo largo del tiempo.

Ejemplo I. 1.

El dispositivo de rendija de la figura I.10 está iluminado con luz procedente de una lámpara de vapor de mercurio filtrada de modo que sólo sea visible la línea de color verde intenso ( $\lambda = 560 \text{ nm}$ ). Las rendijas están separadas por  $0.12 \text{ mm}$ , y la pantalla sobre la que aparece el patrón de interferencia está a  $55 \text{ cm}$  de distancia. ¿Cuál es la posición angular del primer mínimo?

En el primer mínimo hacemos  $m = 0$  en la ecuación I.36, o sea  $\text{sen } \theta = ((m + 1/2)\lambda) / d = ((1/2)(546 \times 10^{-9} \text{ m}) / (0.12 \times 10^{-3} \text{ m})) = 0.0023$

Este valor del  $\text{sen } \theta$  es tan pequeño que podemos considerar su valor  $\theta$ , expresado en radianes; expresado en grados es  $0.13^\circ$ .

### I.2.E. COHERENCIA

Hasta este momento para la superposición de ondas se había considerado que las ondas pueden ser completamente coherentes o completamente incoherentes, más sin embargo esto es solo una idealización con fines matemáticos, más que una realidad física. Entre ambos extremos de coherencia e incoherencia está la llamada **coherencia parcial**. Ejemplo de esto es el caso de la luz solar que se considera de una fuente incoherente, se pueden producir las franjas de interferencia en el experimento de Young siempre y cuando la separación entre las rejillas sea muy cercana, además que la rejilla S debe ser muy angosta (ver figura I.5), asegurándose con esto que los trenes de onda que llegan a S1 y S2 se originan en la misma pequeña región de la fuente. De hecho Young tomó en su experimento como fuente la luz solar.

En las fuentes comunes de luz visible, ya sean alambres incandescentes o por descargas eléctricas a través de un gas, se tiene que la emisión de luz ocurre en átomos individuales, no actuando en modo conjunto. La emisión de luz por un átomo individual dura aproximadamente  $10^{-8}$  seg, siendo en este caso la luz un tren de ondas finito (ver Figura 1.12), y dicho tren de ondas tiene una longitud de unos cuantos metros. En fuentes de luz como la de los tubos de descarga en gas a baja presión, los trenes de ondas son de unos cuantos centímetros de longitud. Constituyéndose dichas longitudes como la limitante para que la luz que procede de dichas fuentes se considere coherente. Hasta 1960 se pudo construir fuentes de luz visible que produjera un tren de onda infinito, como se ve en la figura 1.12. El Laser produce una luz altamente coherente con longitudes de trenes de onda que van de decenas a centenas de kilómetros.

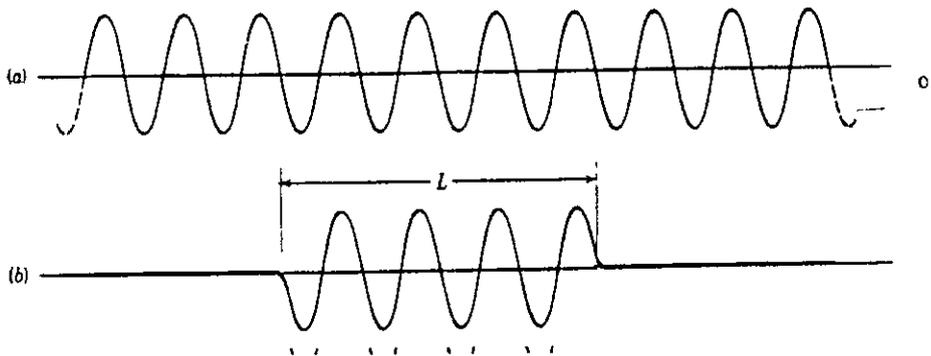


Figura 1. 12 (a) Sección de una onda infinita. (b) Tren de ondas de longitud finita  $L$ .

La coherencia parcial puede demostrarse también en el experimento de Young, utilizando un espejo parcialmente plateado, produciendo la trayectoria de las dos ondas que son  $S_1$  y  $S_2$  tengan longitudes diferentes. Si la diferencia de las trayectorias es pequeña con respecto a la longitud del tren de ondas, las franjas de interferencia se ven claramente y tienden a cero en los mínimos. Si la distancia de las trayectorias se hace más larga, las ondas comienzan a perder su coherencia, y las franjas se vuelven menos brillantes. Por último si la distancia de las trayectorias es más larga que la longitud del tren de ondas, las franjas se pierden por completo, convirtiéndose las ondas en incoherentes, por lo que se tiene en la pantalla de observación solo una luz con intensidad uniforme.

Es conveniente por motivos de cálculo separar los efectos de la coherencia en dos clasificaciones: temporal y espacial, aunque esto resulte artificial.

### **I.2.E.1. COHERENCIA TEMPORAL:**

Se considera que para que una luz fuera completamente monocromática, debería tener una frecuencia constante durante un lapso de tiempo infinito, sin embargo esto no es posible. Por lo que definiremos el tiempo de coherencia ( $\Delta t$ ) como el intervalo temporal sobre el cual podemos predecir razonablemente la fase de la onda luminosa en un punto dado en el espacio. Por lo que se dice que una onda tiene un alto grado de coherencia temporal si su tiempo de coherencia es grande.

### I.2.E.2. COHERENCIA ESPACIAL :

Si dos puntos desplazados lateralmente están sobre el mismo frente de onda en un tiempo dado, los campos en esos puntos se dice que son coherentes espacialmente.

La idea de coherencia espacial se usa frecuentemente para describir efectos que surgen por la extensión finita de las fuentes de luz ordinaria, a diferencia de la coherencia temporal que se refiere a la frecuencia estable de la fuente de luz.

Supongamos que tenemos una fuente monocromática extendida. Dos puntos radiadores sobre ella separados por una distancia lateral grande a comparación de la longitud de onda ( $\lambda$ ), para el ejemplo del experimento de Young, el tiempo durante el cual exista el tren de onda (longitud del tren de onda) de la fuente debe ser tal su duración que sea mayor al tiempo que tarda la onda más cercana al punto de observación en llegar a dicho punto para poder efectuarse la interferencia, y cumplir con la coherencia espacial.

### I.3. SUMA DE PERTURBACIONES ONDULATORIAS

En este caso se utilizará un método gráfico para la suma de dos o más perturbaciones de onda, por ser más fácil de manejar a comparación de un método algebraico. Dicho método gráfico se basa en los fasores rotatorios.

Una perturbación de onda senoidal como la de la ecuación 1.3 puede representarse gráficamente usando un fesor rotatorio. En la figura 1.4.13 un fesor de magnitud  $E_0$  gira con respecto al origen en una dirección antihoraria con una frecuencia angular  $\omega$ .

perturbación alternante  $E_1$  de la onda (ecuación 1.37), se representa mediante la proyección de este fasor sobre el eje vertical.

Una segunda perturbación de onda  $E_2$ , dada por la ecuación (1.38), que tiene la misma amplitud  $E_0$  pero con una diferencia de fases  $\phi$  con respecto a  $E_1$ , puede representarse gráficamente como se observa en la figura 1.13, como la proyección sobre el eje vertical de un segundo fasor de la misma magnitud  $E_0$  que forma un ángulo  $\phi$  con respecto al primer fasor. La suma  $E$  de  $E_1$  y  $E_2$  será la suma de las proyecciones de los dos fasores sobre el eje vertical. Esto se ve con claridad si se dibuja los fasores como se muestra en la figura 1.13, situando el pie de un fasor en la punta de la otra, manteniendo la diferencia de fase apropiada, y haciendo que todo el conjunto siga girando en sentido antihorario alrededor del origen.

En la figura 1.13,  $E$  puede considerarse como la proyección sobre el eje vertical de un fasor de longitud  $E_0$ , que es la suma vectorial de los dos fasores de magnitud  $E_0$ . De dicha figura, se ve que la proyección puede escribirse como

$$E = E_0 \sin(\omega t + \beta)$$

Que es de acuerdo a la ecuación (1.40). Obsérvese que la suma algebraica de las proyecciones de los dos es igual a la proyección de la suma vectorial de los dos fasores.

En la mayoría de los problemas de óptica sólo se considera la amplitud  $E_0$  de la perturbación ondulatoria resultante y no su variación en el tiempo, esto debido a que tanto el ojo humano y los instrumentos de medición comunes responden a la intensidad

resultante de la luz, y no pueden responder a las rápidas variaciones de tiempo que caracteriza a la luz visible. Ejemplo de esto es la luz de sodio, la cual tiene una longitud de onda  $\lambda = 589 \text{ nm}$ , su frecuencia  $\nu = \omega/2\pi$  es de  $5.1 \times 10^{14} \text{ Hz}$ . De lo anterior se desprende que para la mayoría de las ocasiones no se necesita considerar la rotación de los fasores, sino únicamente la amplitud del fesor resultante.

En la figura 1.13 los tres fasores forman un triángulo isosceles cuyos lados tienen longitudes  $E_1$ ,  $E_1$  y  $E_R$ . Para cualquier triángulo, un ángulo exterior ( $\phi$  en este caso) es igual a la suma de los dos ángulos interiores opuestos (en este caso  $\beta$  y  $\beta$ ), por lo que

$$\beta = \pi - \phi$$

De la misma figura 1.13 se ve que la longitud base de este triángulo es

$$E_R = 2E_1 \cos \beta$$

Que son los mismos resultados que las ecuaciones 1.41 y 1.42. El procedimiento general para encontrar la resultante de más de dos perturbaciones ondulatorias que varían senoidalmente será:

- Construir una serie de fasores que representen la función que se va a sumar. Dibujarlos extremo por extremo, manteniendo las relaciones de fase apropiadas entre fasores contiguos.
- Construir la suma de este arreglo de fasores, en forma análoga a una suma de vectores. La longitud del fesor resultante da la amplitud del campo eléctrico. El ángulo entre este fesor y el primer fesor es la fase de la resultante con respecto al primer fesor. La proyección de este fesor sobre el

eje vertical da la variación en el tiempo de la perturbación de la onda resultante.

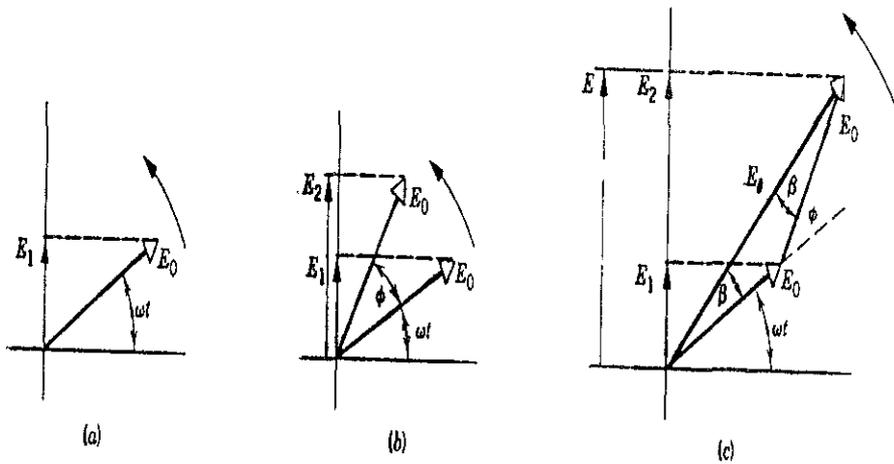


Figura 1. 13 (a) una onda  $E_1$  variable en el tiempo se representa mediante un vector rotatorio o fasor, (b) Dos ondas  $E_1$  y  $E_2$  que difieren en fase por  $\phi$ . (c) Otra manera de trazar (b)

### I.3.A. CAMBIOS DE FASE EN LA REFLEXIÓN

G.G. Stokes para poder investigar la reflexión de la luz en una interfaz entre dos medios, empleo el principio de la reversibilidad óptica, dicho principio que afirma que si no existe absorción de luz, un rayo de luz que se refleje o se

refracte trazara de nuevo su trayectoria original al invertirse su direccion.

La figura I.14.a muestra una onda con amplitud  $E$  reflejada y refractada en una superficie que separa a los medios 1 y 2, donde  $n_2 > n_1$ . La amplitud de la onda reflejada es  $r_{12}E$ , donde  $r_{12}$  es un coeficiente de la amplitud de reflexion. La amplitud de la onda refractada es  $t_{12}E$ , donde  $t_{12}$  es un coeficiente de la amplitud de transmision.

El signo del coeficiente indica la fase relativa de la componente reflejada o transmitida. Si se considera unicamente la posibilidad de los cambios de fase de  $0^\circ$  o de  $180^\circ$ , entonces si  $r_{12} = +0.5$ , por ejemplo, se tiene una reduccion en la amplitud de la reflexion a la mitad y no hay un cambio de fase. Pero si  $r_{12} = -0.5$  se tiene un cambio de fase de  $180^\circ$ , ya que:

$$E \text{ sen } (\omega t - 180^\circ) = -E \text{ sen } \omega t$$

En la figura I.14.b los rayos indicados por  $r_{12}E$  y  $t_{12}E$  se han invertido en direccion. El rayo  $r_{12}E$ , se refleja y se refracta produciendo los rayos de amplitudes  $r_{12}^2E$  y  $r_{12}t_{12}E$ . El rayo  $t_{12}E$  tambien se refleja y se refracta, produciendo los rayos de amplitudes  $t_{12}r_{21}E$  y  $t_{12}t_{21}E$ . Notese que  $r_{12}$  describe un rayo en el medio 1 reflejado del medio 2, y  $r_{21}$  describe un rayo en el medio 2 reflejado en el medio 1. En forma similar  $t_{12}$  describe un rayo que pasa del medio 1 al medio 2;  $t_{21}$  describe un rayo que pasa del medio 2 al medio 1.

Basandose en el principio de reversibilidad, se concluye que los dos rayos que están arriba a la izquierda de la figura I.14.b deben ser equivalentes al rayo incidente de la figura I.14.a,

invertido; los dos rayos que están abajo a la izquierda de la figura I.14.b deben cancelarse. Este segundo requisito conduce a:

$$r_{12}t_{12}E + t_{12}r_{21}E = 0$$

o

$$r_{12} = -r_{21}$$

Este resultado indica que si comparamos una onda reflejada desde el medio 1 con otra reflejada desde el medio 2, se comportan de modo diferente en cuanto a que una o la otra experimenta un cambio de fase de  $180^\circ$ .

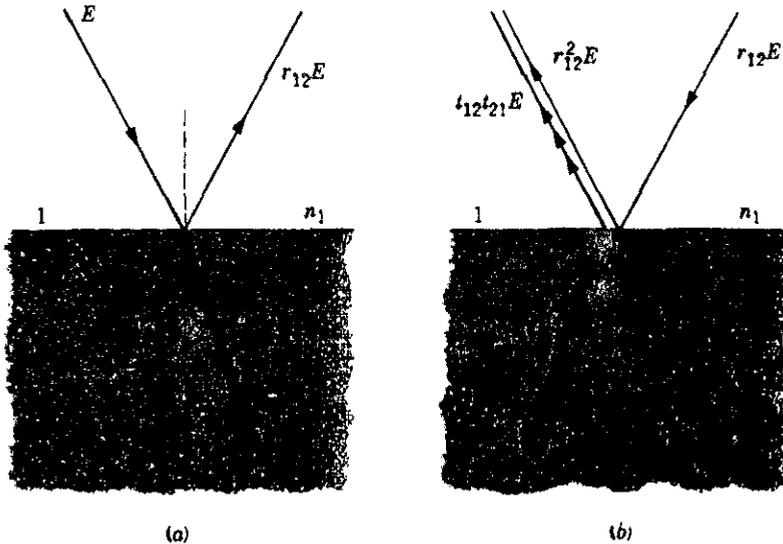


Figura I. 14 (a) Un rayo se refleja y se refracta en una interfaz. (b) La situación ópticamente invertida; los dos rayos abajo a la izquierda deben cancelarse.

Para demostrar el cambio de fase por reflexion se utiliza el experimento del espejo de LLOYD (figura 1.15). En este experimento se demuestra que la onda reflejada desde el medio opticamente mas denso sufre un cambio de fase de  $180^\circ$ . En un punto P arbitrario como resultado del traslape de las ondas directa y reflejada se tiene una interferencia. Aqui al igual que en el experimento de Young se tiene dos fuentes, una S (fuente real) y S' que es una imagen virtual de S en el espejo plano. Sin embargo existe una diferencia importante entre el experimento de LLOYD y el experimento de Young, y es que la luz que parte de la fuente virtual S' se ha reflejado desde el espejo y ha experimentado un cambio de fase de  $180^\circ$ . Como resultado de dicho cambio de fase, el borde inferior de la pantalla (punto C) muestra una franja oscura (ver figura 1.16), en lugar de la franja clara que aparece en el mismo punto O en el experimento de Young (esto es al centro de la pantalla). Expresado de otra forma, la aparicion de la franja oscura en O muestra que uno de las ondas interferentes se ha desplazado  $180^\circ$  en fase. Puesto que no hay nada que cambie la fase del haz directo SP, debe ser el haz reflejado el que experimenta el cambio de fase. Con esto se demuestra que la reflexion desde un medio opticamente mas denso implica un cambio de fase de  $180^\circ$ .

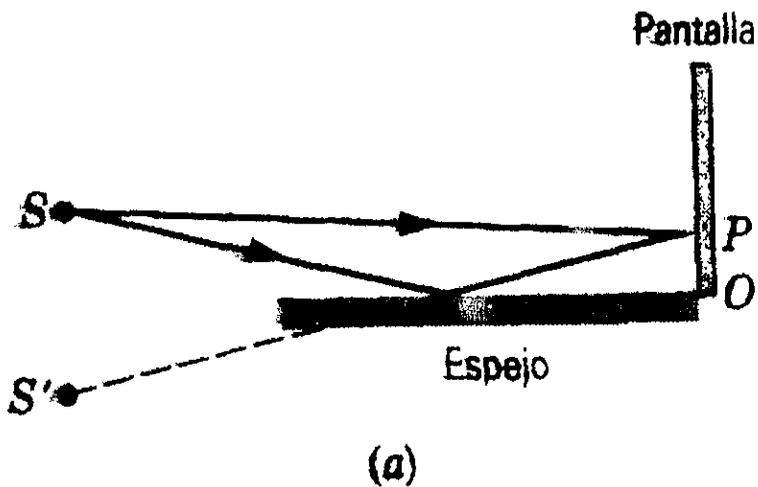


Figura I. 15 El arreglo experimental del espejo de Lloyd. Aparecen franjas en la pantalla como resultado de la interferencia entre los haces directo y reflejado.

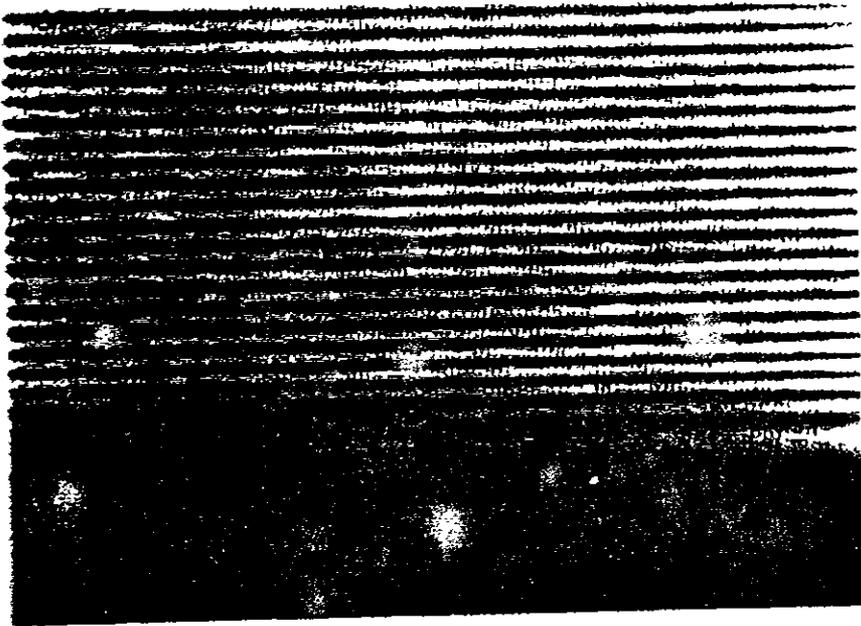


Figura 1. 16 Se observan las franjas en el experimento del espejo de Lloyd.

### I.3.B. EL INTERFEROMETRO DE MICHELSON

Hasta este momento solo habíamos visto el efecto de la interferencia a través del procedimiento de división de frente de onda (como en el experimento de Young o en el experimento de Lloyd), pero el interferómetro de Michelson, está basado en el procedimiento de división de amplitud.

Es importante hacer la mención que existe un interferómetro estelar de Michelson, que no es el mismo que el interferómetro espectral de Michelson, cuyo objetivo es el medir el diámetro angular de estrellas. Dicho interferómetro no lo trataremos en este capítulo.

El interferómetro espectral de Michelson es un instrumento, cuya función básica es medir la longitud de onda ( $\lambda$ ) de una señal. Existen una gran cantidad de interferómetros, y entre los interferómetros de división de amplitud, el más conocido e históricamente el más importante es el interferómetro espectral de Michelson. Su configuración es la que se muestra en la figura 1.17, y se trata de una fuente de luz extensa (que puede ser una placa difusora de vidrio esmerilado iluminada por una lámpara de descarga) que emita una onda perpendicular al conjunto de espejos que se tiene. La luz llega en primera instancia al divisor de haz O (es un espejo semiplateado), el cual está en un ángulo de  $45^\circ$  con respecto al haz de luz que llega. Dicho divisor O divide la onda en dos, una que es la que atraviesa al divisor O y se dirige al espejo  $M_1$ , y otra que fue la reflejada en el divisor O y se dirige al espejo  $M_2$ . Dichas ondas son reflejadas por los espejos  $M_1$  y  $M_2$ , regresando al divisor de haz O. Parte de la onda que viene de  $M_2$  pasa a través del divisor O dirigiéndose al detector D. Así mismo parte de la onda que viene de  $M_1$  es

desviada por el divisor  $O$  y se dirige también al detector  $D$ , por ende las dos ondas se unen y es posible que se produzca la interferencia. Para lograr que ambas ondas recorran la mismas trayectorias se pone una placa compensadora  $C$  entre el divisor  $O$  y el espejo  $M_1$ , dicha placa  $C$  es un duplicado exacto al divisor  $O$ , excepto que no tiene el recubrimiento que tiene el divisor  $O$ . La placa  $C$  es paralela al divisor  $O$  con esto se garantiza que ambos caminos ópticos sean equivalentes y las distancias a  $M_1$  y  $M_2$ , sean iguales cuando se mide a partir del punto en el que se separa el haz.

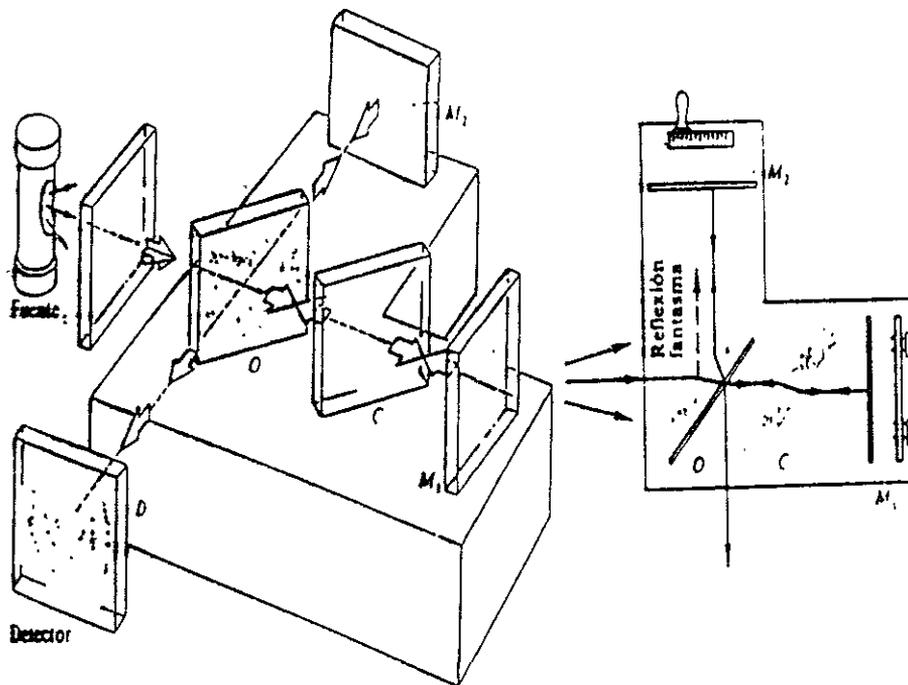


Figura 1. 17 El interferómetro de Michelson.

El espejo  $M_1$  es fijo sólo pudiendo ajustar su inclinación para poder estar siempre perpendicular al haz luminoso, en tanto que el espejo  $M_2$  además de tener la misma característica, está montado en un tornillo micrométrico, por lo que  $M_2$  puede moverse en dirección paralela al haz de luz, y su desplazamiento se puede determinar con toda exactitud. Cuando  $M_2$  se mueve, acercándose o alejándose al divisor  $O$ , la diferencia de trayectorias y por ende de fase entre las dos ondas varía, con lo que el observador ve que las franjas se mueven a través en el detector  $D$ . El detector  $D$  tiene un índice o retícula que proporciona un punto de referencia al observador. A medida que  $M_2$  se mueve, el observador cuenta el número de franjas que cruzan la retícula. Al moverse  $M_2$  la distancia  $\lambda / 4$ , el largo del recorrido entre el divisor  $O$  y el espejo  $M_2$  cambia en una  $\lambda$ , ya que el haz recorre dos veces la distancia  $O M_2$  (una vez de ida y otra de regreso). Un cambio de  $\lambda / 4$  en la posición de  $M_2$  hace que en la retícula una franja oscura pase un lugar a la izquierda o la derecha, reemplazándose por una franja luminosa o viceversa. La distancia que se mueve  $M_2$  dividida entre el número de franjas que cruzan la retícula equivale a  $\lambda / 4$ .

Ejemplo: En un interferómetro de Michelson, para poder medir la longitud de onda de la luz amarilla del sodio, se encontró que para contar 100 franjas el espejo  $M_2$  se desplazó  $1.47 \times 10^{-2}$  mm., entonces:

$N$  ; es el número de franjas:

$d$  ; es el desplazamiento de  $M_2$ .

$$N \times \lambda / 4 = d$$

$$100 \times \lambda / 4 = 1.47 \times 10^{-2} \text{ mm}$$

$$\lambda = 589 \text{ m}\mu$$

Que corresponde al valor de la longitud de onda de la luz amarilla.

#### I.4. DIFRACCION

La base del modelo de rayos de la luz y de las ideas asociadas con la propagación rectilínea de esta, proviene de la observación de las sombras bien recortadas que se presentan cuando una pantalla se ilumina con la luz de una fuente lejana o puntual. Esto se ilustra en la figura 1.18 donde se escogió una fuente puntual para eliminar la penumbra. En realidad la situación es mucho más compleja. La figura 1.18 muestra la distribución detallada de la intensidad de la luz en la región cercana al borde de la sombra.

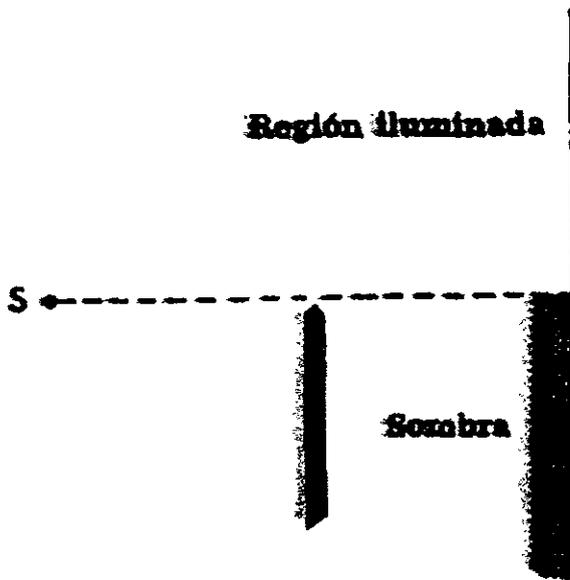


Figura 1. 18 Sombra geométrica formada por un objeto opaco bajo la suposición de la propagación rectilínea de la luz.

Los efectos como el que se indica en la figura I.19 se conocen como efectos de difracción y se observan siempre que un objeto se encuentra en el camino de un frente de onda limitando su tamaño. Frecuentemente no se observan los efectos de la difracción porque la mayor parte de las fuentes son extensas con lo que los patrones de difracción producidos por sus diferentes puntos se superponen y se cancelan estadísticamente. Con frecuencia se considera a la difracción como la flexión de la luz alrededor de los obstáculos. Esto no es estrictamente correcto, la **difracción** es aquella que se presenta en forma natural como resultado de la propagación del frente de onda que no es eliminado por el objeto que se interpone.

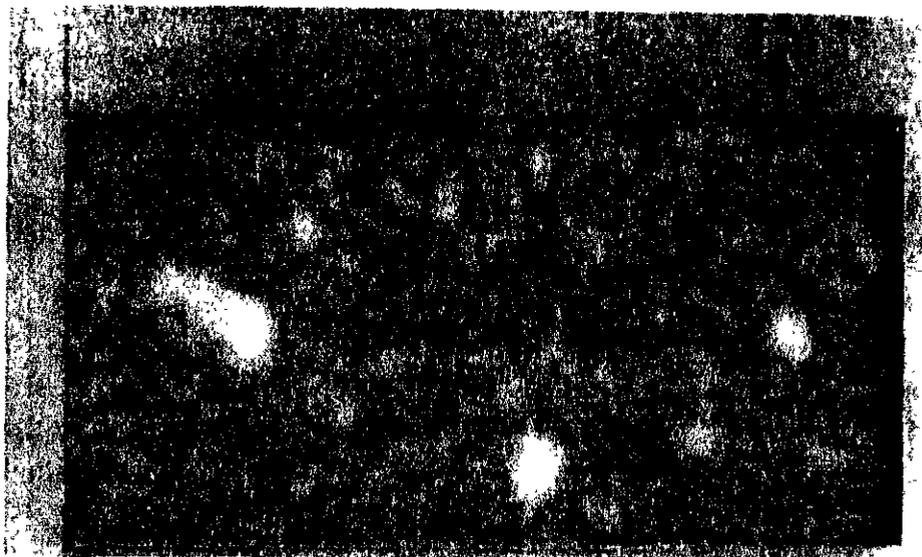


Figura I. 19 Patrón de difracción que resulta de la sombra de un borde recto como el que se indica en la figura anterior. Esta figura se amplificó para mostrar los detalles. La intensidad de la luz en la región cercana a la sombra geométrica es el doble de la de la región abierta de la izquierda.

Para que se presente la difracción, el tamaño del objeto debe ser del mismo orden de magnitud que la longitud de onda de las ondas incidentes, cuando la longitud de onda es mucho menor que el tamaño del objeto, por lo general, la difracción no se observa y el objeto forma una sombra definida.

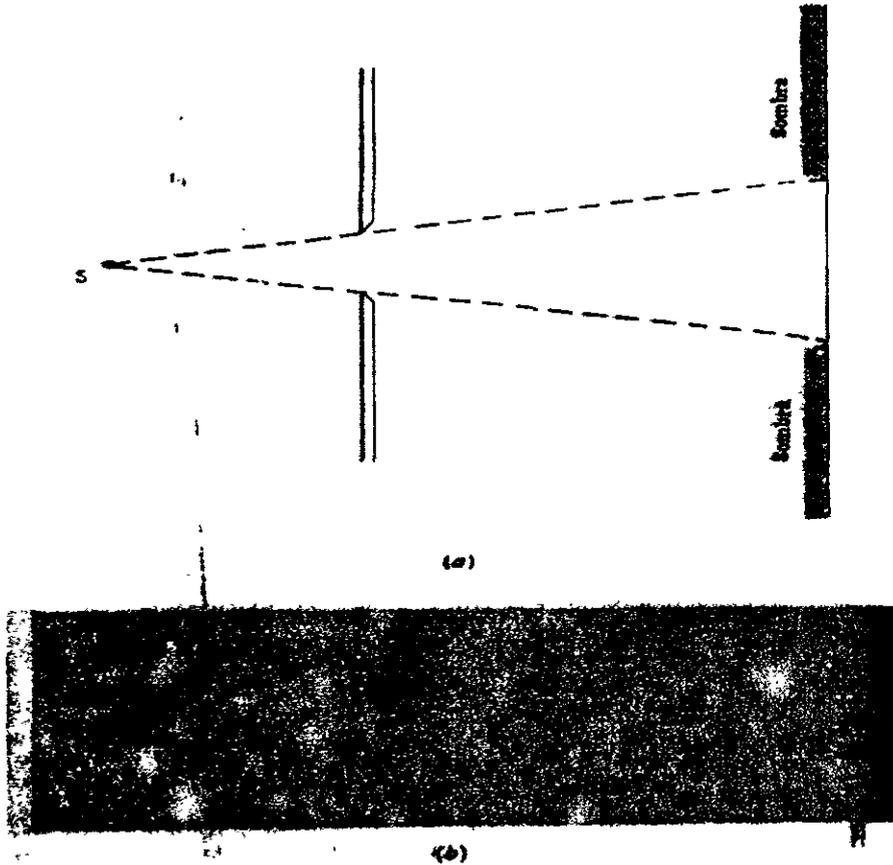


Figura 1. 20 (a) Sombreado producido por una ranura ancha. (b) Difracción producida por una ranura angosta

Los patrones de difracción constan de bandas claras y oscuras. Mediante el análisis de estos patrones podemos aprender acerca del objeto difractante. Por ejemplo, la difracción de rayos X es un método importante en el estudio de la estructura de sólidos, y la difracción de los rayos gamma se usa para estudiar los núcleos. La difracción tiene también efectos indeseables, como el esparcimiento de la luz al entrar en la abertura de un telescopio, lo cual limita su capacidad para resolver o separar las estrellas que se ven una cerca de la otra.

El principio fundamental en que se basa el conocimiento de la difracción es el principio de Huygen que indica que todos los puntos del frente de onda se pueden considerar como fuentes secundarias de las que se propagan las ondas en todas direcciones. Si consideramos el frente de onda sobre el objeto opaco en la figura I.18 podemos ver que las ondas secundarias se pueden propagar hacia la región de la sombra.

#### **I.4.A. DIFRACCIÓN EN UNA SOLA RANURA.**

Uno de los efectos más gráficos de los efectos de la difracción es el que se presenta en una sola ranura. Si hacemos que la luz de una fuente puntual pase por una ranura de ancho variable como se indica en la figura I.20, una abertura amplia, (a) da lugar a un sombreado simple. A medida que el ancho de la ranura se reduce, la región de sombra adquiere una apariencia de franjas, (b). El haz después de pasar por la ranura se extiende produciendo el patrón de difracción que presenta una banda central brillante, seguida de franjas alternadamente brillantes y oscuras, con intensidad que disminuye con rapidez. Fácilmente podemos ver este efecto observando una fuente luminosa entre los

dedos por una abertura que se hace progresivamente más y más angosta.

Se puede notar fácilmente por que ocurre esto tomando como base el principio de Huygen. La figura 1.21 muestra la ranura y las ondas secundarias que se pueden considerar como si se propagaran a partir de cada punto del frente de onda en la ranura. La intensidad en cada punto de la pantalla se determina entonces por la superposición de las ondas que llegan desde las fuentes secundarias. Por supuesto que habrá diferencias de fase debidas a las diferentes distancias desde cada una de las fuentes secundarias a cada punto de la pantalla. Además, el desarrollo matemático del principio de Huygen lleva un *factor de oblicuidad* o *factor de inclinación*, que hace que las ondas secundarias, que se propagan en un ángulo  $\theta$  con respecto a la onda original, tengan una intensidad proporcional a  $\frac{1}{2}(1 + \cos\theta)$ . Este factor indica un máximo de intensidad para la onda que se propaga en la dirección de la onda original, y cero para las ondas que se propagan en un ángulo recto con respecto a dicha onda original. Este factor de oblicuidad es el que causa el rápido decrecimiento de la intensidad en el patrón de difracción a medida que se aleja del *máximo central*. La intensidad de la franja brillante adyacente a la central es de 0.05 de la que presenta la más brillante. El primer límite corresponde al caso en que los rayos que llegan a la pantalla son paraxiales. Este caso se presenta cuando la pantalla se encuentra relativamente alejada de la ranura, o cuando la ranura es muy angosta. Esto se conoce como *difracción de Fraunhofer*. El segundo límite se presenta cuando la ranura es relativamente ancha o la pantalla de observación está cerca de la ranura, por lo que la suposición paraxial no es aplicable. Este caso se llama *difracción de Fresnel*. La

difracción de Fraunhofer también se presenta cuando se coloca una lente entre la ranura y la pantalla para cambiar la convergencia de la luz de la ranura y que pase paralela por ella. En la figura 1.22 se muestra otra representación de la figura 1.21 en la forma de diagrama de rayos. El patrón formado en la pantalla depende de la separación entre la pantalla C y la abertura B. En general, podemos considerar tres casos, dos de los cuales ya mencionamos arriba:

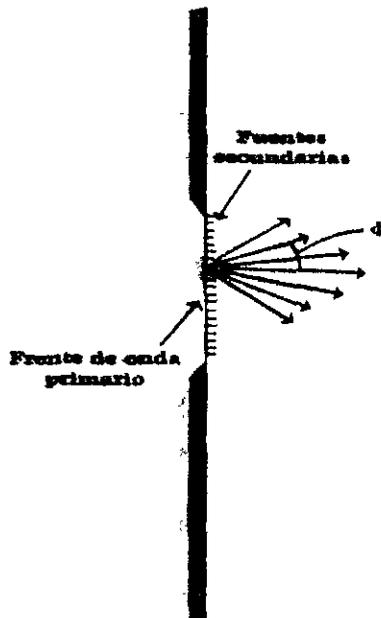


Figura 1. 21 Difracción en una ranura mostrando el frente de onda primario , las fuentes secundarias y los rayos que provienen de una de las fuentes secundarias. El ángulo  $\theta$  es el ángulo que aparece en el factor de oblicuidad

1. **Separacion muy pequeña.** Cuando  $C$  esta muy cerca de  $E$ , las ondas viajan unicamente una distancia corta despues de salir de la abertura, y los rayos divergen muy poco. Los efectos de la difraccion son despreciables, y el patron en la pantalla es la sombra geometrica de la abertura.

2. **Separacion muy grande.** La figura 1.21 (a) representa la situacion cuando la pantalla esta tan lejos de la abertura que podemos ver los rayos como paralelos o, lo que es lo mismo, a los frentes de onda como planos. (En este caso, suponemos tambien que la fuente esta lejos de la abertura, de modo que los frentes de onda incidentes son tambien planos. Puede conseguirse el mismo efecto si la abertura se ilumina con un laser). Una manera de conseguir esta condicior., la cual se conoce como difraccion de Fraunhofer, en el laboratorio es usar dos lentes convergentes, como se muestra en la figura 1.22.(b). La primera lente convierte la luz divergente de la fuente en una onda plana, y la segunda lente enfoca a las ondas planas que salen de la abertura en el punto  $P$ . Todos los rayos que llegan a  $P$  salen de la abertura paralelos a la linea de trazos  $Px$  trazada desde  $F$  a traves del centro de la segunda lente.

3. **Separacion intermedia.** En el caso mostrado en la figura 1.21, la pantalla puede estar a cualquier distancia de la abertura, y los rayos que entran y salen de la abertura no son paralelos. Este caso se llama difraccion de Fresnel.

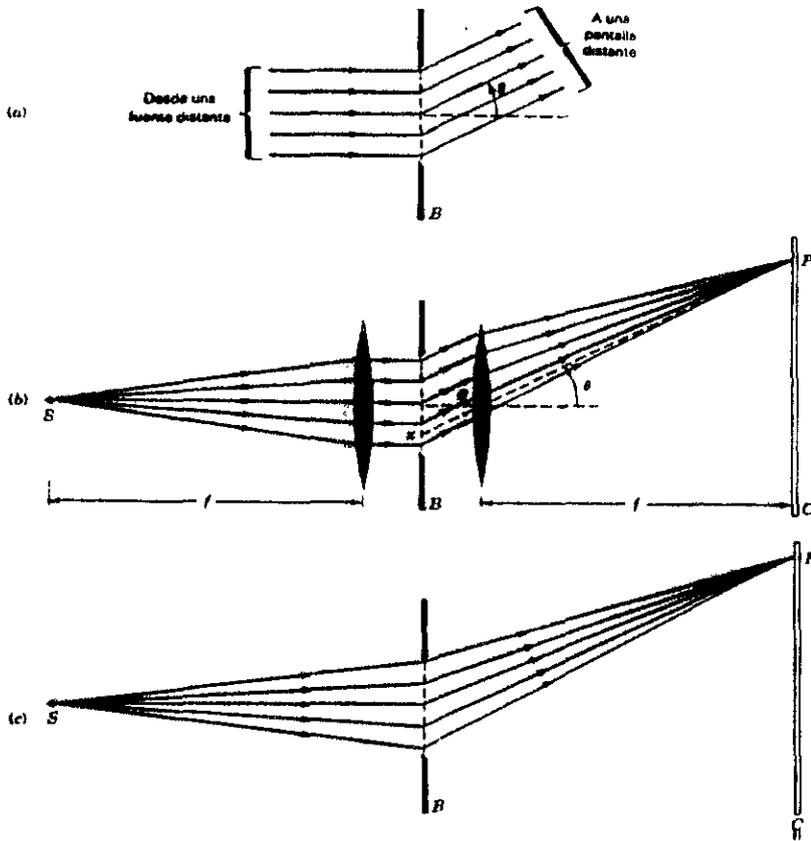


Figura I. 22 La luz de la fuente puntual  $S$  ilumina una rendija en la barrera opaca  $B$ .

La rendija se extiende a una gran distancia arriba y abajo del plano de la figura; esta distancia es mucho mayor que la anchura  $a$  de la rendija. La intensidad en el punto  $P$  de la pantalla  $C$  depende de las fases relativas de la luz recibida desde diversas partes de la rendija. (a) Si la fuente  $S$  y la pantalla  $C$  se mueven a grandes distancia de la rendija, tanto la luz incidente

en E como la emergente de (B) consisten en rayos casi paralelos.  
(c). En lugar de usar distancias grandes, la fuente y la pantalla pueden ser situadas cada una en el plano focal de una lente; una vez mas, en la rendija entran y salen rayos de luz paralelos. (d) sin la lente, los rayos no son paralelos.

## II CARACTERISTICAS DEL PROBLEMA

En la introducción ya se mencionó la importancia y los alcances de la colorimetría, así como sus aplicaciones recientes, por lo que no abundaremos sobre esos temas en esta sección.

Podemos describir nuestro problema como de tipo práctico, si bien para resolverlo satisfactoriamente es indispensable recurrir a la teoría. Es necesario tener presentes los siguientes conocimientos: teoría del muestreo (teorema de Nyquist), sistemas de transmisión de datos, conceptos básicos de programación, electrónica analógica y digital y fundamentos de óptica aplicada.

La idea original puede resumirse en esta pregunta: ¿cómo podemos detectar el color que caracteriza a cierto producto en una línea de producción? Algunas veces, cuando nos encontramos en tiendas departamentales, supermercados, farmacias o cuando acudimos a una tienda de pinturas a solicitar un color, podemos ubicar el producto que estamos buscando por alguna marca o color especial. En algunas ocasiones, es más fácil identificar objetos por medio de colores en lugar de letreros con textos, ya que la tonalidad del color puede expresar mayor información de manera inmediata y nos permite tomar una decisión.

Este proyecto de tesis gira en torno a esta cuestión. Lo que se pretende es identificar un producto cualquiera en una línea de producción de forma inmediata. En nuestro caso, se debe de detectar el color de la caja para realizar una elección.

- **Software actualizado.** Se buscó desarrollar un software con los elementos y herramientas que se utilizan mundialmente (tecnología de punta para los lenguajes de programación). Esto facilita la transmisión de información en ámbitos locales y remotos.
- **Tecnología de punta en hardware.** La electrónica de lógica discreta y la electrónica analógica discreta son muy complicadas en su diseño y desarrollo cuando se utilizan varios elementos activos o pasivos y componentes digitales. Afortunadamente, existen circuitos integrados que funcionan de modo eficiente y económico. Estos circuitos son los microprocesadores, que pueden programarse de acuerdo a las características de nuestros problemas. Debemos elegir un microprocesador y sus elementos periféricos tomando en cuenta la naturaleza de nuestro problema, con el fin de alcanzar un funcionamiento adecuado al proyecto previamente diseñado.
- **Convertidores analógicos a digitales.** Resulta de gran importancia para nuestra investigación la digitalización de una señal analógica, así que se deben considerar todas las teorías sobre el muestreo de señales y las características de convertidores analógicos a digitales para tener una perfecta digitalización de la señal.

### III INTRODUCCIÓN AL DISEÑO DEL HARDWARE:

En el diseño y construcción del Sistema de Adquisición de Datos se tomaron en cuenta, para su buen desempeño, las siguientes características: costo, consumo de potencia, velocidad y comportamiento a variaciones de temperatura e iluminación.

Podemos dividir el Sistema de Adquisición de Datos en dos partes. A la primera le llamaremos propiamente Sistema de Adquisición de Datos (SAD).

En los diseños de ingeniería se busca principalmente el buen funcionamiento de los sistemas valorando todas las opciones y escogiendo la más adecuada para llevar a cabo el diseño y finalmente la construcción del sistema. No siempre resulta ser que problemas complejos se resuelvan con soluciones complejas; por el contrario, la ingeniería busca dar soluciones sencillas pero funcionales a problemas complejos.

El Sistema de Adquisición de Datos (SAD), es la parte más importante del sistema, ya que en ella se realiza la conversión de la señal analógica a digital. Consiste principalmente de un microcontrolador PIC 17C44 de alto desempeño y de 8 bits fabricado por MicroChip Technology. Se utilizó este microcontrolador porque ofrece características más convenientes en el diseño de la tarjeta de Adquisición de Datos (SAD). Una de sus principales cualidades es la velocidad de operación (33 MHz), que resulta suficiente para el muestreo de la señal que se pretende digitalizar (6.2 MHz).

Así mismo, se emplearon dispositivos electrónicos integrados de rápida respuesta como el buffer de tres estados 74f126, el inversor 74F04, el arreglo en cascada de los contadores 74191, la memoria HM66205 y el convertidor ADS901A, que se describirán más detalladamente conforme se desarrolle este tema.

A la segunda parte la nombraremos Señal de Control (SC). La señal de control le indica a la tarjeta SAD cuándo comienza la digitalización y cuándo termina. Consiste en el diseño de un circuito detector de ventana con histéresis y un transistor de respuesta rápida. El circuito integrado fundamental para esta etapa es un Amplificador Operacional Comparador LM311 y un transistor 2N2222A, que tienen un comportamiento excelente en el control de la señal. Se debe mencionar que los elementos tanto activos como pasivos tienen tecnología de especificaciones militares, ya que al trabajar en frecuencias altas se presentan inductancias y capacitancias parásitas que influyen en forma considerable en la respuesta del sistema; los componentes militares son más estables a señales de frecuencias altas.

Al conocer lo anterior podemos describir *grosso modo* el sistema. La cámara CCD tiene el formato estándar 1 Vpp 75 w EIA RS-330 a una velocidad de pixel de 6.25 Mhz. Lo que se pretende es digitalizar la señal analógica de la cámara a una velocidad de muestreo de 18.74MHz, con lo que aseguramos que no estamos violando el criterio de Nyquist sobre el teorema del muestreo de señales, ya que se está alcanzando prácticamente tres veces la velocidad de la señal a muestrear. Más adelante, por medio de algoritmos de software, se aproximará la curva de la señal que se obtiene para poder caracterizar los colores en tonalidades de grises, y así poder definir la imagen y el color que está "viendo" la cámara.

Como se mencionó anteriormente, las características electrónicas que se deben cumplir para obtener una buena adquisición de datos son las siguientes:

- Alta velocidad de conversión de Analógica a Digital
- Alta velocidad de muestreo.
- Espacio suficiente de memoria para la adquisición de los datos
- Transmisión Serial a una PC para poder visualizar y verificar los datos de la conversión.

El diagrama a bloques muestra las etapas de la tarjeta de adquisición de datos:

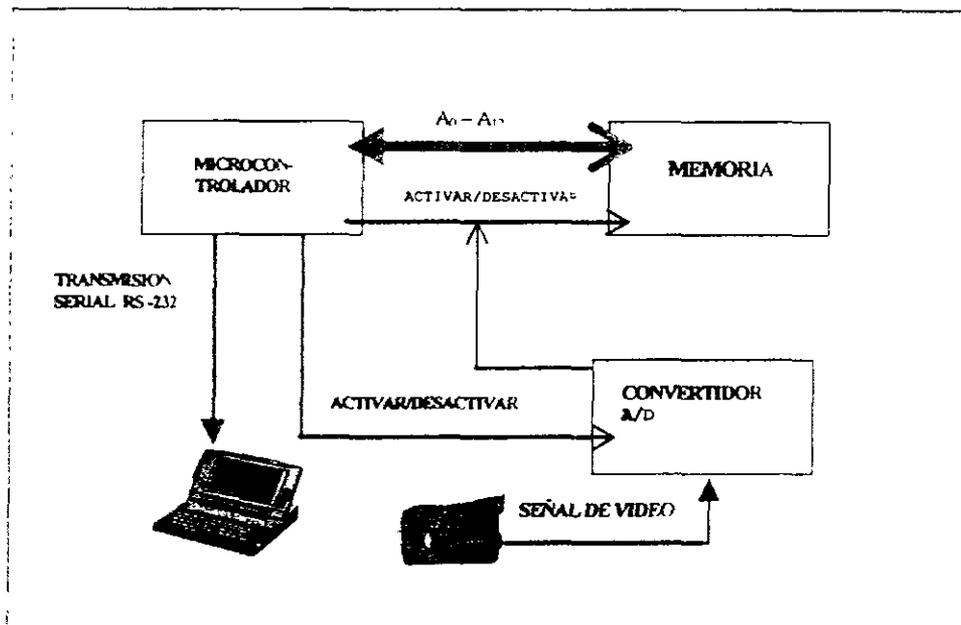


Figura III. 1 Diagrama a bloques del sistema.

A continuación se dará una breve explicación del diagrama a bloques. Lo que se pretende es digitalizar de forma rápida y

eficiente la señal de video analogica. Para poderla manipular y tener la seguridad de que todos los elementos y dispositivos electrónicos actuarán de forma conjunta y especifica, utilizamos un microcontrolador de tecnologia RISC. La característica mas importante de este elemento es la velocidad de procesamiento, a diferencia de un microcontrolador estándar de Von Neuman, ya que este microcontrolador nos permite una velocidad maxima de ejecucion de 33 MHz. El microcontrolador se encarga de habilitar e inhibir las funciones tanto del convertidor como de la memoria, para evitar conflictos en el bus de recepcion y transmision de datos de un elemento a otro.

Pero ¿que funciones tiene la tarjeta SAD? Por medio del microcontrolador la tarjeta se programa con el fin de que, al aplicarle una señal de restablecimiento, habilite al convertidor para indicarle cuándo comienza la digitalización. El convertidor transfiere los datos digitalizados a la memoria externa. El microcontrolador observa el estado de la memoria para saber si ésta se encuentra totalmente ocupada por los datos. Una vez que el microcontrolador verifica que la memoria este llena, empieza a extraer los datos para enviarlos en forma serial (RS - 232) al equipo que finalmente podrá manejar esta información (PC).

Se mencionó también un sistema de control de señal, que es básicamente un detector de ventana con histéresis y proporciona la activación de la señal de reloj del convertidor para indicarle a éste cuándo comenzar la digitalizacion y, de esta manera, obtener de forma sencilla la señal de video sin el pulso de sincronía. Esta característica se explicara de forma mas detallada en el diseño de la etapa de control.

Para poder diseñar la tarjeta SAD y la etapa de control es bueno conocer las características de los elementos a utilizar. Comencemos por conocer al microcontrolador PIC17C44.

### **III.1. HARDWARE. ARQUITECTURA DEL MICROCONTROLADOR PIC17C44**

#### **CARACTERÍSTICAS PRINCIPALES DEL MICROCONTROLADOR**

- Únicamente consta de 58 instrucciones para programarlo.
- Todas las instrucciones sencillas son de un ciclo (121 ns). Excepto para bifurcaciones de programa y tablas de lectura/escritura, que son de dos ciclos.
- Velocidad de operación:
  - DC entrada de reloj: 33 MHz.
  - DC ciclo de instrucción: 121 ns.
- Capacidad de interrupción.
- 16 niveles de la pila (stack) del hardware.
- Modos de direccionamiento: directo, indirecto y relativo.
- Ejecución del programa en memoria interna/externa.
- Espacio de memoria programable: 64Kx16.

#### **CARACTERÍSTICAS PERIFÉRICAS.**

- 33 pines I/O con control de dirección individual.
- Manejo de alta corriente caída/suministro (sink/source) para controlar directamente a los LED's.  
RA2 y RA3 son compuertas abiertas de alto voltaje (12 Volts) y alta corriente (60 mA,, I/C).

- Dos entradas de captura y dos salidas de PWM. Son capturas de 16-bits, resolución máxima de 160 ns. La resolución de PWM es de 1 a 10 bits.
- TMR0: tiempo programable / contador de 16 bits (timer/counter, con 8 bits de programación preescalable.
- TMR1: 8 bits timer/counter.
- TMR2: 8 bits timer/counter.
- TMR3: 16 bits timer/counter.

### **CARACTERÍSTICAS ESPECIALES DEL MICROCONTROLADOR**

- Energía de Restablecimiento (Power - on reset) POR, energía de tiempo programable (Power - up timer) OST.
- Perro de guardia de tiempo programable (Watch dog) WDT. El chip cuenta con un oscilador RC para una operación segura.
- Protección de código.
- Modo de ahorro de energía de espera "sueño" (SLEEP).
- Opciones de seleccionar al oscilador.

### **TECNOLOGIA CMOS.**

- Bajo consumo de potencia, alta velocidad, tecnología EPROM/ROM.
- Diseño completamente estático.
- Amplio rango de operación de voltaje (2.5 a 6.0 Volts).
- Rango de temperatura industrial y comercial.
- Bajo consumo de potencia.
  - < 5 mA @ 5V, 4 MHz.
  - 100 mA típico @ 4.5 V, 32 kHz.
  - < 1 mA en estado de espera corriente @ 5V.

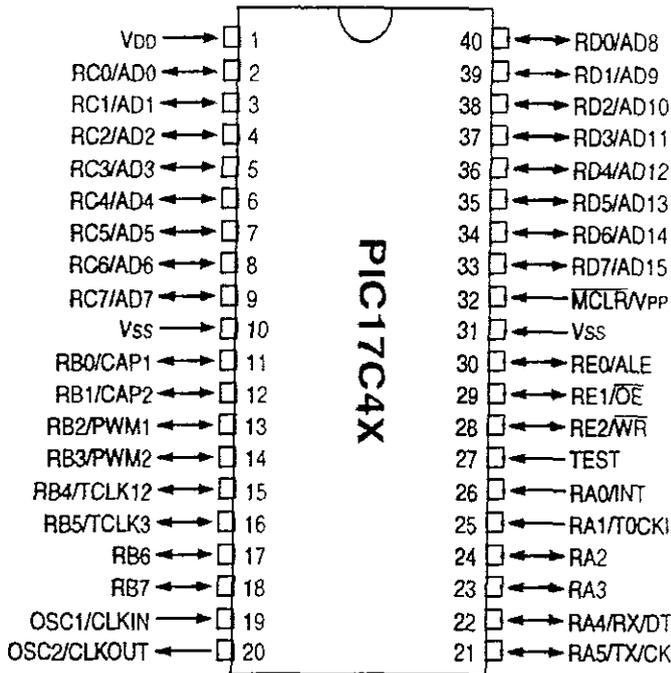


Figura III. 2 Diagrama de pines del PIC17C4X

### III.1.A. PIC17C44.

El PIC17C44 es un microcontrolador que emplea una avanzada arquitectura RISC (Reduce Instruction Set Code). El PIC17C44 tiene un aumento en sus características, 16 niveles en la parte inferior de la pila, múltiples fuentes de interrupción internas y externas. Las instrucciones separadas y los buses de datos con arquitectura Harvard de 16 bits ofrecen un amplio rango de palabras de instrucciones separadas de 8 bits de datos para su procesamiento rápido y eficaz.

Las dos etapas de instrucciones de entubamiento (pipeline) siguen todas las instrucciones para ejecutarlas en un solo ciclo, excepto para las derivaciones de programas (que requieren dos ciclos). Existe un total de 58 instrucciones (se reduce el set de instrucciones), ya que se introduce un registro de mayor capacidad, que es el que contiene al set. Por esta innovación se obtiene un mejor desempeño. Para aplicaciones puramente matemáticas tiene un multiplicador de hardware de 8x8 de ciclo sencillo.

En este microcontrolador se incluyen:

- Cuatro timer / counters.
- Dos entradas de captura
- Dos salidas de PWM
- Un Transmisor Receptor Asíncrono Síncrono Universal (USART).

Estas características especiales reducen los componentes externos. De esta manera se reduce el costo: aumentando la seguridad del sistema y reduciendo el consumo de potencia.

Existen cuatro opciones de osciladores. El más sencillo y económico es el que se realiza en el pin RC. El oscilador LF es para cristales de baja frecuencia que minimizan el consumo de potencia. XT es un cristal estándar y el EC es una entrada de reloj externa. El modo de "sueño" (sleep, power down), ofrece un ahorro de energía adicional. Para "despertar" al chip del modo de "sueño" se pueden realizar diferentes interrupciones, tanto externas como internas, así como diversos dispositivos que proporcionen el restablecimiento del microcontrolador.

Hay cuatro modos de configuración para operar este dispositivo:

- Microprocesador.
- Macrocontrolador.
- Microcontrolador Extendido.
- Microcontrolador Protegido.

Los modos de Microprocesador y Microcontrolador Extendido permiten memorias programables superiores a 64 K.

Un watch dog timer altamente confiable, que se encuentra dentro del chip en el pin del oscilador RC, proporciona protección contra el mal funcionamiento del software.

La tabla III.1 enlista las características de los dispositivos PIC17C4X. Los empaquetados Cerdip pueden ser borrados por rayos ultravioleta y son ideales para el desarrollo de códigos mientras se encuentren en prueba. Posteriormente, pueden ser sustituidos por una versión de dispositivo programable solamente una vez (One-time programmable, OTP), ya que el costo se reduce para una producción de gran volumen.

Features	PIC17C42	PIC17CR42	PIC17C42A	PIC17C43	PIC17CR43	PIC17C44
Maximum Frequency of Operation	25 MHz	33 MHz	33 MHz	33 MHz	33 MHz	33 MHz
Operating Voltage Range	4.5 - 5.5V	2.5 - 6.0V				
Program Memory x16	(EPROM)	2K	2K	4K	-	8K
	(FROM)	-	2K	-	4K	-
Data Memory (bytes)	232	232	232	454	454	454
Hardware Multiplier (8 x 8)	-	Yes	Yes	Yes	Yes	Yes
Timer0 (16-bit + 8-bit postscaler)	Yes	Yes	Yes	Yes	Yes	Yes
Timer1 (8-bit)	Yes	Yes	Yes	Yes	Yes	Yes
Timer2 (8-bit)	Yes	Yes	Yes	Yes	Yes	Yes
Timer3 (16-bit)	Yes	Yes	Yes	Yes	Yes	Yes
Capture inputs (16-bit)	2	2	2	2	2	2
PWM outputs (up to 10-bit)	2	2	2	2	2	2
USART/SCI	Yes	Yes	Yes	Yes	Yes	Yes
Power-on Reset	Yes	Yes	Yes	Yes	Yes	Yes
Watchdog Timer	Yes	Yes	Yes	Yes	Yes	Yes
External Interrupts	Yes	Yes	Yes	Yes	Yes	Yes
Interrupt Sources	11	11	11	11	11	11
Program Memory Code Protect	Yes	Yes	Yes	Yes	Yes	Yes
I/O Pins	33	33	33	33	33	33
I/O High Current Capability	Source	25 mA				
	Sink	25 mA <sup>(1)</sup>				
Package Types	40-pin DIP	40-pin DIP	40-pin DIP	40-pin DIP	40-pin DIP	40-pin DIP
	44-pin PLCC	44-pin PLCC	44-pin PLCC	44-pin PLCC	44-pin PLCC	44-pin PLCC
	44-pin MQFP	44-pin MQFP	44-pin MQFP	44-pin MQFP	44-pin MQFP	44-pin MQFP
	44-pin TQFP	44-pin TQFP	44-pin TQFP	44-pin TQFP	44-pin TQFP	44-pin TQFP

Note 1. Pins RA2 and RA3 can sink up to 80 mA.

Tabla III. 1 FAMILIA DE DISPOSITIVOS DEL PIC17CXX.

### III.1.B. ARQUITECTURA DEL PIC17C44.

La alta funcionalidad del PIC17C44 se puede atribuir a las numerosas características comunes encontradas en la arquitectura RISC de microprocesadores. El PIC17C44 utiliza una arquitectura Harvard modificada. Esta arquitectura tiene el acceso de datos y programas en memorias separadas. De esta manera, el dispositivo tiene un bus para la memoria del programa y otro bus para la memoria de datos. Esto proporciona un procesamiento superior a la arquitectura tradicional de Von Neuman, donde los datos y el programa son puestos en un solo intervalo de búsqueda (fetched)

de la misma memoria (Acceso sobre el mismo bus). El programa y los datos de la memoria se separan con el fin de establecer primero la diferencia de una palabra de 8 bits.

El tamaño de los códigos de operación del PIC17C44 es de 16 bits; su habilitación es simple y emplea instrucciones de palabras sencillas. El rango completo del bus de la memoria del programa es de 16 bits en el ciclo de espera y búsqueda (fetched). En instrucciones sencillas, puede manejar 16 bits. Dos estados de entubamiento (pipeline) están traslapados al ciclo de búsqueda, espera y ejecución de la instrucción. Consecuentemente, todas las instrucciones son ejecutadas en un ciclo sencillo de 12ns @ 33 MHz, excepto las bifurcaciones del programa y dos instrucciones especiales, que son la transferencia de los datos entre el programa y los datos de memoria.

El PIC17C44 integra 8k x 16 bits de EPROM de la memoria del programa. La ejecución del programa puede ser únicamente interna, que maneja el modo de microcontrolador o el microcontrolador protegido, o bien incluirlos a ambos (microcontrolador en modo extendido). El microcontrolador en modo extendido no permite el código de protección. El PIC17C44 puede direccionar, directa o indirectamente, a los archivos de registro y/o la memoria de datos. Todos los registros de funciones especiales, incluyendo el contador del programa (PC) y el registro de trabajo (WREG,, están almacenados en la memoria de datos. El PIC17C44 tiene instrucciones ortogonales (simétricas). El set de instrucciones está diseñado para desarrollar operaciones en cualquier registro y modo de direccionamiento. Su naturaleza simétrica y carente de "situaciones especiales para optimización" hace que la programación del PIC17C44 sea eficiente y sencilla. En suma, el aprendizaje del dispositivo es más simple.

El diseño de la arquitectura de los PIC'S permite que los dos archivos de registros sean usados en algunas instrucciones por los operadores. Esto permite que los datos sean movidos directamente entre dos registros sin pasar a través del registro WREG. De esta forma, su desempeño mejora y disminuye el uso de memoria del programa.

El PIC17C44 contiene una Unidad Lógica Aritmética (ALU) de 8 bits y un registro de trabajo. La ALU es una unidad aritmética de propósito general. El desarrollo aritmético y las funciones booleanas de los datos se desarrollan entre el registro de trabajo y el registro de archivos.

Esta ALU de 8 bits es capaz de sumar, sustraer, cambiar de signo y realizar operaciones lógicas, además de otras operaciones de tipo aritmético o algún tipo de complemento.

El WREG es un registro de trabajo de 8 bits empleado para las operaciones de la ALU. La mayoría de los PICS (17CXX) tiene un multiplicador de Hardware de 8X8. Este multiplicador produce un resultado de 16 bits en un ciclo sencillo.

Dependiendo de la instrucción ejecutada, la ALU puede afectar los valores de "acarreo" (CARRY, C), dígito de acarreo (DIGIT CARRY, DC), el bit cero (Z) y el registro de estado (STATUS). Los bits de C y DC operan como un indicador de "acarreo negativo", esto es, un dígito producido en sustracción directa cuando la diferencia entre los dígitos es menor que cero.

En la ALU se desarrolla la asignación matemática. El bit de sobreflujo (OV) se puede utilizar para designar el signo matemático de los resultados. La asignación aritmética comprende

la magnitud y el signo del bit. El bit de sobreflujo indica la magnitud de sobreflujo y las causas de cambio en el bit de signo.

El signo matemático puede tener valores mayores que 7 bits magnitud, si es que se usa más de un bit. El uso del bit de sobreflujo opera en el bit 6 (MBS de magnitud) y bit 7 (bit de signo). El bit de sobreflujo no se utiliza si se está procesando la magnitud y el signo matemático, por ejemplo 11 bits. Si el valor del signo matemático es mayor que 7 bits (15, 24 o 31 bits), se puede asegurar que al realizar el algoritmo los bits de orden menor se ignoran por el estado del bit de sobreflujo.

Se deberá tener cuidado cuando se sumen y se sustraigan números signados, para asegurar que se realicen correctamente las operaciones. En el ejemplo III.1 se muestra detalladamente cómo se deben tomar en cuenta esos números cuando se realiza una asignación aritmética en la ALU, que opera como una máquina sin signo.

### Ejemplo III.1

Valor hexadecimal	Valor matemático Signado	Valor matemático sin signo
FFh	-127	255
+ 01h	+ 1	+ 1
= ?	= -126 (Feh)	= 0 (00h)
		Bit de acarreo = 1

El resultado en el REG requiere del signo matemático para hacer FE h (-126). Esto es, deberá realizarse para el complemento a uno.



Tabla III. 2 DESCRIPCION DE LOS PINES DE SALIDA.

NOMBRE	NUMERO DE PIN	TIPO I/O/P	TIPO DE BUFFER	DESCRIPCION
OSC1/CLKIN	1		ST	Entrada de oscilador en el modo de cristal/resonador u oscilador RC. Modo de entrada de reloj externo.
OSC2/CLKOUT	2			Salida de oscilador. Conexión para el cristal/resonador en el modo de cristal/resonador. En el modo de oscilador RC o reloj externo OSC2 es pin de salida CLKOUT que es la cuarta parte de la frecuencia de oscilación OSC1 y denota la rapidez de ciclo de instrucción.
MRST/RES	3	I/O	ST	Limpieza maestro (reestablecimiento) entrada de Voltaje V <sub>DD</sub> . Este activa al Reset en el chip al tener un nivel bajo en el pin de entrada.
				PUERTO A (PORTA) es bidireccional, excepto para RA0 y RA1 que son entradas únicamente.
RA2/INT	4		ST	RA2/INT puede seleccionarse como una entrada de interrupción externa, puede ser configurada con borde positivo o negativo.
RA1/TCK1	23		ST	RA1/TCK1 puede ser también seleccionada como entrada de interrupción externa, puede ser configurada con borde positivo o negativo. RA1/TCK1 puede ser también seleccionada como reloj para el timer programable TIMER0 timer/counter.
RA3	24	I/O	ST	Alto voltaje, alta corriente, pin de compuerta abierta, entrada / salida del puerto.
RA2	25	I/O	ST	Alto voltaje, alta corriente, pin de compuerta abierta, entrada / salida del puerto.
RA4/RX/D+	26	I/O	ST	RA4/RX/D+ pueden ser seleccionados como el USART (SCI) Receptor Asíncrono o USART (SCI) datos síncronos.
RA5/TX/C-	27	I/O	ST	RA5/TX/C- puede ser seleccionado como el USART (SCI) Transmisor Asíncrono o USART (SCI) Reloj síncrono.
				Puerto B (PORTB) es un puerto bidireccional de I/O configurable por software, necesita "pull - up".
RB0/CAP1	11	I/O	ST	RB0/CAP1 El pin puede ser también la entrada CAP1.
RB1/CAP2	12	I/O	ST	RB1/CAP2 El pin puede ser también la entrada CAP2.
RB2/PWM1	13	I/O	ST	RB2/PWM1 El pin puede ser también la salida PWM1.
RB3/PWM2	14	I/O	ST	RB3/PWM2 El pin puede ser también la salida PWM2.
RB4/TCLK12	15	I/O	ST	RB4/TCLK12 puede ser también la entrada de reloj externa para el TIMER1 y TIMER2.

RB5/TCLK5	16	I/O	ST	RB5/TCLK5 puede ser también la entrada del reloj externo para el timer.
RBF	17	I/O	ST	
RP	18	I/O	ST	
				<p>PUERTO C (PORT) Es un puerto bidireccional. Es la parte media baja del sistema del bus de 16 bits en el modo de microprocesador o microcontrolador extendido. En la configuración como sistema multiplexado, estos pines son direccionados como salida, dan un espacio a los datos de entrada y salida.</p>
RC0/AD0	2	I/O	TTL	
RC1/AD1	3	I/O	TTL	
RC2/AD2	4	I/O	TTL	
RC3/AD3	5	I/O	TTL	
RC4/AD4	6	I/O	TTL	
RC5/AD5	7	I/O	TTL	
RC6/AD6	8	I/O	TTL	
RC7/AD7	9	I/O	TTL	
				<p>El PUERTO D (PORTD) es un puerto bidireccional de 16 bits. Este puerto también es la parte media superior del bus de 16 bits en el modo de microprocesador o modo de microcontrolador extendido. En la configuración como un sistema multiplexado, estos pines son direccionados como salida, dan un espacio a los datos de entrada y salida.</p>
RD0/AD8	40	I/O	TTL	
RD1/AD9	39	I/O	TTL	
RD2/AD10	38	I/O	TTL	
RD3/AD11	37	I/O	TTL	
RD4/AD12	36	I/O	TTL	
RD5/AD13	35	I/O	TTL	
RD6/AD14	34	I/O	TTL	
RD7/AD15	33	I/O	TTL	
				<p>PUERTO E (PORTE) es un puerto bidireccional de 16 bits.</p>
REQ/ALE	30	I/O	TTL	En el modo de microprocesador o microcontrolador extendido, esta es la dirección que habilita la salida del enclavamiento (Address Latch Enable (ALE)).
RE1/ $\overline{OE}$		I/O	TTL	En el modo de microprocesador o microcontrolador extendido, este es el habilitador de salida ( $\overline{OE}$ ) salida de control (se activa con nivel bajo).

RES/W-		I/O	TTL	En el modo de microprocesador o microcontrolador extendido, este habilita la escritura ( $\overline{WR}$ ) salida de control (se activa con nivel bajo)
TEC			ST	Modo de selección de entrada de prueba. Algunas veces asegura y/o detecta el V <sub>cc</sub> para una operación normal.
V <sub>ss</sub>	GND	F		Referencia de tierra para los pines de I/O lógicos
V <sub>cc</sub>		P		Entrada a la fuente de voltaje positiva lógica.

Leyendas. I= entrada únicamente; O= salida únicamente; I/O=Entrada/Salida; P = Energía; - = No se utiliza; TTL = entrada TTL; ST = Entrada de Schmitt Trigger.

### III.1.C. ESQUEMA DE RELOJ/ CICLO DE INSTRUCCIONES.

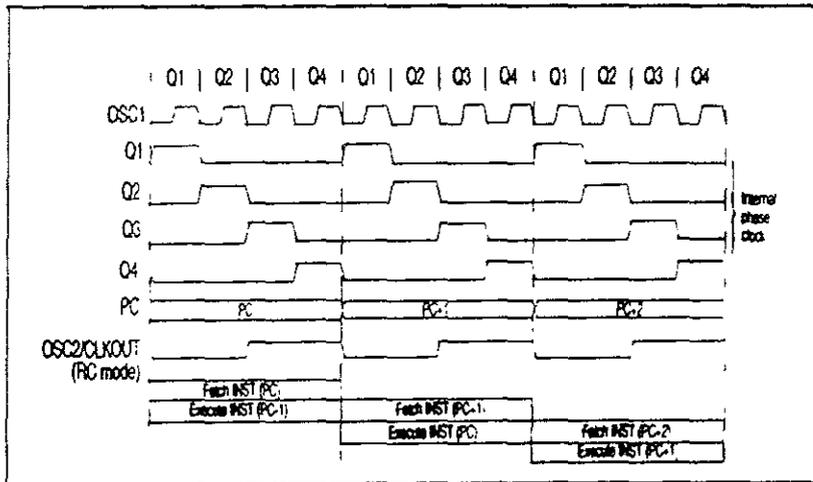


Figura III. 4 Ciclo de Reloj Esquema/Instrucciones

La entrada de reloj (de OSC1) esta dividida internamente entre cuatro, para generar cuatro señales de reloj que no se traslapen entre si, llamados Q1, Q2, Q3 y Q4. Internamente el contador del programa es incrementado (PC) por Q1 la instrucción comienza con el ciclo de búsqueda en la memoria del programa y la "captura" en

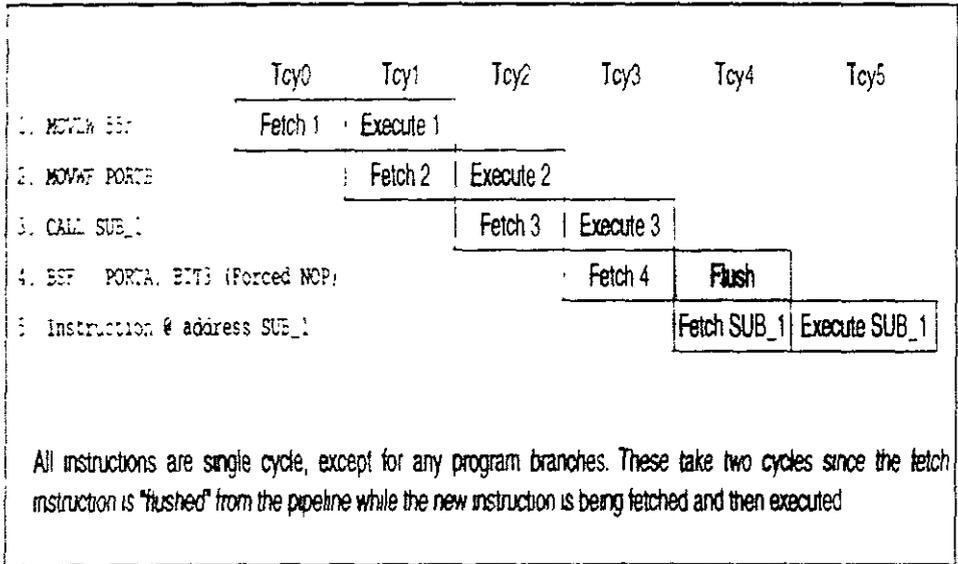
el registro de instrucción en Q4. La instrucción es decodificada y ejecutada de Q1 a Q4. Las señales de reloj y las instrucciones son ejecutadas como se muestra en la figura III.4

### **III.1.D. INSTRUCCIONES DE FLUJO/ENTUBAMIENTO (FLOW/PIPELINE) .**

Un "ciclo de instrucción" consiste de cuatro ciclos de Q (Q1, Q2, Q3 y Q4). La instrucción es buscada, ejecutada, y "entubada", al realizar la búsqueda, esto consume un ciclo, para la decodificación y ejecución se toma otro ciclo para cada una.

Por la característica de entubamiento efectivamente cada instrucción se realiza en un ciclo de instrucción. Si una instrucción origina un cambio en el Contador de Programa, entonces se requieren dos ciclos para completar la instrucción (ejemplo III.2).

Un ciclo de búsqueda comienza cuando el Contador del Programa se incrementa con Q1. En la ejecución del ciclo, cuando la instrucción es buscada, ésta es retenida en el Registro de Instrucciones (IR) en el ciclo Q1. Entonces esta instrucción es decodificada y ejecutada durante los ciclos Q2, Q3 y Q4. Los datos de memoria son leídos durante Q2 (operación de lectura) y el de escritura en Q4 (Destino de escritura).



Ejemplo III. 2 Flujo de entubamiento de la instrucción.

### III.1.E. RESTABLECIMIENTO (RESET).

Las diferencias entre los varios tipos de (RESET),

- Restablecimiento por energía (POR).
- $\overline{MCLR}$  Restablecimiento durante la operación normal.
- WDT Restablecimiento (operación normal):.

Algunos registros no son afectados por la condición de Restablecimiento, su condición o estado es desconocida al aplicarse el POR, y no cambia con cualquier otro tipo de RESET. La mayoría de los registros son obligados o forzados a tomar la condición de estado de restablecimiento al utilizar él POR,  $\overline{MCLR}$

o WDT cuando se programa al PIC para que se encuentre en la condición de "sueño". Estos no son afectados por WDT durante el estado de "sueño", entonces este restablecimiento es visto como el comienzo de la operación normal. Los bits  $\overline{TO}$  y  $\overline{PD}$  son activados o desactivados dependiendo de las diferentes situaciones de restablecimiento, como se muestra en la tabla III.1.5 Estos bits son usados por el software para determinar la naturaleza del restablecimiento. Se puede ver en la tabla III.1.6 una descripción más detallada de los estados de restablecimiento de todos los registros.

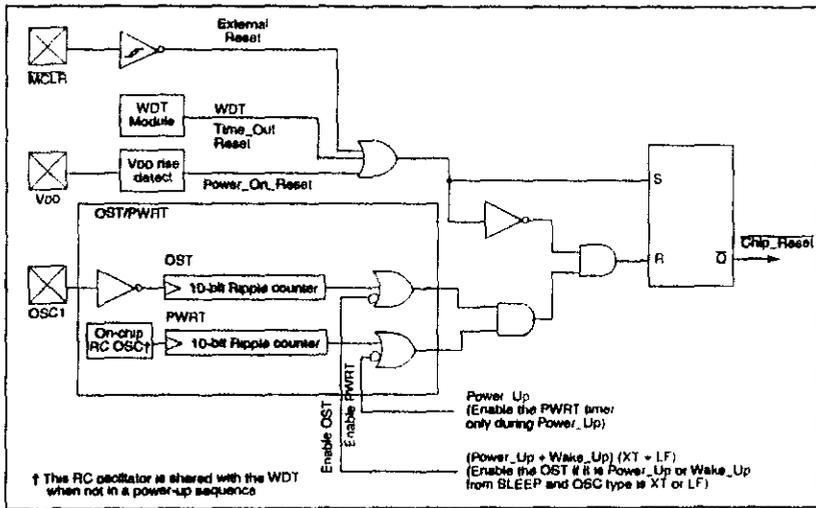


Figura III. 5 Diagrama a Bloques simplificado del circuito de reset del Chip.

NOTA: Cuando el dispositivo se encuentra en el estado de restablecimiento, la fase del reloj interno se detiene en el estado de Q1. Cualquier modo de microprocesador que permita una ejecución externa forzará al pin  $RE0/\overline{OE}$  como una salida baja y al pin  $RE2/\overline{WR}$  como una salida alta.

### III.1.E.1. RESTABLECIMIENTO (RESET).

#### POWER ON RESET (POR).

El circuito de POWER ON RESET mantiene al dispositivo en reset aproximadamente en  $V_{DD}$ , esto es, sobre el punto de nivel bajo (en el rango de 1.4. - 2.3 V). La mayoría de los PIC'S producen un reset interno para levantamiento o caída de  $V_{DD}$ . Para tener un buen sistema de reset es necesario colocar directamente en el pin de  $\overline{MCLR}/V_{PP}$  una configuración de una resistencia y un interruptor que garantice que tomara el valor de  $V_{DD}$ , y de esta manera evitar el uso de circuitos externos RC, para crear el reset de POWER ON RESET. Sólo se necesita un tiempo corto para dar un levantamiento sobre  $V_{DD}$ .

### III.1.E.2. POWER UP TIMER (PWRT).

El power up timer proporciona un tiempo fuera constante de 96ms (nominal) en power up. Esto ocurre al detectarse el levantamiento del borde de la señal del POR y del primer levantamiento del borde del  $\overline{MCLR}$  (detectado como nivel alto). El power up timer opera con un oscilador interno RC. El chip permanece en un reset

largo mientras el PWRT este activado. En muchos casos el PWRT permite un retraso hasta detectar el nivel de  $V_{DD}$  aceptable. El retraso del power up time, depende del tipo de chip, de su variación con respecto a la temperatura así como el nivel de  $V_{DD}$ .

### III.1.E.3. OSCILLATOR START UP TIMER (OST) .

El oscilador start up timer (OST) proporciona un ciclo de oscilación de 1024 (1024 TOSC) de atraso después de que el  $\overline{MCLR}$  es detectado como nivel alto o cuando se "despierta" al sistema del modo de "sueño".

El OST time out es activado únicamente cuando se opera en los modos de XT y Oscilador en Power on Reset o se "despierta" del estado de "sueño" al sistema.

El OST cuenta los pulsos del oscilador en el pin de OSC1/CLKIN. El contador comienza a incrementarse únicamente después de que la amplitud de la señal se iguala con la del pin de entrada. Este retraso permite, tanto al cristal como al resonador, estabilizarse antes de que en el dispositivo exista un reset. La duración del time out está en función de la frecuencia del cristal/resonador.

### III.1.E.4. SECUENCIA DE TIME OUT.

La secuencia de time out en POWER UP es como sigue: primero la señal interna del POR sigue en alto. Si el  $\overline{MCLR}$  está en alto, los timer's del OST y del WRT comienzan. En general, el tiempo fuera del PWRT es largo, excepto en cristales/resonadores de baja frecuencia. El tiempo total fuera depende también de la

configuración del oscilador. La tabla III.3 muestra los tiempos asociados con la configuración del oscilador. Las figuras III.5. y III.6. muestran las secuencias de tiempo fuera.

Oscillator Configuration	Power-up	Wake up from SLEEP	MCLR Reset
XT, LF	Greater of: 96 ms or 1024TOSC	1024TOSC	—
EC, RC	Greater of: 96 ms or 1024TOSC	—	—

Tabla III. 3 Tiempo fuera en varias situaciones

TO	PD	Event
1	1	Power-on Reset, MCLR Reset during normal operation, or CLRWDT instruction executed
1	0	MCLR Reset during SLEEP or interrupt wake-up from SLEEP
0	1	WDT Reset during normal operation
0	0	WDT Reset during SLEEP

Tabla III. 4 Estados de los bits y su significado.

Si el voltaje del dispositivo no está dentro de las especificaciones eléctricas al final del time out, el pin de  $\overline{MCLR}/VPP$  deberá mantener un nivel bajo hasta que el voltaje este dentro de las especificaciones del dispositivo.

El uso de un retraso externo RC es suficiente para cualquiera de estas aplicaciones. La secuencia del time out comienza del primer levantamiento del borde del  $\overline{MCLR}$ . La tabla III.5. muestra las condiciones de reset para algunos registros especiales.

Event		PCH:PCL	CPUSTA	OST Active
Power-on Reset		0000h	--11 11--	Yes
$\overline{MCLR}$ Reset during normal operation		0000h	--11 11--	No
$\overline{MCLR}$ Reset during SLEEP		0000h	--11 10--	Yes <sup>(2)</sup>
WDT Reset during normal operation		0000h	--11 01--	No
WDT Reset during SLEEP <sup>(3)</sup>		0000h	--11 00--	Yes <sup>(2)</sup>
Interrupt wake-up from SLEEP	GLINTD is set	PC + 1	--11 10--	Yes <sup>(2)</sup>
	GLINTD is clear	PC + 1 <sup>(1)</sup>	--10 10--	Yes <sup>(2)</sup>

Tabla III. 5 Condiciones de Reset para el contador del programa y el registro CPUSTA.

También la Tabla III.6 muestra las condiciones de inicio de los demás registros.

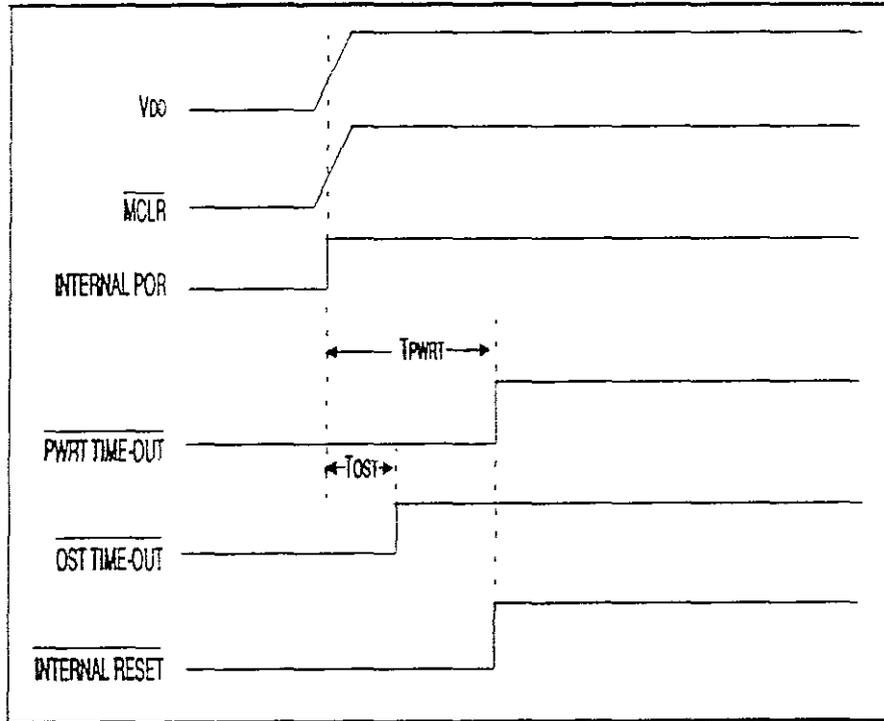


Figura III. 6 Secuencia de tiempo fuera en Power - Up ( $\overline{MCLR}$  Tied To  $V_{DD}$ ).

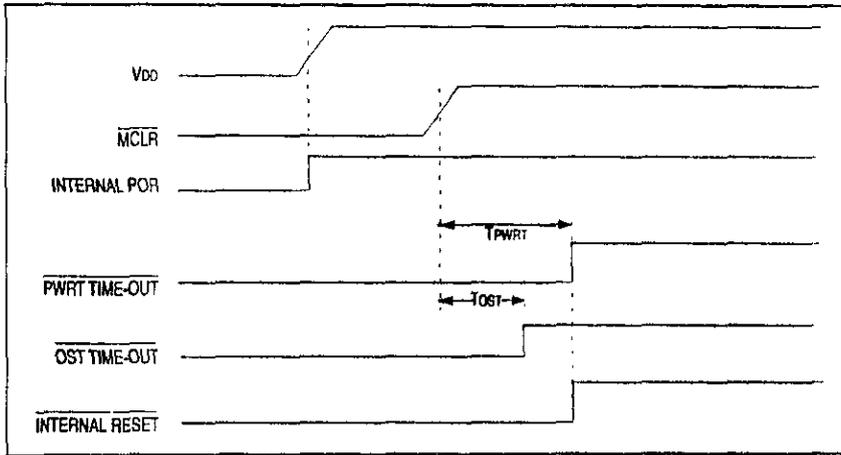


Figura III. 7 Secuencia de tiempo fuera en Power Up (  $\overline{MCLR}$  not tied para  $V_{DD}$ )

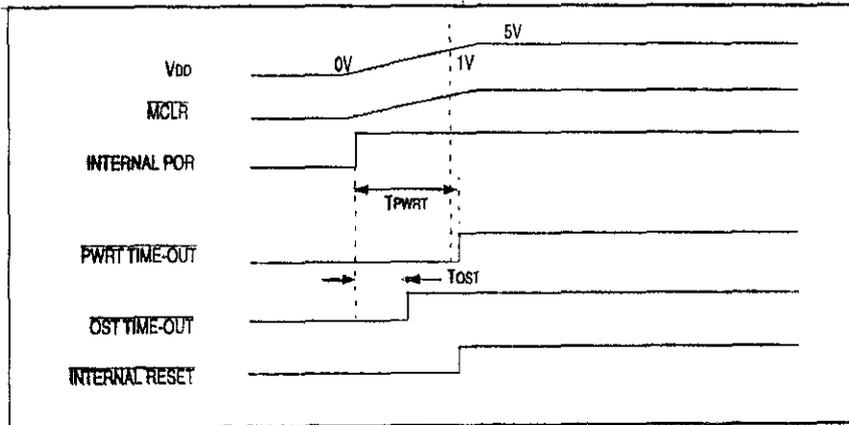


Figura III. 8 Levantamiento lento de tiempo (  $\overline{MCLR}$  tied para  $V_{DD}$ )

Register	Address	Power-on Reset	MCLR Reset WDT Reset	Wake-up from SLEEP through Interrupt
<b>Unbanked</b>				
INDFO	00h	0000 0000	0000 0000	0000 0000
FSR0	01h	xxxx xxxx	xxxx xxxx	xxxx xxxx
PCL	02h	0000h	0000h	PC + 1 <sup>(2)</sup>
PCLATH	03h	0000 0000	0000 0000	xxxx xxxx
ALUSTA	04h	1111 xxxxx	1111 xxxxx	1111 xxxxx
Y0STA	05h	0000 000	0000 000	0000 000
CPUSTA <sup>(3)</sup>	06h	--11 11--	--11 11--	--11 11--
INTSTA	07h	0000 0000	0000 0000	xxxx xxxx <sup>(1)</sup>
INDF1	08h	0000 0000	0000 0000	xxxx xxxx
FSR1	09h	xxxx xxxx	xxxx xxxx	xxxx xxxx
WREG	0Ah	xxxx xxxx	xxxx xxxx	xxxx xxxx
TMR0L	0Bh	xxxx xxxx	xxxx xxxx	xxxx xxxx
TMR0H	0Ch	xxxx xxxx	xxxx xxxx	xxxx xxxx
TBLPTRL <sup>(4)</sup>	0Dh	xxxx xxxx	xxxx xxxx	xxxx xxxx
TBLPTRH <sup>(4)</sup>	0Eh	xxxx xxxx	xxxx xxxx	xxxx xxxx
TBLPTRL <sup>(4)</sup>	0Dh	0000 0000	0000 0000	xxxx xxxx
TBLPTRH <sup>(4)</sup>	0Eh	0000 0000	0000 0000	xxxx xxxx
BSF	0Fh	0000 0000	0000 0000	xxxx xxxx
<b>Bank 0</b>				
PORTA	10h	0-xx xxxx	0-xx xxxx	xxxx xxxx
DORB	11h	1111 111	1111 111	xxxx xxxx
PORTB	12h	xxxx xxxxx	xxxx xxxx	xxxx xxxx
RCSTA	13h	0000 -00x	0000 -00x	xxxx -111
RCREG	14h	xxxx xxxxx	xxxx xxxx	xxxx xxxx
TXSTA	15h	0000 --1x	0000 --1x	xxxx --11
TXREG	16h	xxxx xxxxx	xxxx xxxx	xxxx xxxx
SPBRG	17h	xxxx xxxxx	xxxx xxxx	xxxx xxxx
<b>Bank 1</b>				
DORC	10h	1111 111	1111 111	xxxx xxxx
PORTC	11h	xxxx xxxxx	xxxx xxxx	xxxx xxxx
DORD	12h	1111 111	1111 111	xxxx xxxx
PORTD	13h	xxxx xxxxx	xxxx xxxx	xxxx xxxx
DORE	14h	---- -111	---- -111	---- -111
PORTE	15h	---- -xxx	---- -xxx	---- -111
PIR	16h	0000 0010	0000 0010	xxxx xxxx <sup>(1)</sup>
PIE	17h	0000 0000	0000 0000	xxxx xxxx

Tabla III. 6 Condiciones de inicio para registros de funciones especiales.

Register	Address	Power-on Reset	MCLR Reset WDT Reset	Wake-up from SLEEP through interrupt
<b>Bank 2</b>				
TMR1	10h	XXXX XXXX	UUUU UUUU	UUUU UUUU
TMR2	11h	XXXX XXXX	UUUU UUUU	UUUU UUUU
TMR3L	12h	XXXX XXXX	UUUU UUUU	UUUU UUUU
TMR3H	13h	XXXX XXXX	UUUU UUUU	UUUU UUUU
PR1	14h	XXXX XXXX	UUUU UUUU	UUUU UUUU
PR2	15h	XXXX XXXX	UUUU UUUU	UUUU UUUU
PR3/CA1L	16h	XXXX XXXX	UUUU UUUU	UUUU UUUU
PR3/CA1H	17h	XXXX XXXX	UUUU UUUU	UUUU UUUU
<b>Bank 3</b>				
PW1DCL	10h	XX-- ----	UU-- ----	UU-- ----
PW2DCL	11h	XX-- ----	UU-- ----	UU-- ----
PW1DCH	12h	XXXX XXXX	UUUU UUUU	UUUU UUUU
PW2DCH	13h	XXXX XXXX	UUUU UUUU	UUUU UUUU
CA2L	14h	XXXX XXXX	UUUU UUUU	UUUU UUUU
CA2H	15h	XXXX XXXX	UUUU UUUU	UUUU UUUU
TCON1	16h	0000 0000	0000 0000	UUUU UUUU
TCON2	17h	0000 0000	0000 0000	UUUU UUUU
<b>Unbanked</b>				
PRODL <sup>(B)</sup>	18h	XXXX XXXX	UUUU UUUU	UUUU UUUU
PRODH <sup>(B)</sup>	19h	XXXX XXXX	UUUU UUUU	UUUU UUUU

Tabla III.6 Condiciones de inicio para registros de funciones especiales.

### III.1.F. INTERRUPCIONES

El dispositivo PIC tiene once fuentes de interrupciones.

- Interrupción externa por el pin RA0/INT.
- Sobreflujo del TMR0.
- Sobreflujo del TMR1.
- Sobreflujo del TMR2.
- Sobreflujo del TMR3.
- Transmisión vacía del buffer USART.
- Recepción completa del buffer USART.

- Captura 1.
- Captura 2.
- Borde de ocurrencia TOCK1.

Existen cuatro registros utilizados en el control y estado de las interrupciones, estos son:

- CPUTA
- INTSTA
- PIE
- PIR
- 

El registro CPUTA contiene el bit de GLINTD. Este es el bit de inhibición de interrupción global. Cuando este bit es puesto, todas las interrupciones son inhabilitadas. Este bit es la parte importante de la funcionalidad del controlador. Éste se describe en la sección de la organización de la memoria. Cuando una interrupción está respondiendo, el bit GLINTD es automáticamente puesto para inhibir cualquier otra interrupción, la dirección de retorno es metida en el stack y el PC es cargado con la dirección del vector de interrupción. Cada vector direcciona para una fuente de interrupción específica (excepto para interrupciones periféricas que tienen el mismo vector de dirección). Estas fuentes son:

- Interrupción externa del pin RA0/INT.
- Sobreflujo TMRO.
- Detección de borde TOCK1.
- Cualquier interrupción periférica.

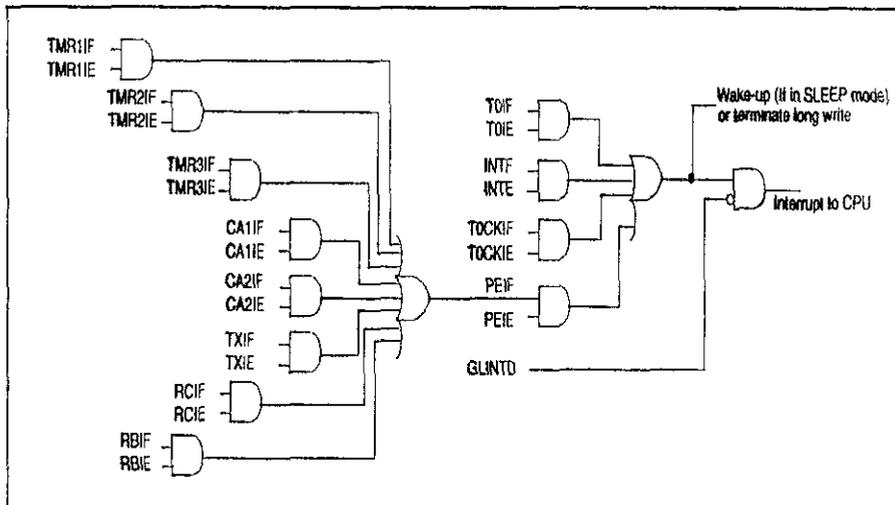


Figura III. 9 Interrupciones Lógicas

Cuando el programa ejecuta una de estas interrupciones en el registro de interrupciones (excepto para las direcciones de interrupciones periféricas), el bit de la bandera de interrupción es automáticamente limpiada. Direccionando la interrupción periférica del vector de dirección, no se limpia automáticamente la fuente de interrupción. En el servicio de interrupción periférica, la(s) fuente(s) de interrupción puede ser determinada para valorar los bits de la bandera de interrupción. Los bits de la bandera de interrupción pueden ser limpiados por el software antes de rehabilitar las interrupciones, evitando llamados de interrupciones infinitas.

Todos los bits de las banderas de interrupciones individuales podrán ser puestas en el bit de GLINTD a pesar del estado de su correspondiente bit de máscara.

Para eventos de interrupción externa, existirá una interrupción latente. Para dos ciclos de instrucciones, deberá ser una instrucción de un ciclo largo para estar latente. La instrucción de "retorno de interrupción", RETFIE, puede ser utilizada para marcar el final de la rutina de servicio de la interrupción. Cuando esta instrucción es ejecutada, el STACK está "esperando", y el bit GLINTD es limpiado (para rehabilitar la interrupción).

### **III.1.F.1. REGISTRO DE ESTADO DE INTERRUPCIÓN (INTERRUPT STATUS REGISTER INTSTA)**

El registro de interrupción de STATUS/CONTROL (INTSTA) graba las llamadas de las interrupciones individualmente en los bits de bandera, y contiene la habilitación de los bits de interrupción individual (no para los periféricos).

El bit PEIF es de lectura únicamente. Todos los bits de banderas periféricas están en el registro PIR. Se deberá tener cuidado al limpiar los bits de habilitación de cualquier registro cuando las interrupciones están activadas (GLINTD esta limpio). Si cualquiera de los bits de las banderas (TOIF, INTF, TOCKIF, o PEIF, es colocado en el mismo ciclo de instrucción, el correspondiente bit que habilita la interrupción es quitado y el dispositivo lo llevará a la dirección de reset (0x00).

Cuando se quita cualquier bit de habilitación del INTSTA, se debe poner el GLINTD (inhibir).

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PEIF	TOCKIF	TOIF	INTF	PEIE	TOCKIE	TOIE	INTE
bit7							bit0
R = Readable bit W = Writable bit - n = Value at POR reset							
bit 7:	<b>PEIF: Peripheral Interrupt Flag bit</b> This bit is the OR of all peripheral interrupt flag bits AND'ed with their corresponding enable bits 1 = A peripheral interrupt is pending 0 = No peripheral interrupt is pending						
bit 6:	<b>TOCKIF: External interrupt on TOCKI Pin Flag bit</b> This bit is cleared by hardware, when the interrupt logic forces program execution to vector (18h) 1 = The software specified edge occurred on the RA1/TOCKI pin 0 = The software specified edge did not occur on the RA1/TOCKI pin						
bit 5:	<b>TOIF: TMRO Overflow Interrupt Flag bit</b> This bit is cleared by hardware, when the interrupt logic forces program execution to vector (10h) 1 = TMRO overflowed 0 = TMRO did not overflow						
bit 4:	<b>INTF: External interrupt on INT Pin Flag bit</b> This bit is cleared by hardware, when the interrupt logic forces program execution to vector (08h). 1 = The software specified edge occurred on the RA0/INT pin 0 = The software specified edge did not occur on the RA0/INT pin						
bit 3:	<b>PEIE: Peripheral Interrupt Enable bit</b> This bit enables all peripheral interrupts that have their corresponding enable bits set. 1 = Enable peripheral interrupts 0 = Disable peripheral interrupts						
bit 2:	<b>TOCKIE: External interrupt on TOCKI Pin Enable bit</b> 1 = Enable software specified edge interrupt on the RA1/TOCKI pin 0 = Disable interrupt on the RA1/TOCKI pin						
bit 1:	<b>TOIE: TMRO Overflow Interrupt Enable bit</b> 1 = Enable TMRO overflow interrupt 0 = Disable TMRO overflow interrupt						
bit 0:	<b>INTE: External interrupt on RA0/INT Pin Enable bit</b> 1 = Enable software specified edge interrupt on the RA0/INT pin 0 = Disable software specified edge interrupt on the RA0/INT pin						

Figura III. 10 Registro INSTA (Dirección: 07h, no está en los bancos).

**III.1.F.2. INTERRUPCIONES PERIFÉRICAS,  
HABILITACIÓN DEL REGISTRO (PERIPHERAL  
INTERRUPT ENABLE REGISTER (PIE)).**

Este registro contiene los bits de las banderas individuales para las interrupciones periféricas.

	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE
bit 7								bit 0
bit 7	<b>RBIE: PORTB Interrupt on Change Enable bit</b> 1 = Enable PORTB interrupt on change 0 = Disable PORTB interrupt on change							
bit 6	<b>TMR3IE: Timer3 Interrupt Enable bit</b> 1 = Enable Timer3 interrupt 0 = Disable Timer3 interrupt							
bit 5	<b>TMR2IE: Timer2 Interrupt Enable bit</b> 1 = Enable Timer2 interrupt 0 = Disable Timer2 interrupt							
bit 4	<b>TMR1IE: Timer1 Interrupt Enable bit</b> 1 = Enable Timer1 interrupt 0 = Disable Timer1 interrupt							
bit 3	<b>CA2IE: Capture2 Interrupt Enable bit</b> 1 = Enable Capture interrupt on RB1/CAP2 pin 0 = Disable Capture interrupt on RB1/CAP2 pin							
bit 2	<b>CA1IE: Capture1 Interrupt Enable bit</b> 1 = Enable Capture interrupt on RB2/CAP1 pin 0 = Disable Capture interrupt on RB2/CAP1 pin							
bit 1	<b>TXIE: USART Transmit Interrupt Enable bit</b> 1 = Enable Transmit buffer empty interrupt 0 = Disable Transmit buffer empty interrupt							
bit 0	<b>RCIE: USART Receive Interrupt Enable bit</b> 1 = Enable Receive buffer full interrupt 0 = Disable Receive buffer full interrupt							

R = Readable bit  
 W = Writable bit  
 -n = Value at POR reset

Figura III. 11 Registro PIE (Dirección: 17h, banco 1).

### III.1.F.3. PERIPHERAL INTERRUPT REQUEST REGISTER

(PIR).

Este registro contiene los bits de las banderas individuales para las interrupciones periféricas.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R-0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF
bit7						bit0	
<p><b>bit 7. RBIF: PORTB Interrupt on Change Flag bit</b>            1 = One of the PORTB inputs changed (Software must end the mismatch condition)            0 = None of the PORTB inputs have changed</p> <p><b>bit 6. TMR3IF: Timer3 Interrupt Flag bit</b>            If Capture1 is enabled (CA1/PR3 = 1)            1 = Timer3 overflowed            0 = Timer3 did not overflow            If Capture1 is disabled (CA1/PR3 = 0)            1 = Timer3 value has rolled over to 0000h from equalling the period register (PR3H:PR3L) value            0 = Timer3 value has not rolled over to 0000h from equalling the period register (PR3H:PR3L) value</p> <p><b>bit 5. TMR2IF: Timer2 Interrupt Flag bit</b>            1 = Timer2 value has rolled over to 0000h from equalling the period register (PR2) value            0 = Timer2 value has not rolled over to 0000h from equalling the period register (PR2) value</p> <p><b>bit 4. TMR1IF: Timer1 Interrupt Flag bit</b>            If Timer1 is in 8-bit mode (T16 = 0)            1 = Timer1 value has rolled over to 0000h from equalling the period register (PR1) value            0 = Timer1 value has not rolled over to 0000h from equalling the period register (PR2) value            If Timer1 is in 16-bit mode (T16 = 1)            1 = TMR1:TMR2 value has rolled over to 0000h from equalling the period register (PR1:PR2) value            0 = TMR1:TMR2 value has not rolled over to 0000h from equalling the period register (PR1:PR2) value</p> <p><b>bit 3. CA2IF: Capture2 Interrupt Flag bit</b>            1 = Capture event occurred on RB1/CAP2 pin            0 = Capture event did not occur on RB1/CAP2 pin</p> <p><b>bit 2. CA1IF: Capture1 Interrupt Flag bit</b>            1 = Capture event occurred on RB0/CAP1 pin            0 = Capture event did not occur on RB0/CAP1 pin</p> <p><b>bit 1. TXIF: USART Transmit Interrupt Flag bit</b>            1 = Transmit buffer is empty            0 = Transmit buffer is full</p> <p><b>bit 0. RCIF: USART Receive Interrupt Flag bit</b>            1 = Receive buffer is full            0 = Receive buffer is empty</p>							

Figura III. 12 Registro PIR (DIRECCIÓN: 16h, BANCO 1)

Nota: Estos bits serán puestos por una condición específica, aunque el bit de la interrupción correspondiente sea limpiado (inhibe la interrupción), o el bit GLINTD sea puesto (todas las interrupciones inhabilitadas). Antes de habilitar una interrupción, el usuario tendrá que limpiar la bandera de interrupción, para asegurar que el programa no se bifurque por el servicio de interrupción periférica.

#### III.1.F.4. OPERACIÓN DE INTERRUPCIÓN.

El bit de inhabilitación de interrupción global, GLINTD (CPUSTA <4 >, habilita todas las interrupciones no enmascarables (son limpiadas, o todas las interrupciones son inhabilitadas (son puestas). Las interrupciones pueden ser inhabilitadas a través de su correspondiente bit de habilitación en el registro INTSTA. Las interrupciones periféricas necesitan cualquier habilitación periférica global, por ejemplo; inhabilitando el bit del PEIE, o habilitando el periférico específico del bit inhabilitado.

Las instrucciones del RETFIE permiten el retorno de las interrupciones y rehabilitan las interrupciones al mismo tiempo. Cuando una interrupción está respondiendo, el bit del GLINTD es puesto automáticamente para inhabilitar cualquier otra interrupción, el retorno de la dirección es metido dentro del "stack" y el contador del programa (PC) es cargado con el vector de interrupciones. Existen cuatro vectores de interrupciones para reducir la interrupción latente.

El vector de interrupción periférica tiene fuentes de interrupción múltiple. Una vez que la rutina de servicio de interrupciones periféricas, las fuentes de la interrupción pueden

ser determinadas por la obtención de los bits de las banderas de interrupción. Los bits de las banderas de interrupción deben ser limpiadas en el software antes de rehabilitar las interrupciones para evitar interrupciones continuas.

Los dispositivos PIC17CX tienen cuatro vectores de interrupción. Estos vectores y su prioridad de interrupción por hardware se muestran en la tabla III.7 Si ocurre la habilitación de dos interrupciones "al mismo tiempo", la interrupción de más alta prioridad será la del primer servicio. Esto quiere decir que la dirección del vector de la interrupción será cargada en el contador del programa (PC).

Tabla III.7 Vectores de Interrupción /prioridades.

DIRECCIÓN	VECTOR	PRIORIDAD
0008h	Interrupción externa en RAO/ INT pin (INTF)	1 (alta)
0010h	Interrupción de sobreflujo (TOIF)	2
0018h	Interrupción externa en TOCKI (TOCKIF)	3
0020h	Periféricos	4 (baja)

**NOTA 1:** Los bits de las banderas de interrupción individual son puestos a pesar del estado de su correspondiente bit mascarable o del bit de GLINDT.

**NOTA 2:** Cuando se inhabilita cualquiera de los bits puestos del INTSTA, el bit del GLINTD deberá ser activado.

### III.1.F.5. INTERRUPCIÓN RAO/INT.

La interrupción externa en el pin **RAO/INT** dispara al borde. Cualquier cambio de borde puede producir un estado diferente. Si el borde está en alto el bit es puesto **INTE** (**TOSTA <7>**), o si el borde cae, entonces el bit del **INTE** es quitado. Cuando un borde válido aparece en el pin de **RAO/INT**, el bit es puesto **INTF** (**INTSTA < 4 >**). Esta interrupción puede ser inhabilitada al limpiarse el bit de control del **INTE** (**INTSTA < 0 >**). La interrupción **INT** puede despertar al procesador del estado de "sueño".

### III.1.F.6. INTERRUPCIÓN TMRO.

Un sobreflujo (**FFFFh → 0000h**) en **TMRO** pondrá al bit **TOIF** (**INTSTA < 5 >**). La interrupción podrá ser habilitada /inhabilitada por el bit de control seleccionar/quitar del **INTE** (**INTSTA < 1 >**).

### III.1.F.7. INTERRUPCIÓN TOCKI

La interrupción externa en el pin **RA1/TOCKI** es disparada por el borde. Si el borde es alto se activa el **TOSE** (**TOSTA < 6 >**), si el nivel es bajo entonces el **TOSE** se limpia. Cuando un borde válido aparece en pin del **RA1/TOCKI**, el bit del **TOCKIF** es puesto (**INTSTA <2>**). La interrupción **TOCKI** puede despertar de su estado de "sueño" al procesador.

### **III.1.F.8. INTERRUPCIONES PERIFÉRICAS**

Las banderas de interrupciones periféricas indican que al menos una de las interrupciones periféricas ocurren (PEIF es puesta). El bit del PEIF es únicamente de lectura: es un bit de estado lógico OR de los demás bits de banderas en el registro PIR, y es de AND con el correspondiente bit habilitado en el registro PIE. Algunas de las interrupciones periféricas pueden despertar al procesador del estado de "sueño".

### **III.1.F.9. CONTEXTOS QUE SE GUARDAN DURANTE LAS INTERRUPCIONES (CONTEXT SAVING DURING INTERRUPTS).**

Durante una interrupción, únicamente el valor que retorna al pc es el que está salvado en el stack. es común que los usuarios deseen salvar la posición del registro durante la interrupción, por ejemplo los registros wreg, alusta y el bsr. esto requiere su desarrollo e indicación en el software.

El ejemplo III.3 muestra la manera de realizar el respaldo y la forma de almacenar la información para un servicio de interrupción de una rutina. Tanto la rutina del PUSH (introducir) como la del POP (extraer) deberán ser, en cada servicio, una interrupción de rutina o subrutina que será llamada. Dependiendo de la aplicación, otros registros pueden también necesitarse para salvarlos, así como el PCLATH.

```

; The addresses that are used to store the CPOSTA and WREG values
; must be in the data memory address range of 18n - 1Fh. Up to
; 8 locations can be saved and restored using
; the MOVFP instruction. This instruction neither affects the status
; bits, nor corrupts the WREG register.
;
;
PUSH  MOVFP  WREG, TEMP_W      ; Save WREG
      MOVFP  ALUSTA, TEMP_STATUS ; Save ALUSTA
      MOVFP  BSR, TEMP_BSR     ; Save BSR

ISR   :                               ; This is the interrupt service routine
      :
POP   MOVFP  TEMP_W, WREG      ; Restore WREG
      MOVFP  TEMP_STATUS, ALUSTA ; Restore ALUSTA
      MOVFP  TEMP_BSR, BSR     ; Restore BSR
      RETFIE                   ; Return from Interrupts enabled

```

Ejemplo III. 3

### III.1.G. ORGANIZACIÓN DE LA MEMORIA.

Existen dos bloques de memoria en el PIC17C4X, el programa de memoria y los datos de la memoria. Cada bloque tiene su propio bus y el acceso a ellos puede ocurrir durante el mismo ciclo del oscilador.

La memoria de los datos puede, además, ser modificada en la RAM de propósito general y en los registros de funciones especiales (SFRs). Las operaciones de los SFRs son las que controlan el "núcleo" que será descrito aquí. La utilización de los SFRs para controlar los módulos periféricos se describirá individualmente.

### **III.1.G.1. ORGANIZACIÓN DE LA MEMORIA DEL PROGRAMA.**

El PIC17CX tiene un contador de programa de 16 bits capaz de direccionar espacios de memoria de programas de 64K x 16. El vector de reset esta en 0000h y los vectores de interrupcion están en la 0008h, 0010h, 0018h y 0020h (Fig. III.1.A.7.1).

### **III.1.G.2. OPERACIÓN DE LA MEMORIA DEL PROGRAMA.**

El PIC17C4X puede operar en una de las cuatro posibles configuraciones de programa. Las configuraciones son seleccionadas por dos configuraciones de bits y los modos posibles son (Figura III.13):

- Microprocesador.
- Microcontrolador.
- Microcontrolador extendido.
- Microcontrolador protegido.

Los modos de microcontrolador y microcontrolador protegido únicamente permiten ejecuciones internas. Cualquier acceso va más allá de la lectura de la memoria del programa, desconociendo los datos.

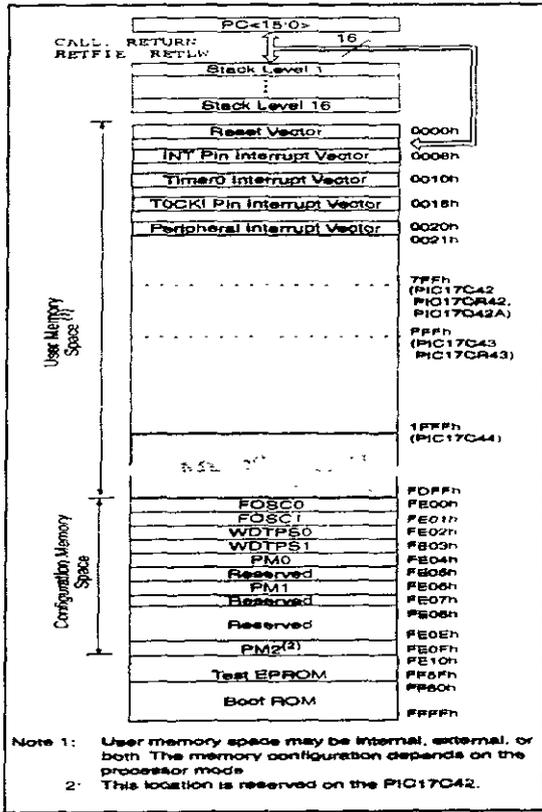


Figura III. 13 Mapa de memoria y del stack.

El modo de microcontrolador protegido también habilita las características de protección de código.

En el modo de microcontrolador extendido se introduce tanto la memoria interna como la externa del programa. La ejecución de conmutación es automática entre la memoria externa y la interna. Los 16 bits de dirección permiten una memoria de programa de 64K de palabras.

En el modo de microprocesador únicamente se introduce la memoria externa del programa. El programa de memoria del chip es ignorado. Los 16 bits de dirección permiten un rango de memoria de 64 palabras. El modo de microprocesador es el modo común de un dispositivo no programado.

Los diferentes modos permiten distintos accesos a la ROM por la configuración de los bits. La tabla III.8 muestra qué modos pueden introducirse con qué áreas de memoria. La memoria de prueba y el acceso de memoria (Boot Memory) no se necesitan en la operación normal del dispositivo. Deberá de tenerse cuidado para asegurar que no se intenten bifurcaciones en estas áreas.

Modo de operación	Memoria interna del programa	Configuración de Bits, memoria de prueba, acceso a la ROM.
Microprocesador	No tiene acceso	No tiene acceso
Microcontrolador	Acceso	Acceso
Microcontrolador Extendido	Acceso	No tiene acceso
Microcontrolador Protegido	Acceso	Acceso

Tabla III. 8 Accesos al modo de memoria.

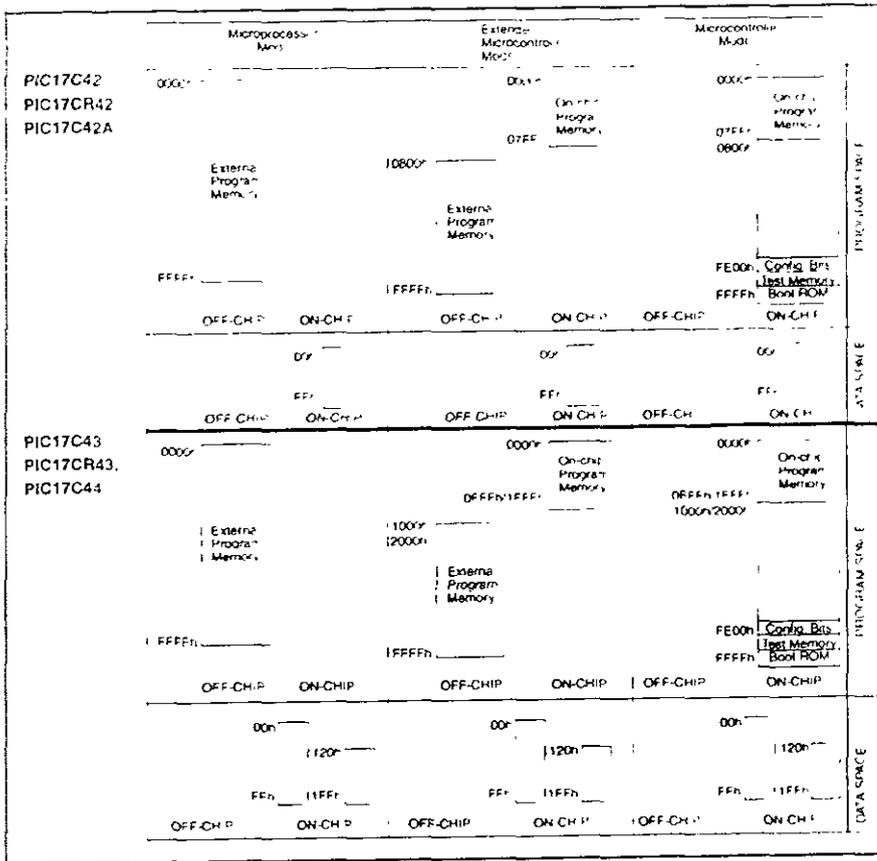


Tabla III. 9 Mapa de memoria en los diferentes modos.

Cuando cualquiera de los dos modos (microprocesador o microcontrolador extendido) es seleccionado, el PORTC, PORTD, PORTE son configurados como el bus del sistema. El PORTC y PORTD son los que multiplexan al bus como direccion/datos, y el PORTE se utiliza para señales de control. Los componentes externos son necesarios para demultiplexar las direcciones y los datos. Esto se puede hacer como se muestra en la Fig.III.14.

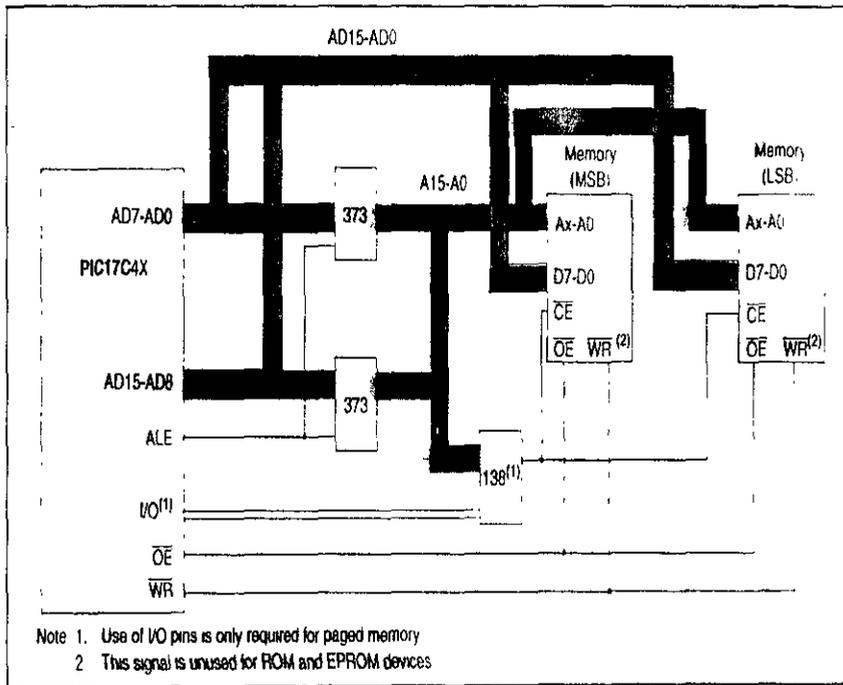


Figura III. 14 Diagrama de conexiones para memoria de programa externa.

Las formas de onda de las direcciones y de los datos son como se muestran en la Fig. III.15

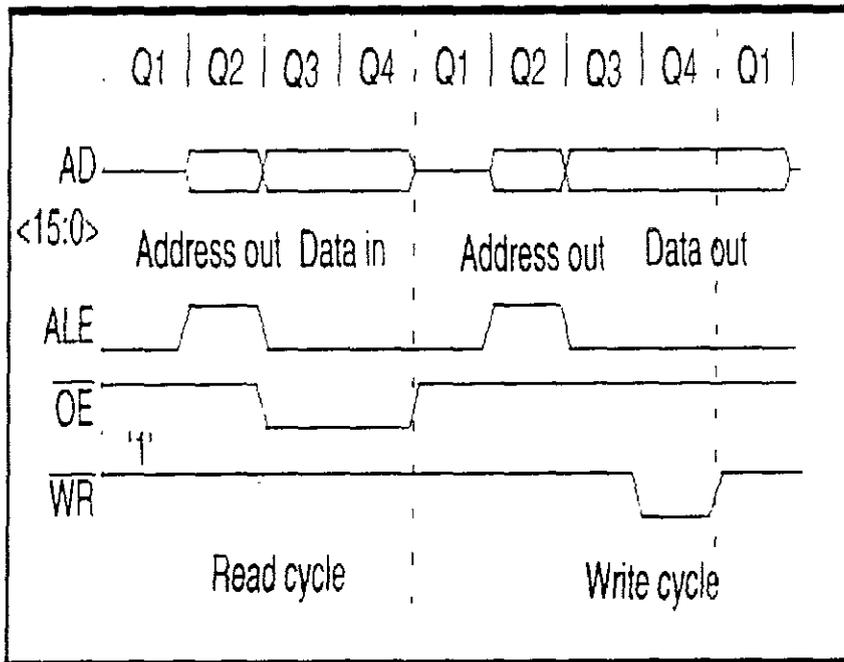


Figura III. 15 Programa externo. Acceso de memoria. Formas de onda

Para que el bus del sistema sea capaz de retener el valor deseado. Se debe tener cuidado de que no existan conflictos en el bus (perdidas mínimas) y que el valor de salida (dirección) sea el correcto.

Cuando se incrementa la velocidad del procesador, se pueden utilizar memorias externas de rapido acceso en tiempo. En la tabla III.10 se encuentran las velocidades de memorias externas requeridas para una frecuencia dada por el dispositivo PIC17CX.

Tabla III. 10 Acceso de la memoria EPROM de alta prioridad.

PIC17C4X FRECUENCIA DE OSCILACIÓN	CICLO DE INSTRUCCIONES TIEMPO ( $T_{CIC}$ )	PRIORIDAD DE LA EPROM
8 MHz	500 ns	-25
16 MHz	250 ns	-15
20 MHz	200 ns	-10
25 MHz	160 ns	-70
33 MHz	121 ns	(1)

Nota: El acceso a estos tiempos requieren el uso de memorias FAST SRAMS.

NOTA : La interface de memoria externa no está soportada por dispositivos LC.

### III.1.G.3. ORGANIZACIÓN DE LOS DATOS DE MEMORIA.

Los datos de memoria están seccionados en dos áreas. El primero es el Registro de Propósito General (GPR), el segundo es el registro de funciones especiales (Special Function Register SFR). El SFR controla las operaciones de los dispositivos.

Las proporciones de los datos de memoria están en bloques ordenados, esto es para ambas áreas. El área del GPR está en un bloque ordenado para permitir tener una RAM de propósito general mayor que 232 bytes. El SFR es para los registros que controlan las funciones periféricas. Los bloques ordenados requieren el uso de bits de control para seleccionar al banco. Estos bits de

control son localizados en el Registro de Seleccior de Banco Bank Select Register BSR. Si un acceso está diseñado para una localización fuera de su región de banco, el bit de BSR es ignorado. La Fig.III.16 muestra la organización de los datos de memoria.

Addr	Unbanked			
00h	INDFO			
01h	FSR0			
02h	PCL			
03h	PCLATH			
04h	ALUSTA			
05h	TOSTA			
06h	CPUSTA			
07h	INTSTA			
08h	INDF1			
09h	FSR1			
0Ah	WREG			
0Bh	TMR0L			
0Ch	TMR0H			
0Dh	TBLPTRL			
0Eh	TBLPTRH			
0Fh	BSR			
	Bank 0	Bank 1 (1)	Bank 2 (1)	Bank 3 (1)
10h	PORTA	DDRC	TMR1	PW1DCL
11h	DDRB	PORTC	TMR2	PW2DCL
12h	PORTB	DDRD	TMR3L	PW1DCH
13h	RCSTA	PORTD	TMR3H	PW2DCH
14h	RCREG	DDRE	PR1	CA2L
15h	TXSTA	PORTE	PR2	CA2H
16h	TXREG	PIR	PR3L/CA1L	TCON1
17h	SPBRG	PIE	PR3H/CA1H	TCON2
18h	PRODL			
19h	PRODH			
1Ah				
1Fh				
20h	General Purpose RAM (2)	General Purpose RAM (2)		
FFh				

Figura III.16 Mapa del Archivo de Registro.

#### **III.1.G.4. REGISTRO DE PROPOSITO GENERAL (GPR) .**

Todos los dispositivos tienen algunas porciones en el area del GPR. La longitud del GPR es de 8 bits. Cuando el area del GPR es mayor que 232, se ordenara en conjuntos para permitir el acceso al espacio de la memoria adicional.

Unicamente los dispositivos PIC17C43 y PIC17C44 tienen conjuntos ordenados de memoria en el área del GPR. Para facilitar la conmutacion entre estos bancos, la instrucción MOVLR bank ha sido adicionada al set de instrucciones. El GPR se inicia con un POWER ON RESET y no es cambiado para todos los demas resets.

#### **III.1.G.5. REGISTROS DE FUNCIONES ESPECIALES (SFR) .**

Los SFR son utilizados por el CPU y funciones periféricas para el control de operaciones del dispositivo. Estos registros estan en una RAM estática.

El SFR puede ser clasificado en dos sets. Esto esta asociado con la parte importante y fundamental de la funcion y esta relacionado con las funciones periféricas. Los registros perifericos se encuentran en la porción del banco de la memoria, permitiendo al núcleo del registro estar en la región que no es la del banco. La instrucción MOVLB bank facilita la conmutacion entre los bancos periféricos.

### III.1.G.6. REGISTROS DE ESTADOS ALU.

El registro ALU contiene el estado de los bits de la Unidad Lógica Aritmética y el modo de los bits de control para el direccionamiento indirecto de los registros.

Así como los demás registros, el registro ALUSTA puede ser destinado para cualquier instrucción. Si el registro ALUSTA es destinado por una instrucción que afecta a los bits de Z, DC o C, entonces él escribe para estos tres bits su inhabilitación.

Estos bits son puestos o quitados de acuerdo al dispositivo lógico. Por lo tanto, el resultado de una instrucción con el registro ALUSTA como un destino puede ser diferente al deseado.

Por ejemplo, CLRF ALUSTA limpiará la parte más significativa de los cuatro bits y pondrá al bit de Z. Esto deja al registro del ALUSTA como 0000uluu (donde u = no hay cambio).

Únicamente las instrucciones BCF, BSF, SWAPF y MOVWF se utilizan para alterar a los registros del ALUSTA, porque estas instrucciones no hacen efecto en cualquier bit de estado.

R/W - 1	R/W - 1	R/W - 1	R/W - 1	R/W - x	R/W - x	R/W - x	R/W - x
FS3	FS2	FS1	FS0	OV	Z	DC	C
bit7				bit0			

R = Readable bit  
W = Writable bit  
-n = Value at POR reset  
(x = unknown)

bit 7-6: **FS3:FS2:** FSR1 Mode Select bits  
00 = Post auto-decrement FSR1 value  
01 = Post auto-increment FSR1 value  
1x = FSR1 value does not change

bit 5-4: **FS1:FS0:** FSR0 Mode Select bits  
00 = Post auto-decrement FSR0 value  
01 = Post auto-increment FSR0 value  
1x = FSR0 value does not change

bit 3: **OV:** Overflow bit  
This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude, which causes the sign bit (bit7) to change state.  
1 = Overflow occurred for signed arithmetic, (in this arithmetic operation)  
0 = No overflow occurred

bit 2: **Z:** Zero bit  
1 = The result of an arithmetic or logic operation is zero  
0 = The results of an arithmetic or logic operation is not zero

bit 1: **DC:** Digit carry/borrow bit  
For ADDWF and ADDLW instructions  
1 = A carry-out from the 4th low order bit of the result occurred  
0 = No carry-out from the 4th low order bit of the result  
Note: For borrow the polarity is reversed.

bit 0: **C:** carry/borrow bit  
For ADDWF and ADDLW instructions.  
1 = A carry-out from the most significant bit of the result occurred  
Note that a subtraction is executed by adding the two's complement of the second operand. For rotate (RRCF, RLCF) instructions, this bit is loaded with either the high or low order bit of the source register  
0 = No carry-out from the most significant bit of the result  
Note: For borrow the polarity is reversed

Figura III. 17 Registro ALUSTA.

NOTA 1: Los bits C y DC operan como un  $\overline{\text{borro}}$  fuera del bit de sustracción.

NOTA 2: El bit de sobreflujo se pondrá si el resultado del complemento a 2's excede de +127 o es menor que -128.

La Unidad Logico Aritmetica es capaz de llevar fuera operaciones aritmeticas y logicas en dos operandos o en un solo operando. Todos las instrucciones de los operandos sencillos funcionan tanto en el registro WREG o en el registro de archivos. Para instrucciones de dos operandos, uno de los operandos esta en el registro WREG y el otro en el registro de archivos o es una constante inmediata de 8 - bits.

### III.1.G.7. REGISTRO DE ESTADO DEL CPU (CPUSTA) .

El registro CPUSTA contiene los bits de estado y control del CPU. Estos registros son utilizados para interrupciones globales de habilitacion/innhabilitación. Si sólo se desea una interrupcion especifica para ser habilitada/inhabilitada, se tiene que referir al registro de Interrupcion de Estado (INTSTA) y al registro de habilitacion de interrupcion periferica (PIE). Estos registros también indican si el stack está disponible y contienen a los bits de POWER - DOWN ( $\overline{PD}$ ) y TIME - OUT ( $\overline{TO}$ ). Los bits ( $\overline{PD}$ ), ( $\overline{TO}$ ) y el STKAN no son de escritura. Estos bits son puestos y limpiados de acuerdo al dispositivo lógico. Por lo tanto, el resultado de la instruccion con el registro del CPUSTA como destino puede ser diferente del que se espera.

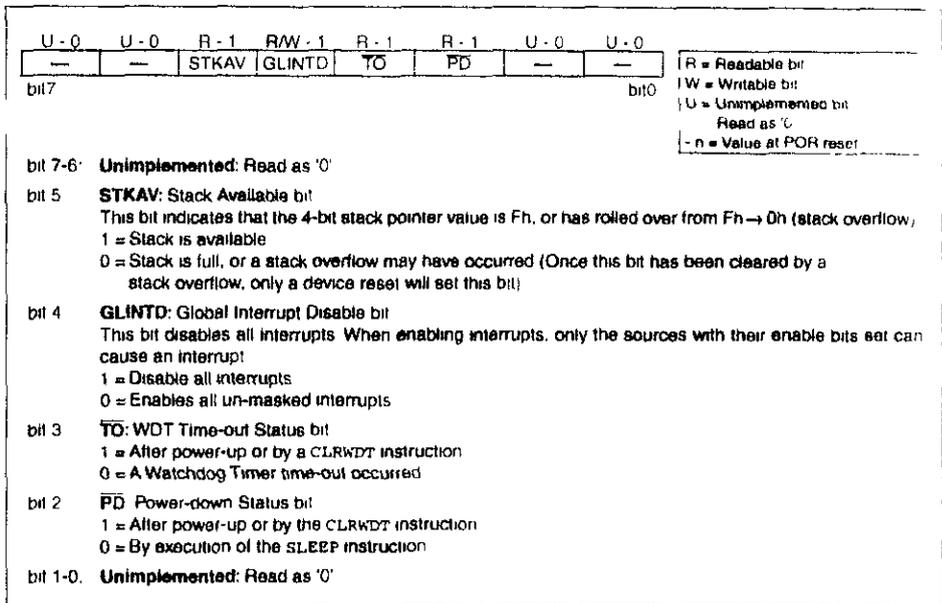


Figura III. 18 Registro del CPUSTA (Dirección: 06h, no está en el banco)

### III.1.G.8. REGISTRO TMRO DE ESTADO Y CONTROL (TOSTA).

Estos registros contienen varios bits de control, el bit 7 (INTEDG) es utilizado para controlar el borde superior como una señal en el pin RA0/INT y pondrá la bandera de interrupción en el RA0/INT. Los otros bits configuran la fuente de reloj y preescalán al TIMER0.

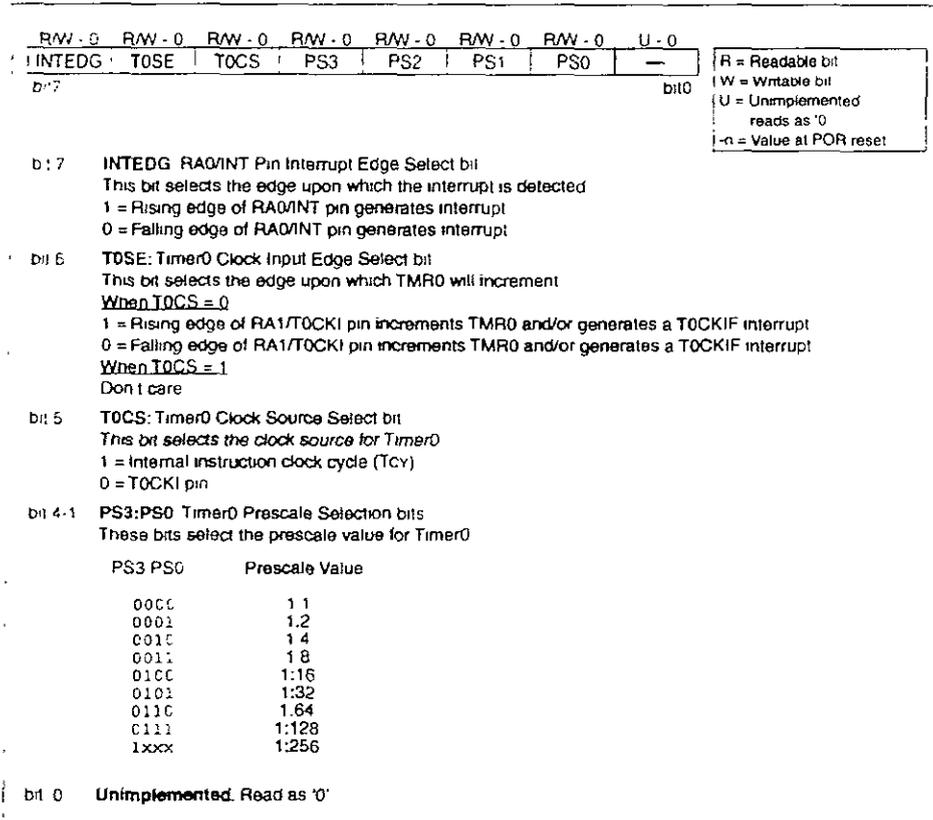


Figura III. 19 Registro TOSTA (DIRECCIÓN: 05h, BANCO).

### III.1.H. OPERACIÓN DEL STACK.

El PIC17C44 tiene un stack de hardware de 16 X 16k. El stack no es parte del programa ni del espacio de los datos de memoria; el stack pointer no es de lectura y tampoco de escritura. El PC (contador del programa) es "introducido" en el stack cuando la

instrucción CALL es ejecutada o una interrupcion es reconocida. El stack está de "salida" en el caso de un RETURN, RETLW o en la ejecución de la instrucción RETFIE; el PCLATH no es afectado por un "introducir" ni por la operacion de "salida".

El stack opera como un buffer circular, el apuntador del stack es inicializado en cero despues de todas las interrupciones. Hay un bit del stack disponible (STKAV) para permitir al software que asegure que el stack no tenga un sobreflujo. El bit STKAV es puesto después de reinicializar al dispositivo, cuando el apuntador del stack es igual a FH, el STKAV es limpiado. Cuando el apuntador del stack pasa de FH a 0H, el bit STAKV permanecerá limpio hasta que el dispositivo sea nuevamente iniciado.

**NOTA 1: No hay un bit de estado para el stack de valor mínimo aceptable. El bit STAKV puede ser utilizado para detectar el valor mínimo aceptable (underflow) que resulta en el apuntador del stack que aparece en el límite del stack.**

**NOTA 2: No hay instrucciones de neomónicos para llamar al PUSH o POP; éstas son acciones que ocurren al ejecutar las instrucciones CALL, RETURN, RETLW y RETFIE, o direccionando al vector de interrupción.**

NOTA 3: Después del reset, si la operacion "POP" ocurre antes de la operación "PUSH", el bit del STAKV será limpiado. Este aparecerá como si el stack estuviera lleno (el underflow ha ocurrido). Si la operación próxima que ha ocurrido es el "PUSH" (antes de otro "POP"), el bit del STAKV se bloqueará en limpio. Únicamente un reset cambiará de estado al bit.

### III.1.1. DIRECCIONAMIENTO INDIRECTO.

El direccionamiento indirecto es un modo de enviar los datos a la memoria, cuando la dirección de los datos de la memoria no está almacenada en la instrucción. Esto es, el registro de lectura o escritura puede ser modificado por el programa. Este puede ser muy útil para datos tabulados en la memoria de datos. La Fig. III.20 realiza la operación de direccionamiento indirecto. Muestra el movimiento del valor a la dirección de la memoria de datos especificado por el valor del registro FSR.

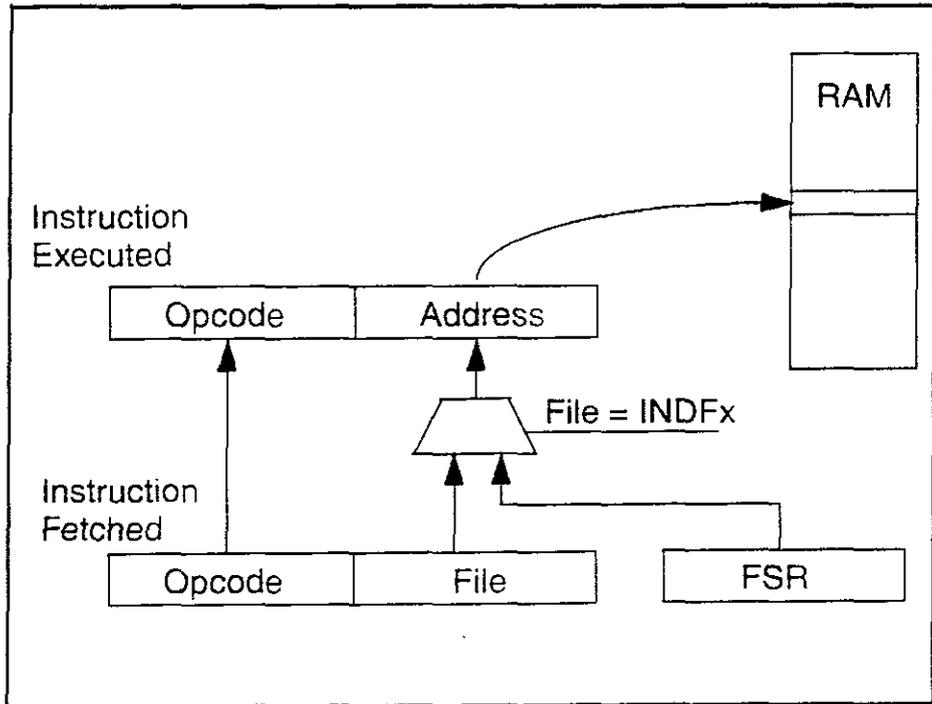


Figura III. 20 Direccionamiento Indirecto.

El ejemplo III.4 muestra el uso del direccionamiento indirecto para limpiar a la RAM en un número mínimo de instrucciones. Un concepto similar puede ser utilizado para mover un determinado número de bits (BLOCK) de datos al registro de transmisión USART (TXREG). Al comenzar el direccionamiento del bloque de datos transmitidos, estos podrían ser fácilmente modificados por el programa.

### III.1.A.7.1. REGISTROS DE DIRECCIONAMIENTO INDIRECTO.

EL PIC17CXX TIENE CUATRO REGISTROS PARA DIRECCIONAMIENTO INDIRECTO:

- INDF0 Y FSR0
- INDF1 Y FSR1

Los registros INDF0 e INDF1 no son físicamente diseñados. Estos registros de direccionamiento indirecto se activan para lectura o escritura, siendo la dirección de los datos el correspondiente valor en el registro FSR. El registro FSR es de 8 bits y permite cualquier direccionamiento en 256 bytes, que es el rango de la dirección de los datos de memoria. Para memoria en bancos, el banco de memoria introducido es especificado por el valor en el FSR.

Si el archivo INDF0 (o INDF1) se lee también por medio del FSR, todos son leídos como cero (se pone el bit 2). Similarmente, si

el INDF0 o INDF1) es escrito indirectamente, la operación equivalente será una instrucción de NOP, y los estados de los bits no son afectados.

### III.1.A.7.2. OPERACIÓN DE DIRECCIONAMIENTO INDIRECTO.

La capacidad del direccionamiento indirecto es superior a los PIC16XX. Existen dos bits de control asociados con cada registro del FSR. Estos dos bits configuran al registro FSR para:

- Auto decremento del valor (dirección) en el FSR después de un acceso indirecto.
- Auto incremento del valor (dirección) en el FSR después de un acceso indirecto.
- No hay cambio de valores (dirección) en el FSR después de un acceso indirecto.

Estos bits de control son localizados en el registro ALUSTA. El registro FSR1 es controlado por los bits FS3: FS2 y FSR0 es controlado por los bits FS1:FS0.

Cuando se utilizan las características de auto decremento o auto incremento, el efecto en el FSR no es reflejado en el registro ALUSTA. Por ejemplo, si el direccionamiento indirecto ocasiona que el FSR sea igual a cero, el bit Z no se pondra.

Si el registro FSR contiene un valor de 0h, una lectura indirecta leerá 0h (Z es puesto); cuando es una escritura indirecta sera equivalente a un NOP (el estado del bit no es afectado..

El direccionamiento indirecto permite un ciclo sencillo de transferencia de datos que esté totalmente dentro del espacio de datos. Esto es posible con el uso de las instrucciones MOVWF y MOVFP, donde cualquiera p o f es especificado como INDF0 (o INDF1).

Si la fuente o destino del direccionamiento indirecto están en los bancos en la memoria, el acceso a la localidad será determinada por el valor en el BSR.

Un programa sencillo para limpiar la RAM de la 20h - FFh es mostrado en el ejemplo III.4

#### Ejemplo III. 4 Direccionamiento indirecto.

```

                MOVLW 0X20
                MOVWF FSR0           ;FSR0 = 20h
                BCF ALUSTA, FS1      ;INCREMENTO FSR
                BSF ALUSTA, FS0      ;DESPUES DEL ACCESO
                BCF ALUSTA, C        ;C = 0
                MOVLW END_RAM +1     ;
LF              INDF0               ;ADDR (FSR) = 0
CLRF

                CPFSEQ FSR0         ;FSR0 = END_RAM+1?
                GOTO LP             ;NO LIMPIA EL
                                   SIGUIENTE
                                   ; SI, TODA LA RAM
                                   ES LIMPIADA.

```

### III.1.J. TABLA DEL PUNTERO (TBLPTRL Y TBLLLPTRH)

El archivo de los registros TBLPTRL Y TBLLLPTRH forman un apuntador de 16 bits para direccionar los 64k de espacio del programa de memoria.

El apuntador de la tabla es utilizado por las instrucciones TABLWT y TABLRD. Las instrucciones TABLWT y TABLRD permiten las transferencias de datos entre el programa y los datos. El apuntador de la tabla sirve como una direccion de 16 bits de las palabras de datos dentro de la memoria del programa.

### III.1.K. MODULO DEL CONTADOR DEL PROGRAMA.

El contador del programa es un registro de 16 bits. PCL es el byte bajo del PC, mapeado en la memoria de datos. El PCL es de lectura y escritura, igual que cualquier otro registro. El PCH es el byte alto del PC y no es directamente direccionable; desde el PC- no se puede mapear en la memoria de datos ni en la del programa. Un registro de 8 bits PCLATH (PC high latch) es utilizado para retener en el latch el byte mayor del PC. El PCLATH esta mapeado en la memoria de datos.

El usuario puede leer o escribir en el PCH a través del PCLATH.

El PC es incrementado después de cada instrucción del fetch durante Q1 a menos que:

- Sea modificado por alguna instrucción GOTO, CALL, LCAL, RETURN, RETLW o RETFIE.
- Sea modificado por una respuesta de interrupción.
- Se destine como escritura al PCL por una instrucción.

Los saltos son equivalentes a forzar un ciclo NOP al saltar de dirección. Las figuras. III.21 y III.22 muestran la operación del contador del programa para varias situaciones.

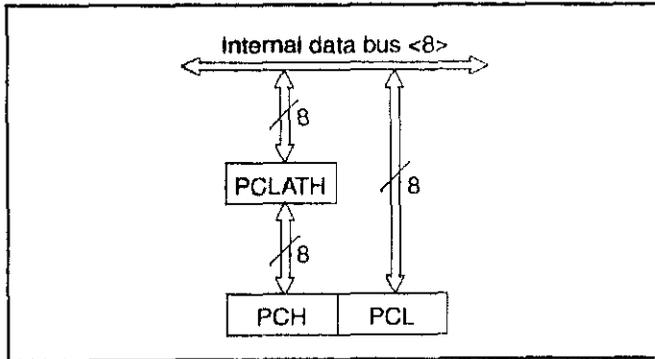


Figura III. 21 Operación del Contador del Programa.

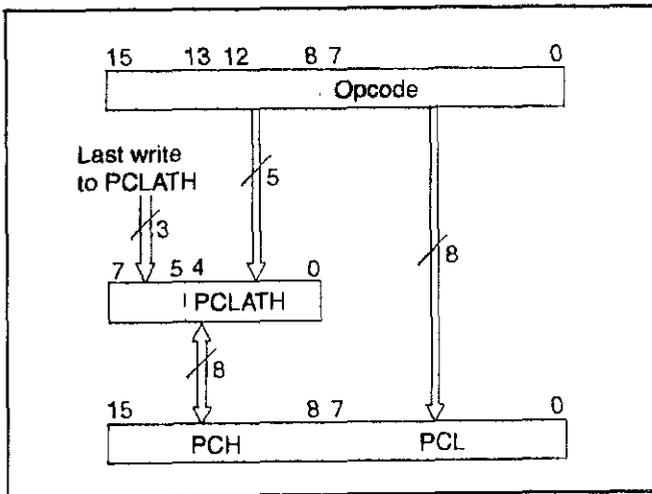


Figura III. 22 Contador del Programa usando las instrucciones Call y Goto.

### III.1.L. REGISTRO DE SELECCION DE BANCOS

El BSR es utilizado para conmutar entre los bancos y el área de la memoria de datos (Fig. III.1.A.11.1) El nibble bajo es utilizado para seleccionar los registros periféricos del banco. El nibble superior es utilizado para seleccionar el banco de memoria de propósito general.

Todo el Registro de Funciones Especiales (SFR) es mapeado dentro del espacio de memoria. Para acomodar un gran número de registros que se utiliza en el esquema de los bancos.

Un segmento del SFR va de la dirección 10h a la 17h; esta sección está localizada dentro de los bancos. El nibble bajo del banco de selección de registros (BSR) selecciona el "banco periférico", que es el que comúnmente está activado. Se han agrupado los registros periféricos para hacer más funcional el banco.

Por lo tanto, para intercambiar los registros de un banco a otro se necesitara una orden que realice el direccionamiento de todos los periféricos relacionados en una tarea sencilla.

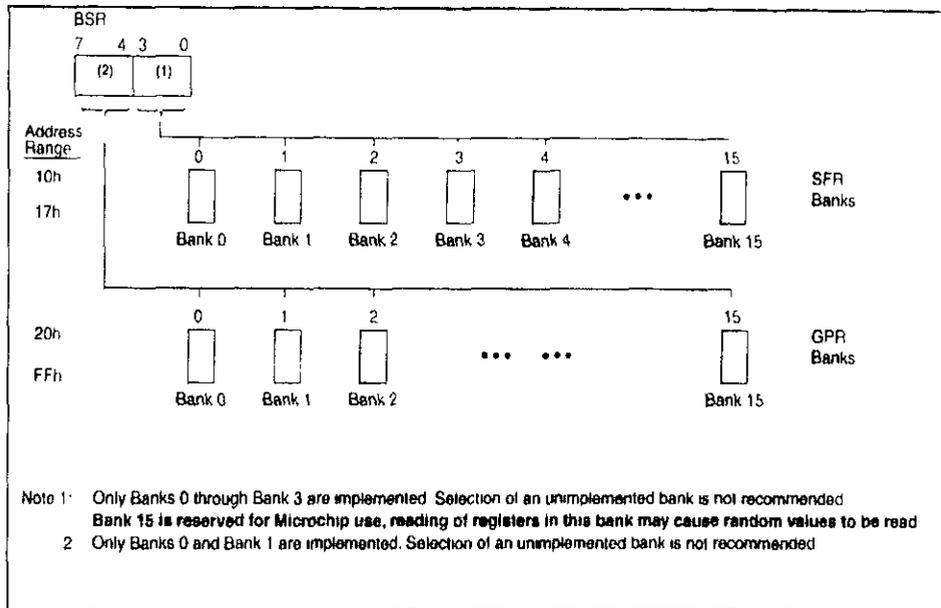


Figura III. 23 Operación BSR

### III.1.M. TABLA DE LECTURA Y ESCRITURA

El PIC17C4X tiene cuatro instrucciones que permiten al procesador mover los datos del espacio de memoria de datos al espacio de memoria de programa, y viceversa.

Desde el espacio de la memoria de programa (que es de 16 bits) y del espacio de memoria de datos (que es de 8 bits), se necesitan dos operaciones para mover 16 bits de valores de los datos de memoria. Las instrucciones TWLWT, t, f y TABLWT t, i, f son utilizadas para escribir los datos, del espacio de memoria de datos, al espacio de la memoria de programa. Las instrucciones

TLWT, TLW, TABLWT y TABLRD son utilizadas para escribir los datos, del espacio de memoria del programa, al espacio de memoria de datos.

La memoria de programa puede ser interna o externa. Para introducir la memoria de programa externa, el dispositivo necesita operar en los modos de microcontrolador extendido o microprocesador.

Las figuras de la III.24 a la III.27 muestran las operaciones de estas cuatro instrucciones.

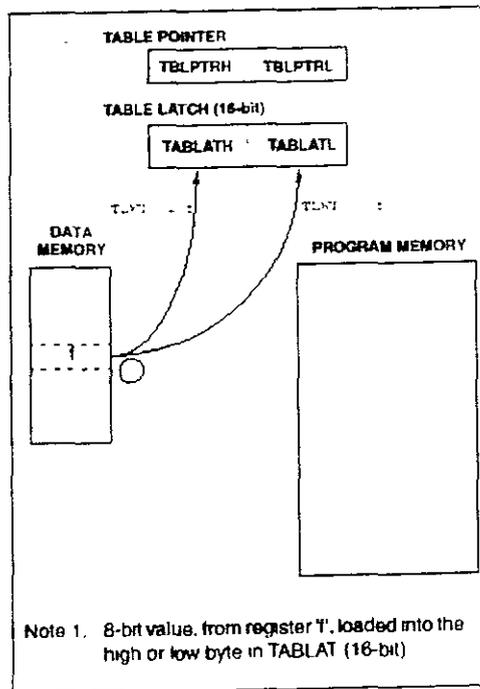


Figura III. 24 Instrucción de la operación TLWT.



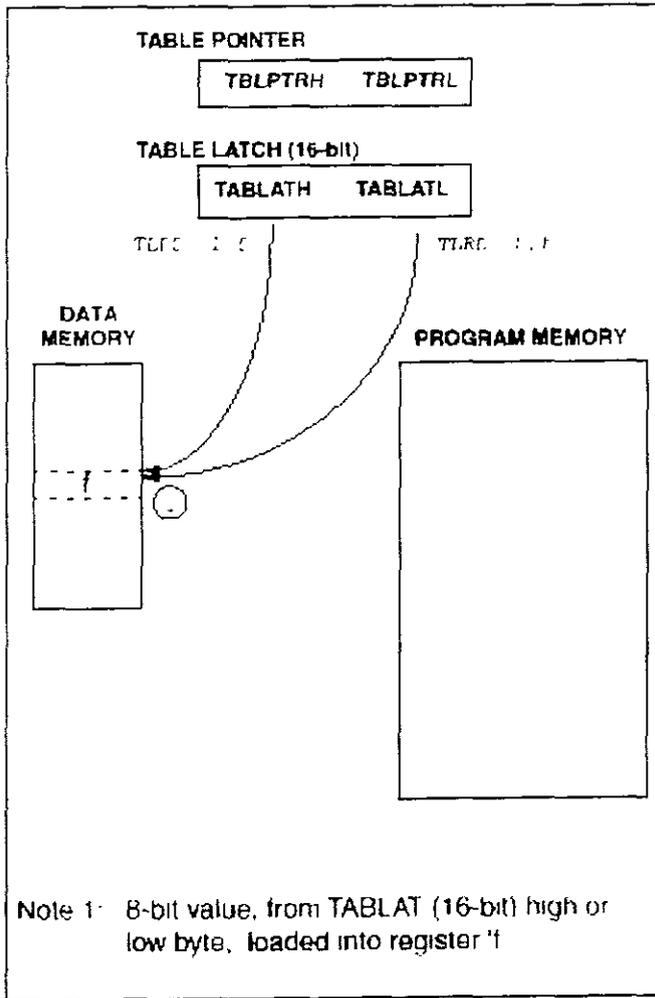


Figura III. 26 Instrucción de la operación TLRD.

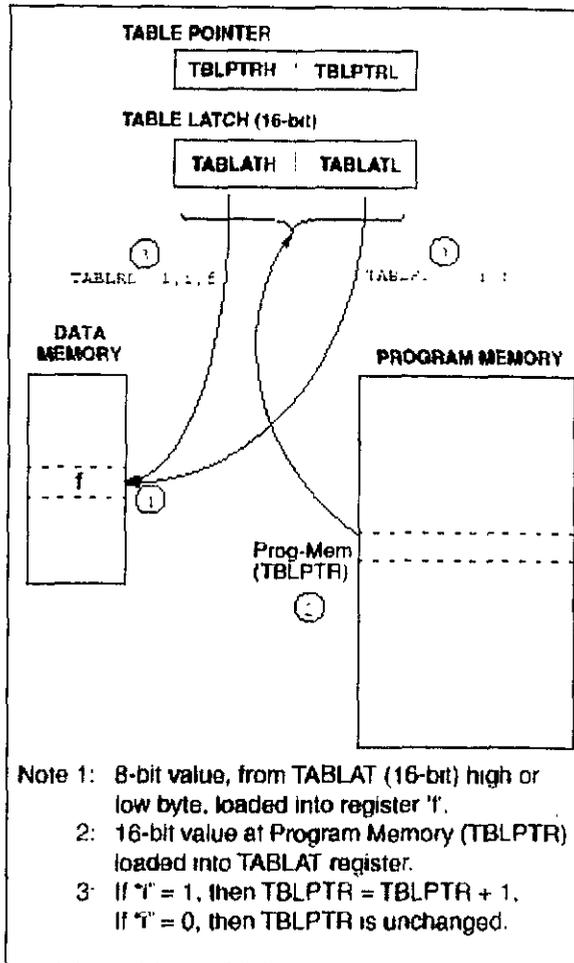


Figura III. 27 Instrucción de la operación TABLRD.

### **III.1.N. MULTIPLICADOR DE HARDWARE.**

El PIC17C4X tiene un multiplicador de hardware de 8x8 incluido en la ALU del dispositivo. Para realizar la operación de la multiplicación de hardware, se debe completar un ciclo de instrucción. Esto es, una multiplicación sin signo que da como resultado 16 bits. El resultado es almacenado dentro del registro de Producto que es de 16 bits (PRODH:PRODL). La multiplicación no afecta ninguna bandera en el registro ALUSTA.

Al realizar la multiplicación de 8x8 ejecutada en un solo ciclo se obtienen las siguientes características:

Alto rendimiento de procesamiento.

Reduce el tamaño de los códigos requeridos para la multiplicación de algoritmos.

Estas características permiten que este dispositivo sea utilizado en aplicaciones previamente reservadas para Procesadores Digitales de Señales.

### **III.1.O. PUERTOS DE ENTRADA/SALIDA.**

El PIC17C4X tiene cinco puertos de entrada/salida (I/O). Que van desde el PORTA al PORTE. El PORTB a través del PORTE tiene su correspondiente registro de dirección de datos (DDR), el registro DDR es el que se encarga de configurar los pines de los puertos como entradas o salidas. Estos cinco puertos están constituidos en 33 pines I/O. Algunos de estos pines de los puertos son multiplexados con alguna función alterna.

El PORTC, PORTD y el PORTE están multiplexados como el bus del sistema. Estos pines son configurados como el bus del sistema cuando la configuración de los bits del dispositivo es seleccionada en el modo de Microprocesador o Microcontrolador extendido. En los otros dos modos del microcontrolador, estos pines son de propósito general.

El PORTA y PORTB son multiplexados con las características periféricas del dispositivo. Estas características periféricas son:

- Módulos de los TIMERS
- Módulo de captura
- Módulo PWM
- Módulo del USART/SCI
- Pin de Interrupción Externa.

Cuando algunos de estos módulos periféricos son encendidos, el pin del puerto se configurará automáticamente para la función alterna. Los módulos que realizan esto son:

- Módulo PWM
- Módulo USART/SCI

Cuando un pin es configurado de manera automática como una salida por el módulo periférico, los pines de direcciones de datos desconocen al bit DDR. Después de inhabilitar al módulo periférico, el usuario debe reiniciar al bit DDR para la configuración que se desea. Los otros módulos periféricos (que requieren una entrada) deben tener configurado apropiadamente el bit de dirección de datos.

NOTA: Un pin que se configura como entrada periférica necesita un "1". Si se quiere utilizar como salida, tiene que ser limpiado con un "0".

Los eventos periféricos serán determinados por la acción de salida en el pin del puerto.

### III.1.0.1. REGISTRO DEL PORTA.

El PORTA es un latch de 6 bit. El PORTA no tiene su correspondiente Registro de Dirección de Datos (DDR).

Leyendo el PORTA, se leen los estados de los pines. El pin RA1 es multiplexado con la entrada de reloj TMRC, el RA4 y el RA5 son multiplexados con las funciones del USART. El control de RA4 y RA5 son salidas automáticamente configuradas por el módulo del USART.

#### III.1.0.1.1. UTILIZANDO RA2, RA3 COMO SALIDAS.

Los pines RA2 y RA3 son utilizados como salidas del "drain" (compuerta, abierta. Para utilizar los pines RA2 y RA3 como salidas, simplemente se necesita escribir el valor deseado en el registro del PORTA. Un "0" causará un nivel bajo en el pin y cuando se tiene "1" el pin quedará flotando (alta impedancia). Para poner un nivel alto al pin se deberán utilizar resistencias para proporcionar un nivel alto (pull-up).

Cuando se escribe en el PORTA no se ven afectados otros pines.

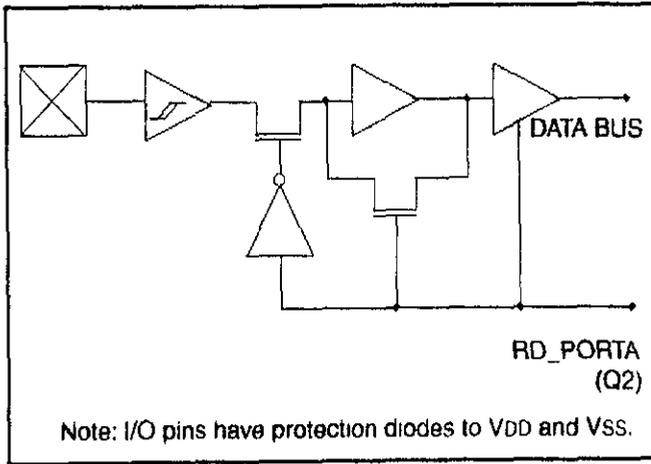


Figura III.28 Diagrama a bloques de RA0 y RA1.

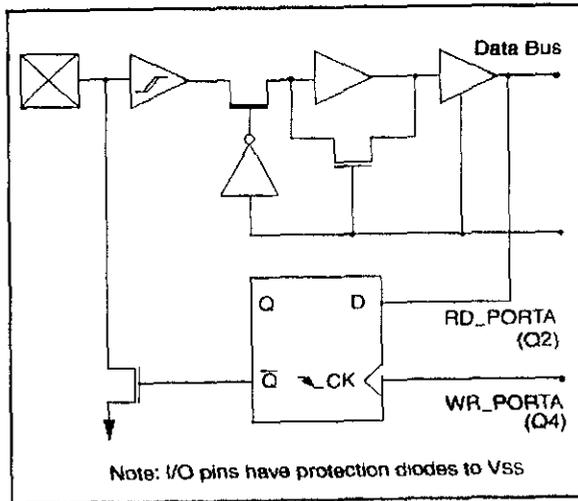


Figura III.29 Diagrama a bloques de RA2 y RA3.

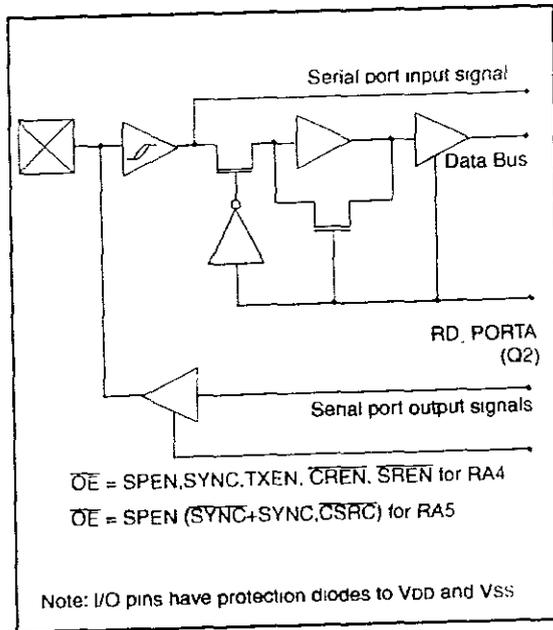


Figura III.30 Diagrama a bloques de R4 y R5.

Name	Bit0	Buffer Type	Function
RA0/INT	bit0	ST	Input or external interrupt input
RA1/T0CK1	bit1	ST	Input or clock input to the TMR0 timer/counter, and/or an external interrupt input
RA2	bit2	ST	Input/Output. Output is open drain type
RA3	bit3	ST	Input/Output. Output is open drain type
RA4/RX/DT	bit4	ST	Input or USART Asynchronous Receive or USART Synchronous Data
RA5/TX/CK	bit5	ST	Input or USART Asynchronous Transmit or USART Synchronous Clock
RBPU	bit7	—	Control bit for PORTB weak pull-ups.

Tabla III.11 Funciones del PORTA.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note 1)
10h, Bank 0	PORTA	$\overline{RBP}U$	-	RA5	RA4	RA3	RA2	RA1/TCLK	RA0/INT	0-xx xxxx	0-uu uuuu
05h, Unbanked	TOSTA	INTEDG	TOSE	TOS0	PS3	PS2	PS1	PS0	-	0000 000-	0000 000-
13h, Bank 0	RCSTA	SPEN	RC0	SREN	CREN	-	FERR	OERR	RC0D	0000 -00x	0000 -00-
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	-	-	TRMT	TX9D	0000 --1x	0000 --1-

Legend: x = unknown, u = unchanged, - = unimplemented reads as '0'. Shaded cells are not used by PORTA.

Note 1: Other (non power-up) resets include: external reset through  $\overline{MCLR}$  and the Watchdog Timer Reset.

Tabla III.12 Registros asociados con el PORTA.

### III.1.0.2. PORTB Y REGISTRO DDRB.

El PORTB es un puerto bidireccional de 8 bits. El correspondiente registro de dirección de datos es el DDRB. Un "1" en el DDRB configura el pin correspondiente del puerto como una entrada. Un "0" en el registro DDRB configura a los pines del puerto como salida. Cuando se lee al PORTB también se pueden leer los estados de los pines; considerando qué se está escribiendo, entonces se escribirá en el latch del puerto.

Cada pin del PORTB tiene una estructura interna de pull-up. Un bit de control sencillo puede encender todos los pull-ups. Esto se hace al limpiar el bit  $\overline{RBP}U$  (PORTA<7>). La característica de pull-up es automáticamente apagada cuando el pin del puerto es configurado como salida. Los pull-ups son habilitados con cualquier reset.

El PORTB tiene la característica de generar interrupciones. Únicamente cuando se configuran los pines como entrada pueden ocasionar que las interrupciones ocurran. Los pines de entrada (PE0:PE7) son comparados con los valores de los datos del latch del puerto. El desacopio de salidas de RB0:RB7 es una OR y al mismo tiempo genera en el PORTB la bandera de interrupción del RBIF (PIR<\*>).

Esta interrupción puede despertar al dispositivo del "sueño". Cuando una rutina esta corriendo y se genera la interrupcion el usuario, puede limpiar la interrupción por:

- Lectura - Escritura del PORTB (como es por las instrucciones MOVWF PORTB, PORTB). Esto será la condición de desacopio.
- Entonces, se limpia el bit RBIF.

Una condición de acoplamiento pondrá nuevamente al bit RBIF. Leyendo y escribiendo en el PORTB, finalizará la condición de acoplamiento y permitirá al bit RBIF que sea limpiado.

#### Ejemplo III.5 INICIO DEL PORTB.

```

MOVLE    0           ; Selección del Banco 0
CLRF     PORTB      ; Inicializa el PORTB y limpia la salida
                    ; de los Latches
MOVLW    0x0F       ; Valor utilizado para inicializar la
                    ; direccion de datos.
MOVWF    DDRE       ; Pone RB < 3:0 > como entradas
                    ; RB < 5: 4> como salidas
                    ; RB < 7:6 > como entradas

```

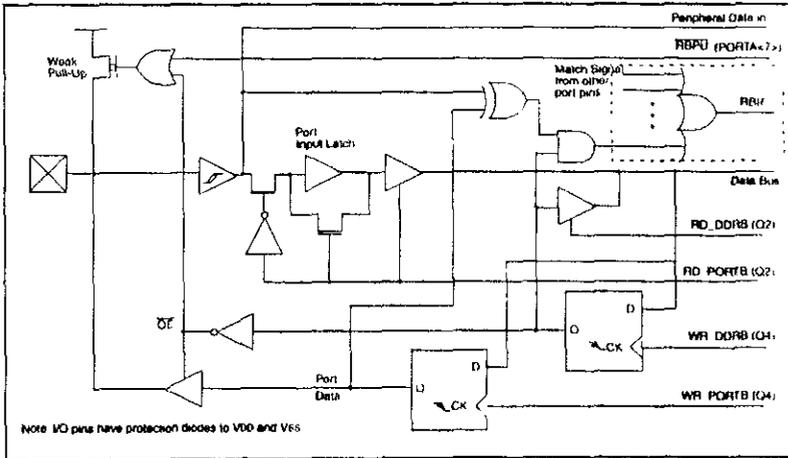


Figura III.31 Diagrama a bloques de RB <7:4> y RB <1:0> pines del puerto.

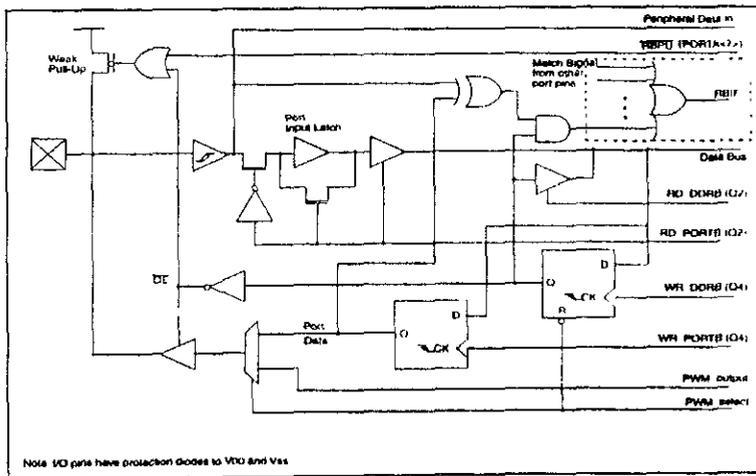


Figura III.32 Diagrama a bloques de los pines del PORTB RB3 y RB2.

Name	Bit	Buffer Type	Function
RB0/CAP1	bit0	ST	Input/Output or the RB0/CAP1 input pin. Software programmable weak pull-up and interrupt on change features
RB1/CAP2	bit1	ST	Input/Output or the RB1/CAP2 input pin. Software programmable weak pull-up and interrupt on change features
RB2/PWM1	bit2	ST	Input/Output or the RB2/PWM1 output pin. Software programmable weak pull-up and interrupt on change features
RB3/PWM2	bit3	ST	Input/Output or the RB3/PWM2 output pin. Software programmable weak pull-up and interrupt on change features
RB4/TCLK12	bit4	ST	Input/Output or the external clock input to Timer1 and Timer2. Software programmable weak pull-up and interrupt on change features
RB5/TCLK3	bit5	ST	Input/Output or the external clock input to Timer3. Software programmable weak pull-up and interrupt on change features
RB6	bit6	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features
RB7	bit7	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features

Tabla III.13 Funciones del PORTB.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
12h Bank 0	PORTB	PORTB data latch								xxxx xxxx	xxxx xxxx
11h Bank 0	DDRB	Data direction register for PORTB								xxxx xxxx	xxxx xxxx
10h Bank 0	PORTA	REPU	—	RAS	RAM	RA3	RA2	RA1/TCLK1	RA0/INT	xxxx xxxx	xxxx xxxx
09h Unbanked	CPUSTA	—	—	STKAV	GUIND	TO	PD	—	—	... ..	... ..
07h Unbanked	INTSTA	PEIF	TCLKIF	TOIF	BTIF	PEIE	TCLKIE	TOIE	INTIE	0000 0000	0000 0000
16h Bank 4	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TOIF	RCIF	0000 0100	0000 0000
17h Bank 4	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TOIE	RCIE	0000 0000	0000 0000
16h Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1PFS	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000

Legend x = unknown, u = unchanged, - = unimplemented read as '0', q = Value depends on condition

Shaded cells are not used by PORTB

Note 1 Other (non power-up) resets include: external reset through MCLR and the Watchdog Timer Reset

Tabla III.14 Registros y bits asociados al PORTB.

### III.1.0.3. PORTC Y REGISTROS DEL DDRC.

El PORTC es un puerto bidireccional. El correspondiente registro de dirección de datos es el DDRC. Un "1" en el DDRC configura el pin del puerto como una entrada. Un "0" configura al pin del puerto como salida. Leyendo el PORTC, se leen también los estados de los pines, considerando que se realiza la escritura, entonces se escribirá en el Latch. El PORTC es multiplexado con el bus del sistema.

Cuando opera como bus del sistema, el PORTC es el byte de mas bajo orden del bus dirección/datos (AD7:AD0).

NOTA: Este puerto es configurado como el bus del sistema cuando los bits configuran al dispositivo en el modo de Microprocesador o Microcontrolador Extendido. En los otros dos modos de Microcontrolador, este puerto es de propósito general de I/O.

El ejemplo III.6 Muestra la secuencia de inicio del PORTC. El Registro de Selección de Banco (BSR) debe ser seleccionado el banco 1 para iniciar al puerto.

Ejemplo III.6 Inicio del PORTC.

```
MOVLB      1           ; Selecciona al Banco 1
CLRF       PORTC      ; Inicializa el latch de datos del
                       PORTC antes de poner la configuracion
                       de los pines; datos en el registro
MOVLW      0xCF       ; Valor utilizado para inicializar la
                       dirección de datos
MOVWF      DDRC       ; Poner RC <3:0> como entradas
                       ; RC <5:4> como salidas
                       ; RC <7:6 > como entradas
```

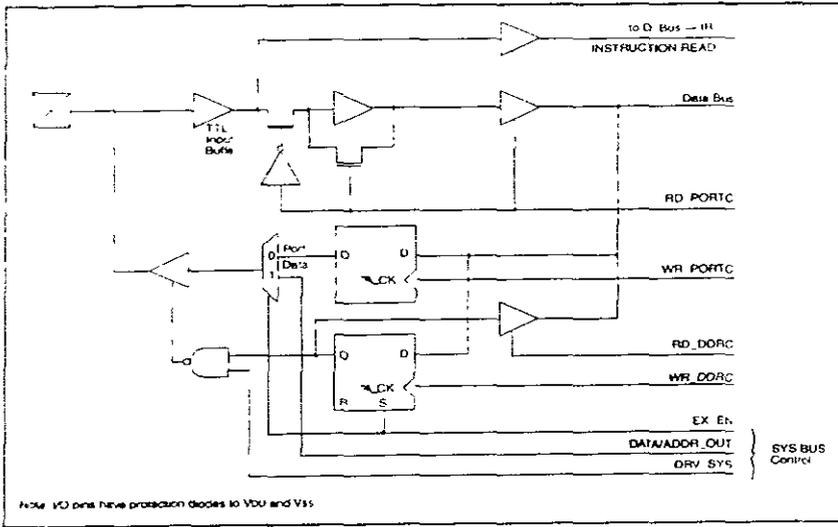


Figura III.33 Diagrama a bloques de los pines del PORTC

PC<R>0>

Name	Bit	Buffer Type	Function
RC0/AD0	bit0	TTL	Input/Output or system bus address/data pin
RC1/AD1	bit1	TTL	Input/Output or system bus address/data pin.
RC2/AD2	bit2	TTL	Input/Output or system bus address/data pin
RC3/AD3	bit3	TTL	Input/Output or system bus address/data pin
RC4/AD4	bit4	TTL	Input/Output or system bus address/data pin.
RC5/AD5	bit5	TTL	Input/Output or system bus address/data pin
RC6/AD6	bit6	TTL	Input/Output or system bus address/data pin.
RC7/AD7	bit7	TTL	Input/Output or system bus address/data pin.

Tabla III.14 Funciones del PORTC.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
10h, Bank 1	PORTC	RC7/ AD7	RC6/ AD6	RC5/ AD5	RC4/ AD4	RC3/ AD3	RC2/ AD2	RC1/ AD1	RC0/ AD0	xxxx xxxx	uuuu uuuu
10h, Bank 1	DDRC	Data direction register for PORTC								1111 1111	1111 1111

Legend: x = unknown, u = unchanged.

Note 1: Other (non power-up) resets include: external reset through  $\overline{MCLR}$  and the Watchdog Timer Reset.

Tabla III.15 Registros/bits asociados con el PORTC.

#### III.1.0.4. PORTD Y REGISTROS DDRD.

El PORTD es un puerto bidireccional. El correspondiente registro de dirección de datos es el DDRD. Un "1" en DDRD configura al pin del puerto como entrada. Un "0" en el DDRD configura al pin del puerto como salida. Leyendo el PORTD, se puede también leer el estado de los bits.

Considerando qué se está escribiendo, se escribirá en el latch del puerto. El PORTD es multiplexado con el bus del sistema. Cuando opera como bus del sistema, el PORTD es el byte de mayor orden del bus de dirección/datos (AD15:AD).

**NOTA:** Este puerto está configurado como el bus del sistema cuando los bits del dispositivo son configurados para seleccionar el modo de Microprocesador o Microcontrolador. En los otros dos

modos del microcontrolador, está puesto como de propósito general de I/O.

### Ejemplo III.7 INICIO DEL PORTD

```

MOVLE 1      ;Selección del banco 1
CLRF  PORTD ;Inicio los latch del PORTD, antes de poner la
            ;configuración de los pines;
MOVLW 0xCF  ;Se utiliza este valor para comenzar la
            ;dirección de los datos
MOVWF DDRD  ;pone RD<3:0> como entradas
            ;RD< 5:4 > como salidas
            ;RD < 7:6 > como entradas
    
```

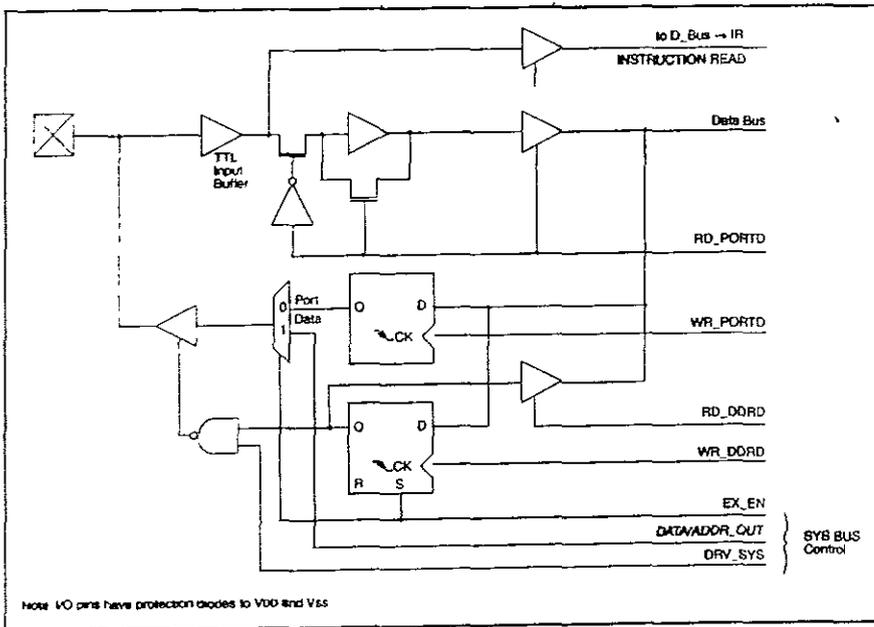


Figura III.34 Diagrama a bloques (Modo del PORTD I/O)

Name	Bit	Buffer Type	Function
RD0/AD8	bit0	TTL	Input/Output or system bus address/data pin.
RD1/AD9	bit1	TTL	Input/Output or system bus address/data pin.
RD2/AD10	bit2	TTL	Input/Output or system bus address/data pin.
RD3/AD11	bit3	TTL	Input/Output or system bus address/data pin.
RD4/AD12	bit4	TTL	Input/Output or system bus address/data pin.
RD5/AD13	bit5	TTL	Input/Output or system bus address/data pin.
RD6/AD14	bit6	TTL	Input/Output or system bus address/data pin.
RD7/AD15	bit7	TTL	Input/Output or system bus address/data pin.

Tabla III.16 Funciones del PORTD.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
13h, Bank 1	PORTD	RD7/AD15	RD6/AD14	RD5/AD13	RD4/AD12	RD3/AD11	RD2/AD10	RD1/AD9	RD0/AD8	XXXX XXXX	XXXX XXXX
12h, Bank 1	DDRD	Data direction register for PORTD								1111 1111	1111 1111

Tabla III.17 Registros/bits asociados con el PORTD.

### III.1.0.5. PORTE Y EL REGISTRO DDRE

El PORTE es un puerto bidireccional. El correspondiente registro de dirección de datos es el DDRD. Un "1" configura al pin correspondiente del puerto como entrada. Un "0" en el DDRD configura a los pines del puerto como salida. Al igual que los puertos anteriores, al leer en PORTE se pueden leer también el estado de los pines.

Considerando qué se está escribiendo, se escribirá realmente en el latch del puerto. El PORTE es multiplexado como bus del sistema.

El PORTE contiene la señal de control para el bus de dirección/datos (AD15:AD0). Estas señales de control son la habilitación de la dirección del Latch (ALE). La habilitación de salida ( $\overline{OE}$ ), y escritura ( $\overline{WR}$ ). La señal de control  $\overline{OE}$  y  $\overline{WR}$  son activadas con un nivel bajo.

**NOTA:** Este puerto es configurado como el bus del sistema cuando los bits del dispositivo son configurados en los modos de Microprocesador o Microcontrolador Extendido. En los otros dos modos del microcontrolador, este puerto es de propósito general de I/O.

Ejemplo III.8 Inicio del PORTE.

```
MOVLB    1      ;Selección del banco 1
CLRF     PORTE ;inicia el Latch de datos del PORTE antes de
              poner el registro ;
MOVLW    0x03   ;Este valor es utilizado para iniciar la
              dirección de datos
MOVWF    DDRE   ;Pone RE<1:0> como entrada
              ; RE <2> como salidas
              ;RE <7:3> son algunas veces de lectura como
              "0".
```

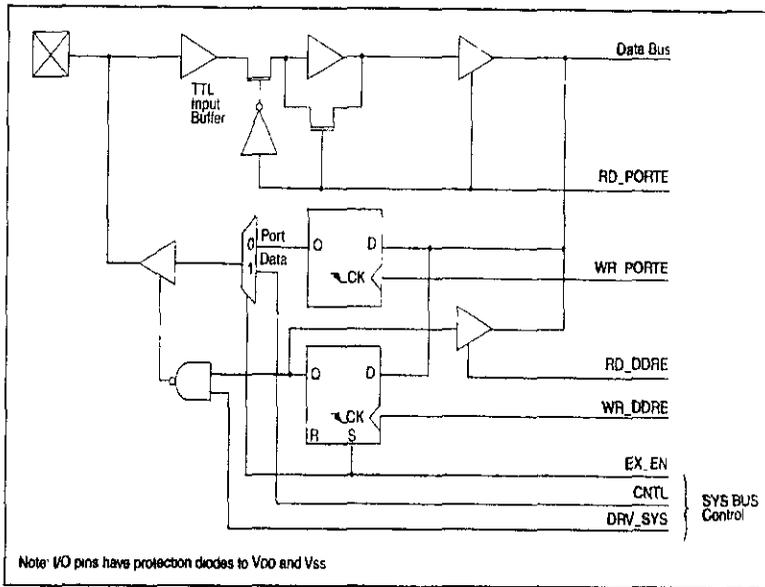


Figura III.35 Diagrama a bloques del PORTE (Modo de I/N).

Name	Bit	Buffer Type	Function
REQ/ALE	bit0	TTL	Input/Output or system bus Address Latch Enable (ALE) control pin.
RE1/OE	bit1	TTL	Input/Output or system bus Output Enable ( $\overline{OE}$ ) control pin.
RE2/WR	bit2	TTL	Input/Output or system bus Write ( $\overline{WP}$ ) control pin.

Tabla III.18 Funciones del PORTE.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note 1)
15h, Bank 1	PORTE						RE2WR	RE1OE	RE0VALE	--- -xxx	--- -uuu
14h, Bank 1	DDRE	Data direction register for PORTE								--- -111	--- -111

Tabla III.19 Registros/Bits asociados con el PORTE.

### III.1.0.6. CONSIDERACIONES DE PROGRAMACIÓN DE I/O.

Cualquier instrucción que es de escritura opera internamente como lectura seguida por una operación de escritura. Por ejemplo, las instrucciones BCF y BSF leen el registro dentro del CPU, ejecutan la operación del bit y escriben el resultado en la parte siguiente del registro.

Se debe tener mucho cuidado cuando estas instrucciones son aplicadas a un puerto con entradas y salidas definidas. Por ejemplo, la operación BSF enciende el bit 5 del PORTB y ocasionará que los 8 bits del PORTB sean utilizados para lectura dentro del CPU. Entonces la operación BSF toma lugar en el bit 5 y el PORTB estará escrito en los latches de salida. Si otro bit del puerto es utilizado como un pin bidireccional de I/O y es definido como una entrada al mismo tiempo, la señal de entrada presente en el pin será leída por sí misma dentro del CPU. Y será escrita nuevamente en el latch de datos de su pin particular, sobre escribiendo en él, el contenido previo, y el pin estará en espera en el modo de entrada sin ningún problema. De cualquier

modo, si el bit cero es conmutado después en el modo de salida, el contenido del latch de datos puede ser nuevamente desconocido. Leyendo nuevamente los valores de los pines del puerto, se conocerá la configuración de los latch del puerto. Cuando utilizamos las instrucciones de leer - modificar- escribir (BCF, BSF, BTG, etc.) en un puerto, el valor de los pines del puerto es leído, la operación deseada se desarrolla con este valor, y el valor es entonces escrito en el latch del puerto.

### **III.1.P. MÓDULO DE TRANSMISIÓN RECEPCIÓN UNIVERSAL ASÍNCRONO SÍNCRONO (USART).**

El módulo USART es un módulo serial de I/O. El USART puede ser configurado como un sistema asíncrono "full duplex" capaz de comunicarse con dispositivos periféricos, terminales CTR y computadoras personales, o también puede ser configurada como sistema síncrono "half duplex", que puede comunicarse con dispositivos periféricos y circuitos integrados A/D o D/A, EEPROM seriales etc. El USART puede ser configurado en los siguientes modos:

ASÍNCRONO (full duplex)

SÍNCRONO - AMO (half duplex)

SÍNCRONO - ESCLAVO (half duplex)

El bit SPEN (RCSTA<7>) tiene que ser puesto en orden para configurar RA4 y RA5 como una interconexión de comunicación serial.

El módulo del USART controlará las direcciones de los pines RA4/RX/DT y RA5/TX/CK, dependiendo del estado de los pines de configuración del USART en los registros RCSTA y TXSTA. Los bits de I/O controlan las direcciones de:

- SPEN
- TXEN
- SREN
- CREN
- CSRC

El estado de transmisión y los registros de control son mostrados en la figura. III.36, mientras que el Receptor de Estados y los registros de control son mostrados en la figura III.37

	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R-1	R/W-x
	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D
b.17								bit0

R = Readable bit  
W = Writable bit  
n = Value at POR reset  
(x = unknown)

b.17 CSRC: Clock Source Select bit  
Synchronous mode  
1 = Master Mode (Clock generated internally from BRG)  
0 = Slave mode (Clock from external source)  
Asynchronous mode  
Don't care

bit 6 TX9: 9-bit Transmit Enable bit  
1 = Selects 9-bit transmission  
0 = Selects 8-bit transmission

bit 5 TXEN: Transmit Enable bit  
1 = Transmit enabled  
0 = Transmit disabled  
SREN/CREN overrides TXEN in SYNC mode

bit 4 SYNC: USART mode Select bit  
(Synchronous/Asynchronous)  
1 = Synchronous mode  
0 = Asynchronous mode

bit 3-2 Unimplemented: Read as '0'

bit 1 TRMT: Transmit Shift Register (TSR) Empty bit  
1 = TSR empty  
0 = TSR full

bit 0 TX9D: 9th bit of transmit data (can be used to calculated the parity in software)

Figura III.36 Registro TXSTA (Dirección:15h, banco 0).

RW-0	RW-0	RW-0	RW-0	U-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D
bit 7							bit 0
<div style="border: 1px solid black; padding: 5px; display: inline-block;">           R = Readable bit            W = Writable bit            -n = Value at POR reset            (x = unknown)         </div>							
bit 7:	<b>SPEN: Serial Port Enable bit</b> 1 = Configures RA5/RX/DT and RA4/TX/CK pins as serial port pins 0 = Serial port disabled						
bit 6:	<b>RX9: 9-bit Receive Enable bit</b> 1 = Selects 9-bit reception 0 = Selects 8-bit reception						
bit 5:	<b>SREN: Single Receive Enable bit</b> This bit enables the reception of a single byte. After receiving the byte, this bit is automatically cleared. <u>Synchronous mode:</u> 1 = Enable reception 0 = Disable reception Note: This bit is ignored in synchronous slave reception. <u>Asynchronous mode:</u> Don't care						
bit 4:	<b>CREN: Continuous Receive Enable bit</b> This bit enables the continuous reception of serial data. <u>Asynchronous mode:</u> 1 = Enable reception 0 = Disables reception <u>Synchronous mode:</u> 1 = Enables continuous reception until CREN is cleared (CREN overrides SREN) 0 = Disables continuous reception						
bit 3:	<b>Unimplemented:</b> Read as '0'						
bit 2:	<b>FERR: Framing Error bit</b> 1 = Framing error (Updated by reading RCREG) 0 = No framing error						
bit 1:	<b>OERR: Overrun Error bit</b> 1 = Overrun (Cleared by clearing CREN) 0 = No overrun error						
bit 0:	<b>RX9D: 9th bit of receive data (can be the software calculated parity bit)</b>						

Figura III.37 Registro RCSTA (Dirección 13h, banco 0)

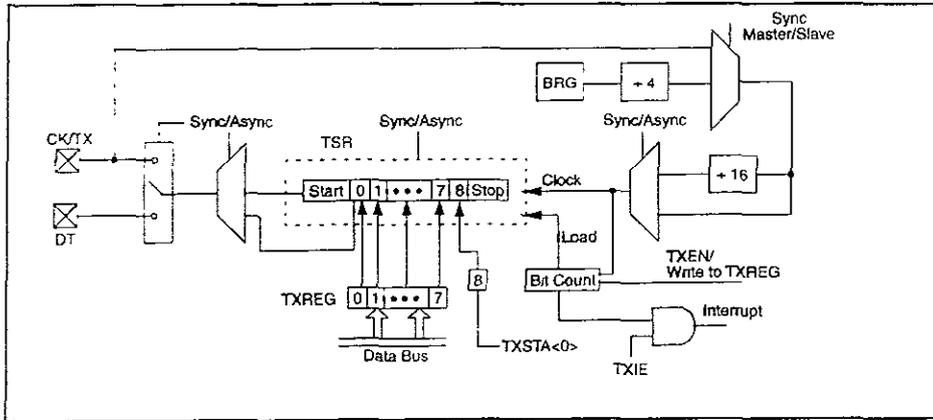


Figura III.38 Transmisor USART.

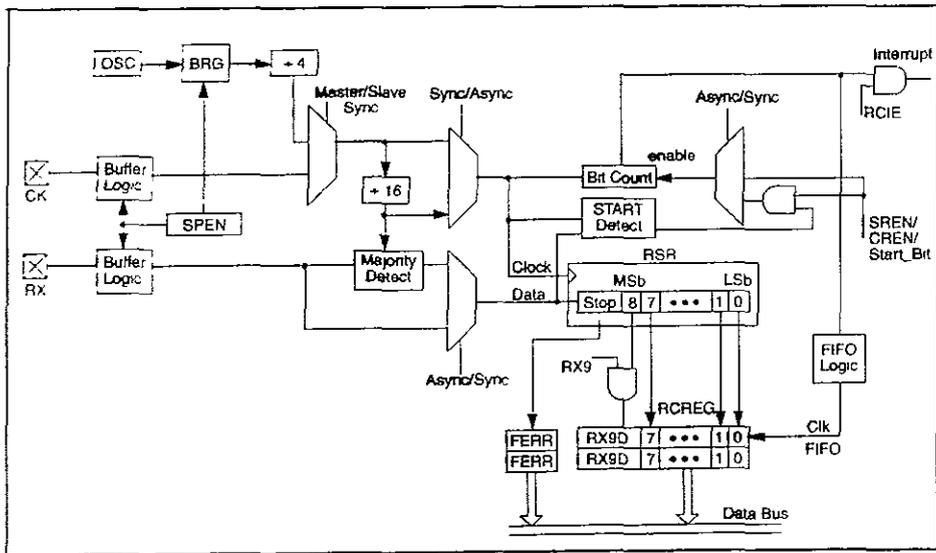


Figura III.39 Recepción USART.

### III.1.P.1. USART GENERADOR DE VELOCIDAD DE BAUDS (BRG) .

El BRG soporta el USART en el modo Asíncrono y Síncrono. Esto es, 8 bit dedicados al generador de velocidad de bauds del USART. El registro SPBRG controla el periodo de un temporizador de 8 bits que se ejecuta libremente. La tabla III.19 muestra las ecuaciones para el cálculo de la velocidad de bauds para diferentes modos del USART.

Esto únicamente se aplica cuando el USART está en el modo de maestro síncrono (reloj interno) y modo asíncrono. Al obtener la velocidad deseada de los bauds y la FOSC, el valor se debe de aproximar a un entero entre 0 y 255 y puede ser calculado utilizando la fórmula ya mencionada. El error en la velocidad de los bauds también puede ser determinado.

Tabla III.19 FORMULA DE VELOCIDAD DE BAUDS

SYNC	MODO	VELOCIDAD DE BAUDS
0	Asíncrono	$F_{osc}/(64(X+1))$
1	Síncrono	$F_{osc}/(4(X+1))$

X = Valor en SPBRG (0 a 255)

El ejemplo III.9 muestra el cálculo del error de la velocidad de bauds para las siguientes condiciones:

Fosc = 16 MHz.

Razón de Velocidad Deseada ≈ 9600

SYNC = 0

Ejemplo III.9 Cálculo del error de la velocidad de bauds.

$$\text{Velocidad de Bauds deseada} = F_{osc} / (64(X+1))$$

$$9600 = 16000000 / (64(X+1))$$

$$X = 25.024 \approx 25$$

$$\text{Velocidad de Bauds calculada} = 16000000 / (64(25+1))$$

$$= 9615$$

$$\text{Error} = \frac{\text{Velocidad de Bauds calculada} - \text{Velocidad de Bauds deseada}}{\text{Velocidad de Bauds deseada}}$$

$$= (9615 - 9600) / 9600$$

$$= 0.16\%$$

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
13h Bank 0	RCSTA	SPEN	<del>TX9</del>	<del>SREN</del>	<del>REN</del>	—	<del>ERR</del>	<del>ERR</del>	<del>TX9D</del>	0000 --0x	0000 --0u
15h Bank 0	TXSTA	CSRC	<del>TX9</del>	TXEN	SYNC	—	—	<del>TRMT</del>	<del>TX9D</del>	0000 --1x	0000 --1u
17h Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used by the Baud Rate Generator.

Note 1: Other (non power-up) resets include: external reset through  $\overline{MCLR}$  and Watchdog Timer Reset.

Tabla III.20 Registros Asociados con el generador de velocidad de Bauds.

BAUD RATE (K)	FOSC = 33 MHz			FOSC = 25 MHz			FOSC = 20 MHz			FOSC = 16 MHz		
	KBAUD	%ERROR	SPBRG value (decimal)									
0.3	NA	—	—									
1.2	NA	—	—									
2.4	NA	—	—									
9.6	NA	—	—									
19.2	NA	—	—	NA	—	—	19.53	+1.73	255	19.23	+0.16	207
76.8	77.10	+0.39	106	77.16	+0.47	80	76.92	+0.16	64	76.82	+0.16	51
96	95.93	-0.07	85	96.15	+0.16	64	96.15	+0.16	51	95.24	-0.79	41
300	294.64	-1.79	27	297.62	-0.79	20	294.1	-1.96	16	307.69	+2.56	12
500	485.20	-2.94	16	480.77	-3.85	12	500	0	9	500	0	7
HIGH	8250	—	0	6250	—	0	6000	—	0	4000	—	0
LOW	32.22	—	255	24.41	—	255	19.53	—	255	15.625	—	255

BAUD RATE (K)	FOSC = 10 MHz			FOSC = 7.169 MHz			FOSC = 5.088 MHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	9.766	+1.73	255	9.622	+0.23	185	9.6	0	131
19.2	19.23	+0.16	129	19.24	+0.23	92	19.2	0	65
76.8	75.76	-1.36	32	77.82	+1.32	22	79.2	+3.13	15
96	96.15	+0.16	25	94.20	-1.86	18	97.48	+1.54	12
300	312.5	+4.17	7	298.3	-0.57	5	316.8	+6.60	3
500	500	0	4	NA	—	—	NA	—	—
HIGH	2500	—	0	1789.8	—	0	1267	—	0
LOW	9.766	—	255	8.991	—	255	4.960	—	255

BAUD RATE (K)	FOSC = 3.579 MHz			FOSC = 1 MHz			FOSC = 22.768 kHz		
	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)	KBAUD	%ERROR	SPBRG value (decimal)
0.3	NA	—	—	NA	—	—	0.803	+1.14	26
1.2	NA	—	—	1.202	+0.16	207	1.170	-2.46	6
2.4	NA	—	—	2.404	+0.16	103	NA	—	—
9.6	9.622	+0.23	92	9.615	+0.16	25	NA	—	—
19.2	19.04	-0.83	46	19.24	+0.16	12	NA	—	—
76.8	74.57	-2.90	11	83.34	+8.51	2	NA	—	—
96	99.43	+3.57	8	NA	—	—	NA	—	—
300	298.3	-0.57	2	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	894.9	—	0	250	—	0	8.192	—	0
LOW	3.496	—	255	0.976	—	255	0.032	—	255

Tabla III.21 Velocidad de BAUDS para el modo síncrono.

BAUD RATE	IFOSC = 33 MHz			IFOSC = 25 MHz			IFOSC = 20 MHz			IFOSC = 18 MHz		
	KBAUD	%ERROR	SPBRG VALUE (decimal)									
0	NA	-	-									
1	NA	-	-	NA	-	-	1.22	+1.77	25	1.22	+0.10	25
2.4	2.35	-0.4	21	2.39	0.14	16	2.404	+0.16	125	2.414	+0.14	125
9.6	9.54	-0.6	5	9.5	-0.1	4	9.469	-1.36	30	9.61	+0.11	25
19.2	19.2	+0.04	26	19.53	+1.75	15	19.5	+1.7	15	19.21	+0.11	-
76.8	73.8	-4.04	5	78.13	+1.7	4	78.1	+1.5	3	83.31	+8.51	2
96	102	+7.4	4	97.6	+1.7	3	104	+8.1	2	NA	-	-
300	257	-14.0	-	300.6	+30.2	-	312.5	+4.1	-	NA	-	-
50	515.6	+2	-	NA	-	-	NA	-	-	NA	-	-
HIGH	515.6	-	-	-	-	-	312	-	-	25	-	-
LOW	20	-	25	10	-	25	1.22	-	25	0.9	-	25

BAUD RATE	IFOSC = 10 MHz			IFOSC = 7.159 MHz			IFOSC = 5.068 MHz			
	KBAUD	%ERROR	SPBRG VALUE (decimal)	KBAUD	%ERROR	SPBRG VALUE (decimal)	KBAUD	%ERROR	SPBRG VALUE (decimal)	
0	NA	-	-	NA	-	-	0.3	+3.1	25	
1	1.2	+0.1	12	1.2	-0.2	92	1	-	-	6
2.4	2.4	+0.1	6	2.36	-0.8	46	2.4	-	-	6
9.6	9.6	+1.1	-	9.32	-2.9	11	9.4	-3.1	-	-
19.2	19.2	+1.1	-	18.64	-2.9	5	19.8	+3.1	-	2
76.8	78.1	+1.7	-	NA	-	-	79.2	+3.1	-	-
96	NA	-	-	NA	-	-	NA	-	-	-
300	NA	-	-	NA	-	-	NA	-	-	-
50	NA	-	-	NA	-	-	NA	-	-	-
HIGH	157	-	-	111.5	-	-	79.2	-	-	-
LOW	0.6	-	25	0.43	-	25	0.306	-	255	-

BAUD RATE	IFOSC = 3.579 MHz			IFOSC = 3 MHz			IFOSC = 32.768 MHz			
	KBAUD	%ERROR	SPBRG VALUE (decimal)	KBAUD	%ERROR	SPBRG VALUE (decimal)	KBAUD	%ERROR	SPBRG VALUE (decimal)	
0	0.3	+0.2	18	0.30	+0.16	-	0.25	-14.3	-	-
1	1.19	-0.8	46	1.2	+0.16	-	NA	-	-	-
2.4	2.4	+1.3	23	2.23	-6.9	-	NA	-	-	-
9.6	9.3	-2.9	-	NA	-	-	NA	-	-	-
19.2	18.7	-2.9	-	NA	-	-	NA	-	-	-
76.8	NA	-	-	NA	-	-	NA	-	-	-
96	NA	-	-	NA	-	-	NA	-	-	-
300	NA	-	-	NA	-	-	NA	-	-	-
50	NA	-	-	NA	-	-	NA	-	-	-
HIGH	56.4	-	-	15.5	-	-	0.1	-	-	-
LOW	0.2	-	25	0.0	-	25	0.0	-	25	-

Tabla III.11 Velocidad de BAUD para el modo sincrónico

### III.1.P.2. MODO ASÍNCRONO DEL USART.

En este modo, el USART utiliza el formato estándar de no retorno a cero (NRZ), un bit de inicio, ocho o nueve bits de datos, y un bit de paro. El formato más común de datos es de 8 bits. Un chip generador de velocidad de bauds de 8 bits puede ser utilizado para derivar las frecuencias de velocidad del oscilador.

El transmisor y el receptor del USART son funcionalmente independientes, pero utilizan el mismo formato de datos y la misma velocidad de bauds. El generador de velocidad produce un reloj x64 de bits de velocidad de cambio. La paridad no es soportada por el hardware, pero puede ser diseñada en software y almacenada como el noveno bit de dato. En el modo asíncrono, es detenido durante el modo de "sueño".

El modo asíncrono es seleccionado al limpiar el bit SYNC (TXSTA<4>).

El módulo asíncrono USART consiste de los siguientes elementos importantes.

- **Generador de Velocidad de Bauds.**
- **Circuito de muestreo.**
- **Transmisor asíncrono.**
- **Receptor Asíncrono.**

El diagrama a bloques del transmisor del USART se muestra en la figura III.38. La parte más importante del transmisor es el registro de cambio (TSP). El registro de cambio obtiene los datos de lectura/escritura en el buffer del transmisor (TXREG). El TXREG es cargado con los datos en software. El TSP no es cargado hasta que el bit de paronaya sido transmitido de la carga. Como

falta poco para que el bit de paro sea transmitido, el TSR es cargado con los nuevos datos del registro del TXREG (si está disponible). Una vez que el TXREG transfiere los datos al TSR ocurre el ciclo al principio en el  $T_{CY}$  y termina en el BRG) el TXREG es desocupado con un bit de interrupción, y el bit TXIF (PIR<1>), es puesto. Esta interrupción puede ser habilitada o inhabilitada por el bit TXIE (PIE<1>).

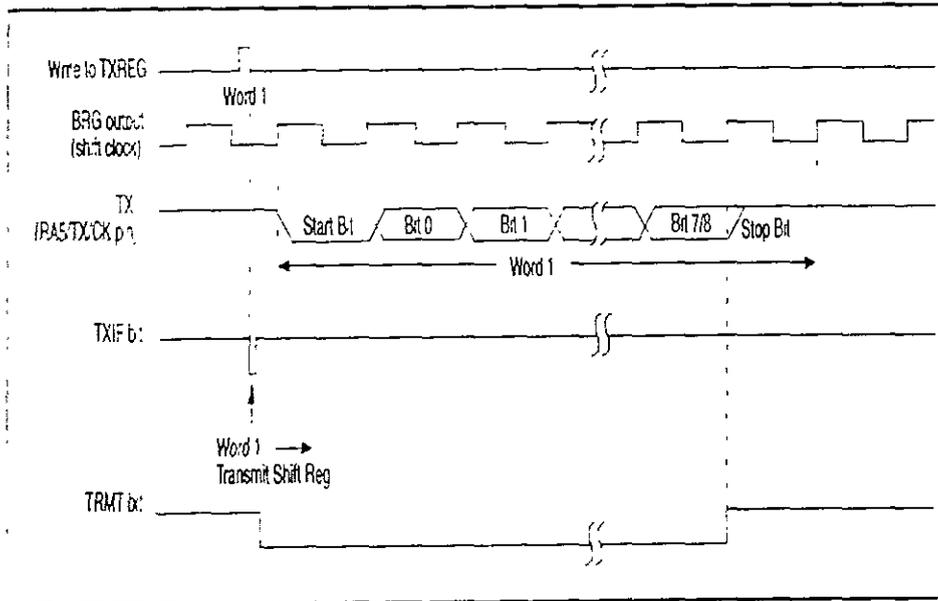


Figura III. 40 Transmisión asincrónica

En el Apéndice A se encuentran las instrucciones de programación del PIC.

Anteriormente, mencionamos que la tarjeta de adquisición de datos se compone de dos etapas que consideramos importantes:

- a) Sistema de adquisición de datos (SAD)
- b) Señal de control

El SAD permite la digitalización de la señal. Esta etapa se compone de los siguientes dispositivos electrónicos:

- Microcontrolador PIC 17C44
- Convertidor Analógico/Digital ADS901E
- Contadores en cascada 74F191
- Memoria SRAM HM66205

La etapa de la señal de control está compuesta de:

- Comparador LM311
- Transistor 2N2222A

A continuación, se hará una descripción de cada dispositivo y su función dentro de cada etapa. Comenzaremos describiendo el convertidor ADS901E.

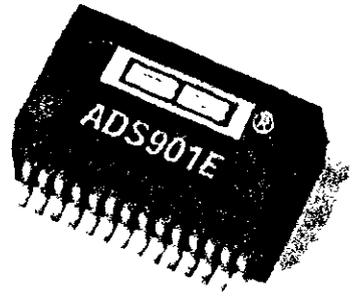


Figura III.2.4 ADS901E

### III.2.A. DESCRIPCION.

El ADS901E es un velocizador convertidor analógico a digital de entubamiento que opera con una alimentación de 5V. Este convertidor ofrece un ancho de banda completo de transmisión y claridad de detalle. La escala completa de conversión de entrada se define por referencias externas.

El ADS901E emplea las técnicas de corriente de entrada digital para proporcionar una linealidad diferencial excelente. Por ejemplo, para aplicaciones de imágenes, el dispositivo ofrece un alto SNR con el margen necesario para telecomunicaciones, video y aplicaciones de instrumentación.

Los ADS901E están disponibles en empaquetados de 16-pines, en SSOP.

### **Características Generales**

10-Bit, 20MHz, Voltaje de polarización +V  
Convertidor analógico-a-digital

### **Aplicaciones**

- Equipo de potencia de baterías
- Video grabadoras
- Cámaras digitales
- Scanners para computadoras
- Comunicaciones

### **III.2.B. CARACTERÍSTICAS.**

- Bajo consumo de energía: 48mW a +V
- Rango de alimentación de: +2.7V a +3.7V
- Rango completo de escala de digitalización ajustable por referencias externas.

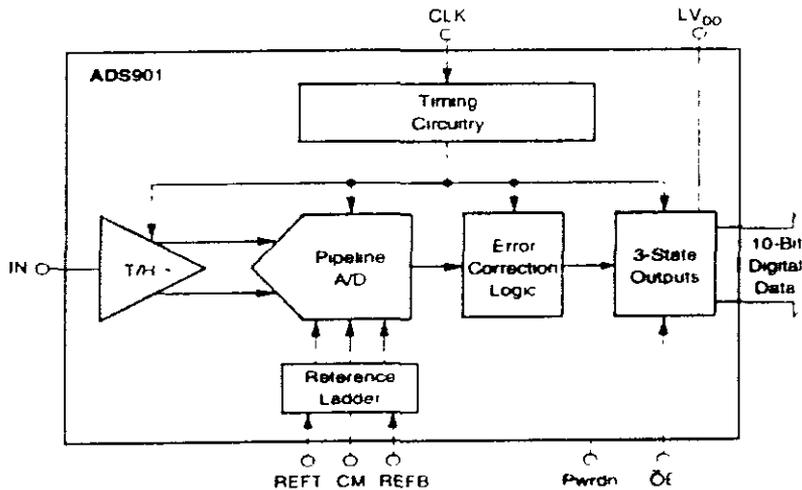


Figura III.4. Diagrama a bloques del Convertidor A/D.

### III.2.C. TEORÍA DE FUNCIONAMIENTO.

Los ADS901E capturan a alta velocidad la conversión de analógico a digital. El convertidor utiliza una arquitectura de entubamiento. La topología totalmente diferencial y la garantía de corrección de error digital da una resolución de 10-bits. El circuito de track/hold diferencial se muestra en la figura III.4a.

Los interruptores son controlados por un reloj interior que tiene los dos signos de la fase que no se traslapan,  $\phi_1$  ,  $\phi_2$

En el tiempo de muestreo, la señal de entrada se prueba en el fondo de las placas de los capacitores de entrada. En la proxima fase del reloj  $\phi_1$ , las placas del fondo de los capacitores de entrada se conectan juntos a través de la retroalimentación de los capacitores, que controlan los interruptores de la salida en

el amplificador operacional. En ese momento, las cargas se redistribuyen entre  $C_i$  y  $C_H$ , completando un ciclo de track/hold. La salida diferencial es una representación de  $V_i$  sostenida en la entrada analógica durante el tiempo de muestra. El circuit de track/hold puede convertir la señal de la entrada en una totalmente diferencial para cuantizar. Por consiguiente, se puede determinar la influencia de la señal de ruido a la entrada, y esta fase también se definen otros parámetros con la de señal pequeña, es. ancho de banda para potencia total y el ancho de banda del ruido.

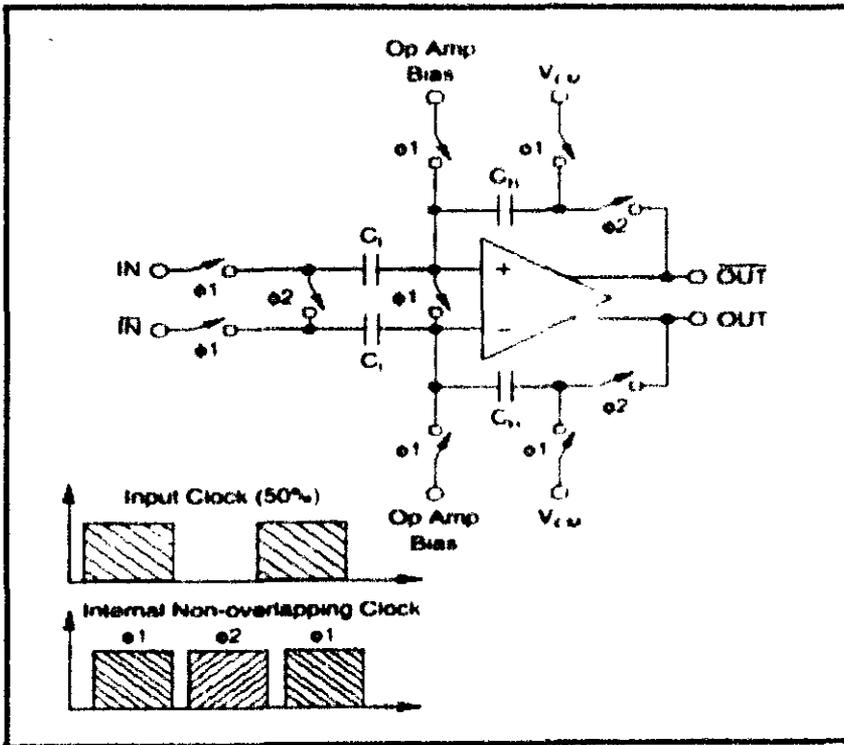


Figura III.4.3 Configuración de entrada de Track/Hold con diagrama de tiempos.

La arquitectura de cuantizar en entubamiento tiene 9 estados. Cada estado tiene que contabilizar dos bits y convertir dos bits de digital a analógico, como se muestra en la Figura III.43. Cada dos bits se contabilizan al estado siguiente del borde de un subreloj, que es de la misma frecuencia del reloj externamente aplicado.

La salida del contador se alimenta con su propia línea de retraso para encuadrarlo en tiempo con los datos creados por los siguientes estados de la cuenta. Estos datos se alinean dentro del circuito de corrección de error digital. En él se puede ajustar la base de datos del rendimiento a la información encontrada en los bits redundantes. Esta técnica proporciona al ADS901E una excelente linealidad diferencial y garantiza que ningún código será perdido en los 10-bits de nivel.

Acomodando una señal bipolar balanceada, los ADS901E operan con un voltaje de modo común ( $V_{CM}$ ) que se deriva de las referencias externas. Debido a las resistencias simétricas de escalera dentro de los ADS901E, el  $V_{CM}$  se sitúa entre el voltaje superior e inferior de referencia. La ecuación III.1 puede usarse para calcular el nivel de voltaje en modo común.

$$V_{CM} = (REF1 + REF2) / 2 \dots\dots\dots EC. III.1.$$

### **III.2.D. SALIDA DE DATOS DIGITALES.**

Los 10 bits de salida son proporcionados por niveles lógicos CMOS. Existen 5 ciclos de reloj de datos latentes al comenzar la señal convertida para la salida de los datos válidos. La salida se codifica en el estándar "Straight Offset Binary", donde la

señal a escala completa de la señal digitalizada corresponde a "1" en la salida.

La salida digital del ADS901E puede ser puesta en el estado de alta impedancia para manejar los tres estados (pin 16) con un nivel lógico alto. Debido a las resistencias internas de "pull down", el funcionamiento normal se logra con el pin 16 "bajo" o de flotación. Esta función se mantiene para pruebas de propósitos de estabilidad pero no se recomienda para manejar cargas capacitivas mayores que 15pF.

### **III.2.E. SEÑAL BALANCEADA Y MODO COMÚN.**

#### **III.2.E.1.1. CONSIDERACIONES.**

Los ADS901E se diseñan para operar en un voltaje de suministro de sólo +3V. La entrada nominal de la señal es de 1Vp-p, situada entre +1V y +2V. Esto significa que el nivel de la señal está entre 0.5V alrededor de un voltaje de modo común de +1.5V, que es la mitad del voltaje suministrado ( $V_{CM} = V_s / 2$ ).

En algunas aplicaciones podría ser ventajoso aumentar el nivel de la señal de entrada. Esto mejora el rechazo a la señal de ruido. Sin embargo, debe conservarse el nivel de la señal dentro del rango lineal de funcionamiento del circuito para evitar cualquier distorsión excesiva. En situaciones extremas el desempeño del convertidor empezará a degradarse debido a las variaciones del interruptor de la resistencia de entrada por encima del voltaje de la entrada. Por consiguiente, durante el funcionamiento normal el nivel de la señal debe seguir siendo aproximadamente 0.5V, lejos de cada vía de polarización.

### III.2.F. MANEJO DE LAS ENTRADAS ANALÓGICAS.

#### III.2.F.1.1. MANEJO DE ACOPLAMIENTO DE AC.

La figura III.44 muestra un ejemplo de acoplamiento de AC, utilizando un amplificador operacional de alta velocidad que opera con polarizaciones duales (OPA650, OPA658). Puede manejar sin problemas la referencia de voltaje medio,  $V_{CM}$ , gracias a las polarizaciones bipolares. El capacitor  $C_1$  y la resistencia  $R_1$  forman un filtro paso altas de  $-3dB$ .

Fig. III.44 - Acoplamiento de AC, circuito de interface de Single - Endes.

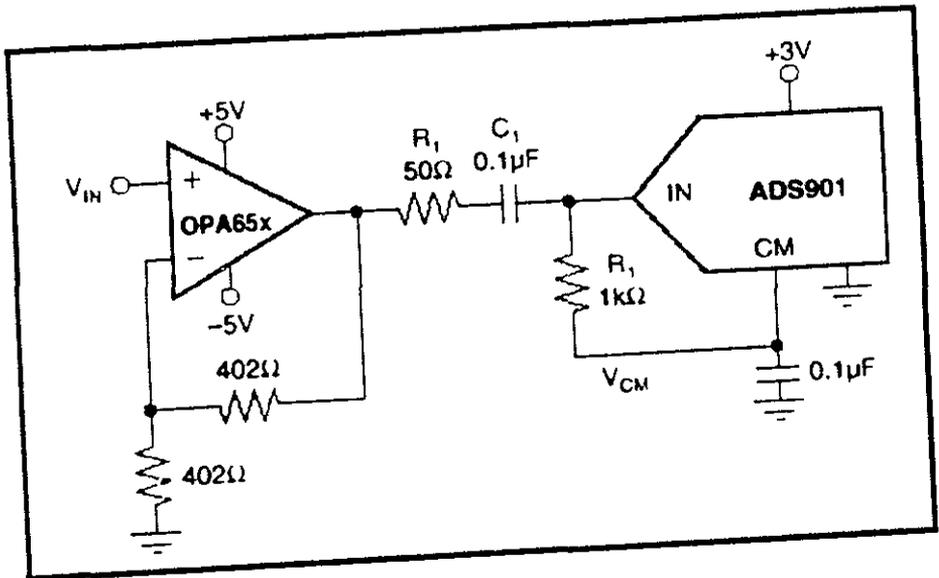


Figura III.44 Acoplamiento de AC, circuito de interface de Single - Endes.

Los valores para  $C_1$  y  $R_1$  no son críticos en la mayoría de las aplicaciones y pueden seleccionarse libremente, de acuerdo a los parámetros de diseño del filtro que se quiera tener. Los valores mostrados corresponden a una frecuencia de 1.6kHz.

La Figura III.45 muestra un circuito que puede utilizarse con una sola fuente. La fuente de 5 Volts polariza al amplificador operacional hasta aproximarse al voltaje de modo común apropiado, por ejemplo,  $V_{cm} = +1.5V$ . Con el uso de capacitor  $C_0$ , la ganancia en DC para la entrada no invertida del amplificador operacional se pone a  $+1V/V$ . Como resultado de la función de transferencia modificada:

$$V_{out} = V_{in} \left\{ (1 + R_f / R_0) + V_{cm} \right\} \dots \dots \dots \text{EC. III.2}$$

La entrada que acopla al capacitor  $C_1$  y la resistencia  $R_1$  forma un filtro de paso altas. Al mismo tiempo, la impedancia de la entrada es definida por  $R_1$ . La resistencia  $R_2$  aísla la salida del amplificador operacional de las cargas capacitivas para evitar los picos de ganancia o posibles oscilaciones. También puede usarse para establecer un ancho de banda definido que reduzca el ruido. El valor recomendado normalmente es entre 10 $\Omega$  y 100 $\Omega$ .

### III.2.F.1.2. CIRCUITO DE INTERFACE DE ACOPLAMIENTO DE DC.

En la figura III.46 se muestra una fuente simple para polarizar al convertidor. El circuito de acoplamiento de DC puede ser seleccionado con una ganancia de  $-1V/V$  o mayor y depende de la ganancia que proporciona el divisor de voltaje dado por las resistencias  $R_1$  y  $R_2$ . Este circuito debe ser ajustado para proporcionar un voltaje de modo común adecuado para el ADS901E. El valor del voltaje suministrado debe ser de  $+3V$ . El rango de la entrada recomendado para el ADS901E es  $1Vp-p$ . Este rango se cuenta a partir del voltaje de modo común, que es de  $+1.5V$ . De este voltaje de modo común puede derivarse el voltaje de las referencias externas.

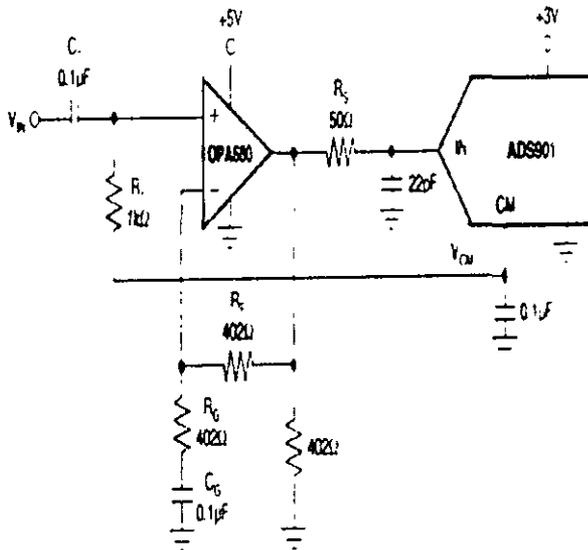


Figura III.45 Circuito de Interface de Alterna con una sola fuente de polarización.

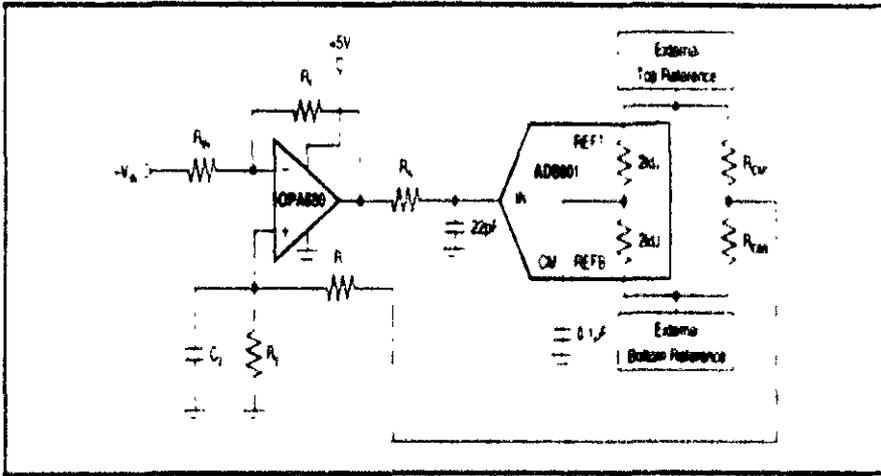


Figura III.4e Circuito de Acoplamiento de I<sub>REF</sub> con una sola fuente de polarización.

### III.2.F.1.3. REFERENCIA EXTERNA

Los ADS90C01E requieren referencias externas (pin 20 (REF+) y 24 (REF-)). Internamente estos pines se conectan a través de una resistencias en serie que suman una resistencia nominal de 4kΩ (1kΩ). Para establecer una caída de voltaje correcta en el arreglo de las resistencias de referencias, este arreglo debe ser capaz de proporcionar típicamente 250mA de corriente. Con esta corriente el rango completo de entrada del ADS90C01E se fija entre +1V y -1V, o 1Vp-p. En general, la diferencia de voltaje por REF+ y REF- determina la entrada del rango máximo (FSR) del ADS90C01E. La ecuación III.3 puede usarse para calcular el span.

$$FSR = REF+ - REF- \dots \dots \dots \text{EC. III.3}$$

Dependiendo de la aplicación, son posibles varias opciones para proporcionar los voltajes de referencia externos a los ADS901E.

### III.2.F.1.4. SOLUCIÓN DE LA REFERENCIA DE MANERA ECONÓMICA.

En la Figura III.47 se muestra la manera mas sencilla de lograr los voltajes de referencia requeridos para polarizar las resistencias de referencia en serie del ADS901E desde las fuentes de alimentación. Se necesitan dos resistencias adicionales ( $R_1$ ,  $R_2$ ) para que la corriente fluya correctamente a través del arreglo de resistencias. Sin embargo, los valores de la resistencia podran ser calculados para un buen funcionamiento del convertidor tomando en cuenta el rango máximo deseado y el voltaje suministrado por la fuente de alimentación.

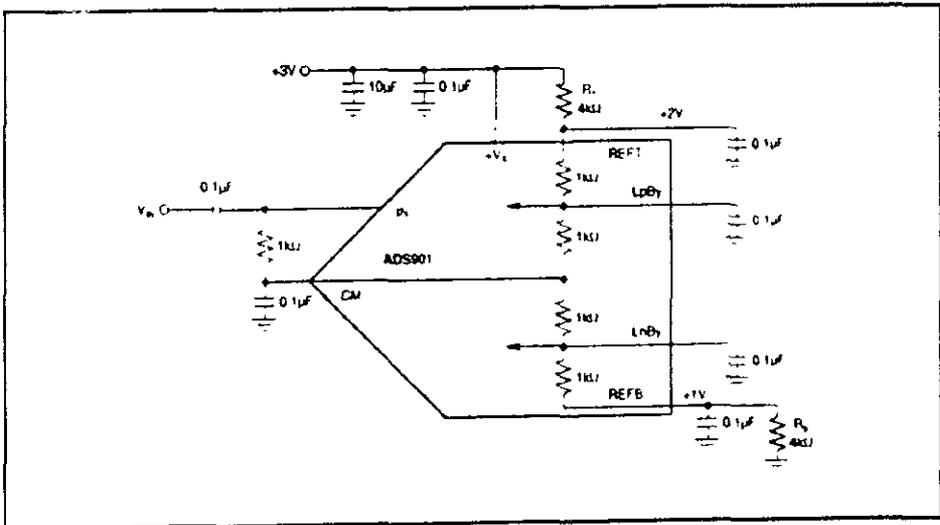


Figura III.47 Solución de bajo costo para polarizar al convertidor con resistencias externas.

Esta configuración es la más recomendable por los pocos elementos que necesita para dar la referencia de voltaje, además de que presenta una pequeña variación en el voltaje de referencia. Las variaciones en los voltajes de referencia se deben a la tolerancia de los componentes y a las variaciones del suministro de la fuente de poder. En todo caso, se recomiendan capacitores cerámicos de desvío por lo menos de  $0.1 \mu\text{F}$  colocados en los pines de las referencias, como se muestra en la Figura III.47. Los capacitores tienen un propósito dual. Ellos desviarán la mayor parte del ruido transitorio, resultado de la retroalimentación del reloj y de la señal de ruido causada por los interruptores de las fases de T/H. Asimismo, sirven como un depósito de carga para proporcionar corriente instantánea a los nodos internos.

#### **III.2.F.1.5. SOLUCIÓN A LA REFERENCIA EXACTA.**

Para aplicaciones que requieren un nivel más alto de exactitud en las referencias de entrada del convertidor, se requiere un circuito que dé una referencia precisa, como se muestra en la Figura III.48. Una referencia de voltaje estable de  $+1.2\text{V}$  se obtiene con un diodo Zener de dos terminales, por ejemplo el REF1004-1.2, así como empleando un amplificador operacional dual de propósito general que se polariza con una sola fuente (A1), como el OPA2237, OPA2234 o MC34072. Los dos amplificadores que se requieren para proporcionar los voltajes de referencia al ADS901E pueden ser generados poniendo cada operacional con la ganancia apropiada; por ejemplo: REFT fijo a  $+2\text{V}$  y REFB a  $+1\text{V}$ .

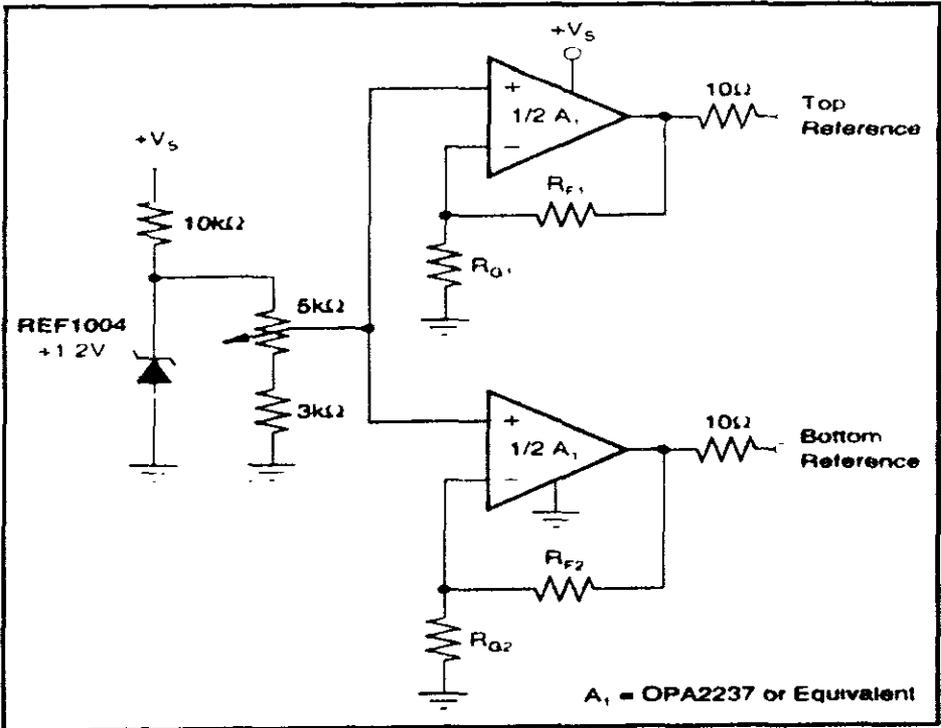


Figura III.4- Referencia de Voltaje de una forma más exacta por medio de un diodo Zener.

### III.2.G. ENTRADA DE RELOJ.

La entrada de reloj de los ADS901E se diseña para operar de  $+1V$  a  $+3V$  con niveles lógicos CMOS. Se utiliza la alta velocidad y la avanzada lógica CMOS (HC/HCT, AC/ACT) para manejar el reloj de entrada con una cantidad mínima de variación del ciclo de trabajo y soportar una velocidad de muestreo máxima (20MHz). Cuando se digitaliza a una alta velocidad de muestreo, al 50% del ciclo de trabajo de reloj con levantamiento rápido y con caída en un

tiempo de 2ns o menos, se recomienda conocer las especificaciones de desempeño del dispositivo o del circuito que nos proporcionará esta señal. El ADS901E es tolerante a las variaciones del ciclo de trabajo en un ±10% sin presentar degradación.

Para aplicaciones que operan a frecuencias altas se debe tomar en cuenta la teoría de muestreo de Nyquist, así como consideraciones especiales para proporcionar un reloj con muy pocas variaciones.

Las variaciones de reloj ocasionan también variaciones de apertura ( $t_{UN}$ ), que son uno de los obstáculos para lograr un buen funcionamiento del SNR. La ecuación III.4' muestra la relación entre la variación de la apertura, la frecuencia de entrada y la proporción de la señal de ruido:

$$SNR = 20 \log_{10} [1 / (2 \pi F_{IN} t_{UN})] \dots \text{EC. III.4}$$

Por ejemplo, el SNR es fijado a 58dB con 10MHz y una señal de entrada a escala máxima con una variación de apertura de  $t_{UN} = 20 \times 10^{-12}$ .

### III.2.H. SEÑALES DIGITALES.

Las salidas digitales de los ADS901E son estándares de estados CMOS normales y diseñados para ser compatibles con las familias lógicas TTL y CMOS, ambas de alta velocidad. Los umbrales de la lógica son para bajo voltaje CMOS:  $V_{OL} = 0.4V$ ,  $V_{OH} = 2.4V$ , que permiten a los ADS901E conectarse directamente a la lógica de 3V.

La salida digital del ADS901E utiliza un pin de suministro digital dedicado (pin 2, LV DD). Ajustando el voltaje en LV DD,

los niveles de salida digital variarán respectivamente. En todo caso, se recomienda limitar el fan - out a uno, manteniendo la carga capacitiva debajo de la especificación de 15pF.

Si es necesario, pueden utilizarse buffers o latches externos para obtener los beneficios adicionales de aislar al convertidor A/D de cualquier actividad digital, ayudar al bus a realizar el acoplamiento a las altas frecuencias del ruido y degradar su funcionalidad dentro del convertidor.

### **III.2.1. MODO DE BAJO CONSUMO.**

El consumo de energía del ADS901E puede ser muy reducido. Para esto, el pin Pwrdn-pin (pin 17) debe conectarse a una lógica de nivel "alto", disminuyendo la corriente de suministro en un 70% aproximadamente. En su funcionamiento normal, el modo de bajo consumo es inhabilitado por una resistencia de pull - down (50k $\Omega$ ). Durante el modo de bajo consumo, la salida digital se encuentra fija en tres estados.

## III.2.J. DESACOPLAMIENTO Y CONEXIONES A TIERRA.

### III.2.J.1.1. CONSIDERACIONES.

El convertidor ADS901E tiene algunos pines que le suministran energía. Algunos se dedican a proporcionar un manejo de alto rendimiento de energía. Y los pines de suministro restantes separan el suministro analógico del digital, sino que se filtra desde que son conectados internamente en el chip. Por esta razón se recomienda tratar al convertidor como un componente analógico y acoplarlo únicamente a la fuente analógica. Las líneas de alimentación digital llevan a menudo niveles altos de señal de ruido que se pueden "meter" al convertidor y afectar su funcionamiento.

Debido a la arquitectura de entubamiento, el convertidor también origina ruido y transitorios de alta frecuencia que se alimentan de la retroalimentación y de las líneas de referencia. Para resolver esto se requiere que los pines de alimentación y los pines de referencia sean desacoplados adecuadamente.

La Figura III.49 muestra los desacoplamientos sugeridos. En la mayoría de los casos, los capacitores cerámicos de 0.1µF son recomendados para mantener la baja impedancia sobre el rango de las altas frecuencias. Su efectividad depende principalmente de la proximidad del pin a la alimentación individual. Por consiguiente, deben estar localizados cerca de los pines de la alimentación.

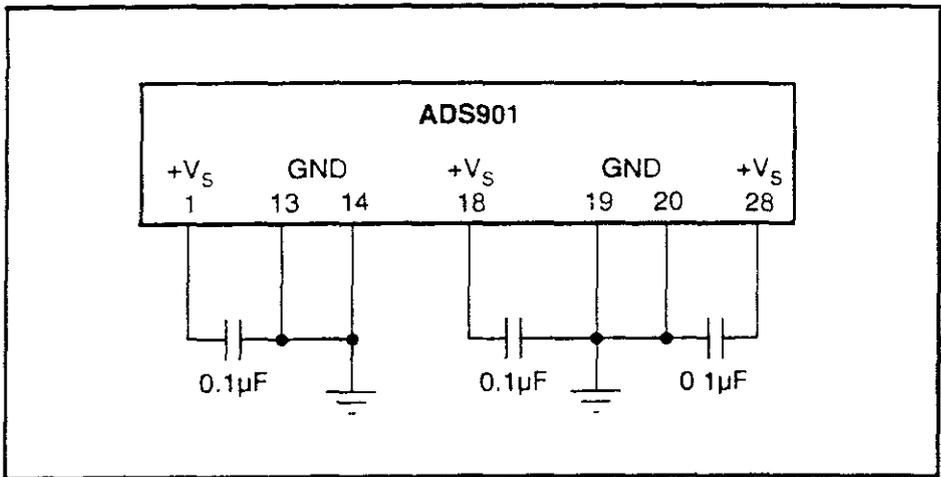


Figura III.4- Desacoplamiento de tierra:

Una vez familiarizados con el convertidor y habiendo observado sus características electricas podemos determinar que, para manejar los niveles de referencia de acuerdo a las recomendaciones del fabricante, debemos tener cuidado en el voltaje que lo esta polarizando. El arreglo de resistencias internas del convertidor y el voltaje de polarización no dan datos suficientes para calcular las resistencias de referencia las referencias externas. El fabricante nos recomienda una solución de bajo costo para tener las referencias del voltaje de entrada. Como se efectuó la prueba físicamente, optamos por esta para el diseño final. Tomamos en cuenta que se lleva a máxima velocidad del reloj del convertidor al colocarse un oscilador, el cual proporciona una señal de reloj con una frecuencia de 20 MHz. Afortunadamente, el ADS901E lo soporta sin dificultades. Ahora bien, la etapa que consideramos problematica en el convertidor es la del valor de los voltajes de referencia, ya que esos voltajes deben adecuarse a la amplitud de la señal de la cámara, que nos

proporciona un voltaje de 1 V Vpp. Debemos tomar en cuenta este valor junto con los datos anteriores para realizar el cálculo del divisor de voltaje. Éste se forma conectando las resistencias externas con las internas del convertidor para dar la referencia de la amplitud de la señal a digitalizar.

47

En la figura III.48 podemos observar las conexiones tanto internas como externas para definir los voltajes de referencia. El cálculo de estos valores parte de una herramienta básica que es el **DIVISOR DE VOLTAJE**. Primero se realizara el análisis para dos elementos y este resultado se utilizara finalmente para los valores de los elementos que necesitamos.

### III.2.K. DIVISOR DE VOLTAJE.

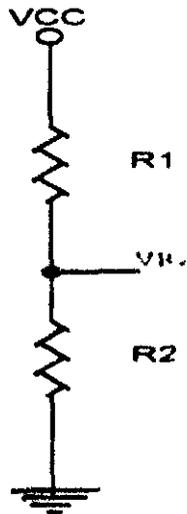


Figura III.50 Divisor de voltaje general.

La ecuación del divisor de voltaje es la siguiente:

$$V_{R_2} = \frac{R_2}{R_1 + R_2} V \quad \text{EC. III.5}$$

Si deseo encontrar los valores de  $R_2$  o  $R_1$ , entonces:

$$V_{R_2} R_1 = R_2 (V_{in} - V_{R_2}) \quad \text{EC. III.6}$$

De aquí deducimos que los valores de  $R_1$  y  $R_2$  son:

$$R_2 = \frac{V_{R_2} R_1}{V_{in} - V_{R_2}} \quad \text{EC. III.7}$$
$$R_1 = \frac{R_2 (V_{in} - V_{R_2})}{V_{R_2}}$$

Debemos observar que la mayoría de los elementos utilizados en la tarjeta SAD son elementos del tipo TTL y se polarizan con un voltaje mínimo de 4.5 Volts. El fabricante del ADS901E señala que el voltaje de polarización adecuado es de 3 Volts, pero observa que los valores máximos permitidos son aproximadamente 6 Volts. La polarización se presenta a diferentes voltajes, desde 2.5 Volts hasta 7 Volts, aunque existe el riesgo de dañar los componentes.

La tarjeta se comportó muy estable del rango de 3.7 Volts a 5.5, pero en un voltaje de 4.5 Volts presentó características adecuadas para polarizar finalmente a la tarjeta. Algunas de las características que se presentaron en este nivel de alimentación de la tarjeta fueron: baja disminución de ruido, bajo consumo de potencia y buena digitalización de la señal.

Los valores de referencia deseados se pueden calcular partiendo de este voltaje de polarización, las características de la señal de video y el análisis del divisor de voltaje:

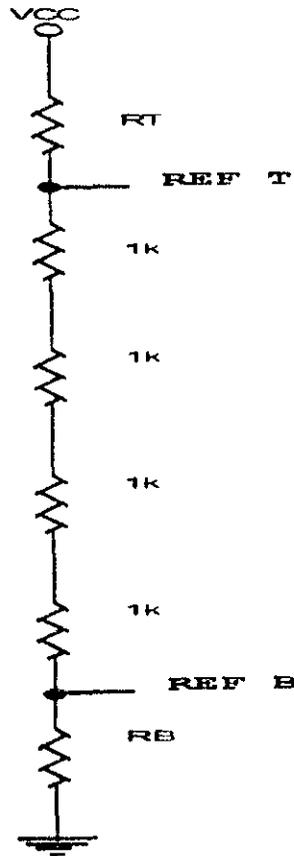


Figura III.51. Divisor de Voltaje para la referencia del convertidor.

Se calcula el valor de  $R_B$  tomando hasta  $REF\ T$  como si fuera  $R_2$  y  $R_B$  como  $R_1$ . El voltaje de  $REF\ T$  es de 2 Volts. Siguiendo con la similitud,  $V_{CC}=REF\ T$ ,  $V_{R_2}=REF\ B$ . Entonces:

Si REF T = 2 Volts, REF B = 1 Volt

Y la suma de las resistencias es igual a  $4 \text{ k}\Omega \Rightarrow$

$$R2 = \frac{4\text{k}\Omega(1-2)}{(2-1)} \quad \text{Eq. III.1}$$

$$R2 = 4\text{k}\Omega$$

Por tanto, el valor de RB =  $4 \text{ k}\Omega$

Ahora, con el dato ya obtenido, se realiza nuevamente el cálculo de Voltaje, tomando como una sola resistencia desde RB hasta el valor de REF T.

$$R1 = \frac{8\text{k}\Omega(5-2)}{(2)} \quad \text{Eq. III.2}$$

$$R1 = 10\text{k}\Omega$$

El valor de RT =  $10 \text{ k}\Omega$

Con estos valores, podemos asegurar una referencia precisa de convertidor para manejar apropiadamente la señal de voltaje.

Es importante verificar que se sigan las recomendaciones del fabricante al conectar el ADC0811 con el fin de aprovechar al máximo las características y cualidades que son indispensables para el dispositivo electrónico. Por ejemplo:

- Aislamiento de la fuente de polarización.
- Consideraciones para obtener una buena referencia a tierra.
- Solución de referencia a bajo costo (recomendación del fabricante).

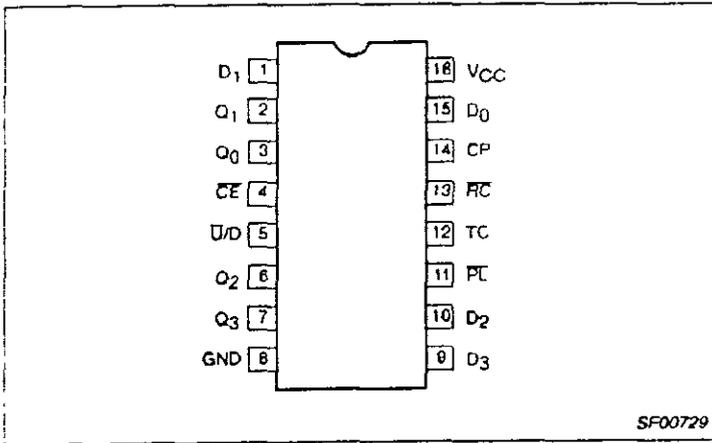
### III.3. CONTADORES

Un contador es, esencialmente, un registro que pasa por una secuencia predeterminada de estados después de la aplicación de pulsos de entrada. Las compuertas en un contador se conectan de manera que se produce una secuencia preestablecida de estados binarios en el registro. Aunque los contadores son un tipo especial de registro, es común diferenciarlos dándoles un nombre diferente.

#### III.3.A. OPERACIÓN DE CONTADORES ASINCRONOS.

El término asincrónico o de rizado de reloj, se refiere a que la transición de salida del flip flop sirve como fuente para disparar los demás flip flops. En otras palabras, las salidas  $Q_i$  de todos los flip flops (con excepción de la primera) se disparan no por los pulsos de la entrada sino por la transición que ocurre en los otros flip flops. El cambio de estado de un flip flop en particular es dependiente del estado presente de otros flip flops.

El contador 74F191 es un contador binario que contiene 4 flip flops Maestro/esclavo disparados por borde para realizar operaciones del tipo de retención y salida asincrónica y cuenta arriba/abajo sincrónico.



SF00729

Figura III.52 Configuración de los pines.

PINS	DESCRIPTION	74F(U.L.) HIGH/LOW	LOAD VALUE HIGH/LOW
D <sub>0</sub> - D <sub>3</sub>	Data inputs	1.0/1.0	20µA/0.6mA
CE	Count enable input (active Low)	1.0/3.0	20µA/1.8mA
CP	Clock pulse input (active rising edge)	1.0/1.0	20µA/0.6mA
PE	Asynchronous parallel load control input (active Low)	1.0/1.0	20µA/0.6mA
U/D	Up/down count control input	1.0/1.0	20µA/0.6mA
Q <sub>0</sub> - Q <sub>3</sub>	Flip-flop outputs	50/33	1.0mA/20mA
RC	Ripple clock output (active low)	50/33	1.0mA/20mA
TC	Terminal count output	50/33	1.0mA/20mA

Tabla III.23 Tabla de configuración de los pines.

INPUTS					OUTPUTS	OPERATING MODE
PE	U/D	CE	CP	D <sub>n</sub>	Q <sub>n</sub>	
L	X	X	X	L	L	Parallel load
L	X	X	X	H	H	
H	L	↓	↑	X	Count up	Count up
H	H	↓	↑	X	Count down	Count down
H	X	H	X	X	No change	Hold (do nothing)

Tabla III.24 Modo de selección.

INPUTS			TERMINAL COUNT STATE				OUTPUTS	
U/D	CE	CP	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	TC	RC
H	H	X	H	H	H	H	L	H
L	H	X	H	H	H	H	H	H
L	L	↓	H	H	H	H	H	
L	H	X	L	L	L	L	L	H
H	H	X	L	L	L	L	H	H
H	L	↓	L	L	L	L	H	

Tabla III.25 Funciones TC y RC.

H = Alto Voltaje

L = Bajo Voltaje

X = No importa

↓ = Pulso bajo

↑ = Transición de Reloj de bajo a alto

↓ = Nivel de voltaje bajo en un solo tiempo dando prioridad a la transición de Reloj de bajo a alto.

Como la transferencia de los datos a la memoria debe darse en el menor tiempo posible, se utilizo la configuracion de contador de N-estados utilizando el rizo de reloj o asincrono, que es una de las aplicaciones que nos proporciona el fabricante.

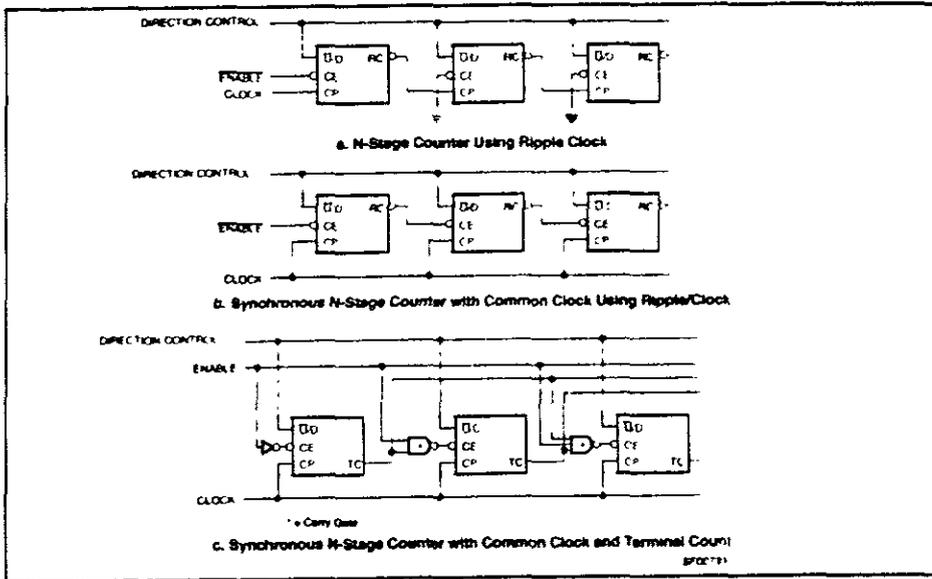


Figura III.53 Aplicaciones típicas.

El 74F19i simplifica el diseño de los contadores de multiples estados como lo indica la Figura III.53. La señal de RC es utilizada en cada salida para que el siguiente estado encuentre un nivel alto. Cuando la fuente que proporciona la señal de reloj es limitada en el manejo de la compatibilidad, esta configuracion es particularmente ventajosa, ya que la fuente de la señal de reloj solo maneja el primer estado. Unicamente se necesita esto para inhabilitar el primer estado y prevenir que se inicie un conteo en todos los estados siguientes. Estando en un nivel alto

CE se inhabilita la salida de pulsos de RC', como lo indica la tabla de selección de modos. El cambio en el pulso de tiempo entre el estado inicial se representa con un retraso acumulado, como si fuera un rizo de reloj a través de los estados que lo preceden. La carga paralela asincrónica permite dar un número cualquiera preestablecido en las entradas. La información presente en los datos de entrada paralela (D) es cargada dentro del contador y aparece en las salidas cuando la entrada de carga paralela se encuentra en estado bajo (PL). Esta operación anula la función de cuenta. La operación de cuenta también es desactivada por un nivel alto en la entrada de habilitación del contador (CE). Cuando está en nivel bajo, los estados cambian al estado de inicio. Las indicaciones de error (errores) típicas son proporcionadas por dos tipos de salidas, la terminal de cuenta (TC) y el Rizo de Reloj. Normalmente la salida de TC es baja y va a alta en las siguientes condiciones:

- 1) Cuando alcanza un cero en el modo de contador descendente;
- 2) Cuando llega a "15" en el modo de contador ascendente. La salida de TC permanecerá en alto hasta que ocurra un cambio en cualquiera de los dos modos del contador, de algún número establecido o hasta que sea cambiado.

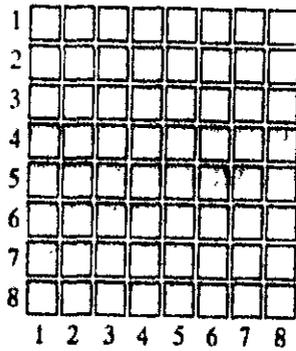
La salida de TC no debe utilizarse como salida de reloj porque está sujeta a los cambios de estado de los flip flops. La señal de TC es utilizada internamente para habilitar la salida de RC'. Cuando TC está en alto y CE está en bajo, el RC' sigue el pulso de reloj. La salida de RC' duplica el ancho del pulso de reloj en su nivel bajo por el retraso en el tiempo debido a los dos atrasos de las compuertas lógicas.

### III.4. MEMORIAS

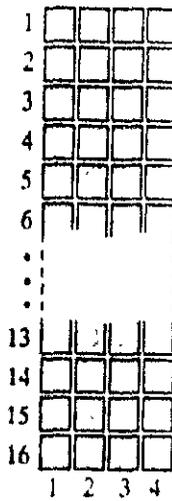
El dispositivo que nos permite guardar la información dentro de la tarjeta SAE es la memoria RAM que existe en ella. Pero hablemos acerca de este elemento electrónico. La memoria es un dispositivo electrónico que nos sirve para almacenar datos binarios sobre una base de larga o corta duración. Los dos tipos básicos de memoria son los semiconductores y los magnéticos. Las memorias semiconductoras consisten en arreglos de elementos de almacenaje, que son flip flops o capacitores.

#### III.4.A. ARREGLO BÁSICO DE MEMORIA SEMICONDUCTORA.

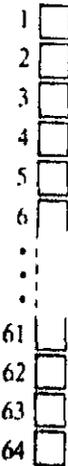
Cada elemento de almacenamiento en una memoria puede tener un "1" o un "0" y se llama celda. Las memorias se forman de arreglos de celdas, como se ilustra en la figura III.54, utilizándose 64 celdas como un ejemplo. Cada bloque en el arreglo de memoria representa una celda de almacenamiento y su ubicación puede identificarse especificando una fila y una columna.



(a) Arreglo 8 x 8.



(b) Arreglo 16 x 4



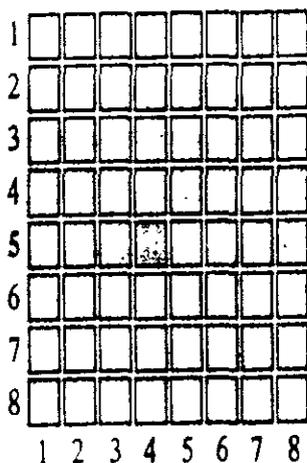
(c) Arreglo 64 x 1

Figura III.54 Arreglo de almacenamiento de 64 celdas organizado en tres formas diferentes.

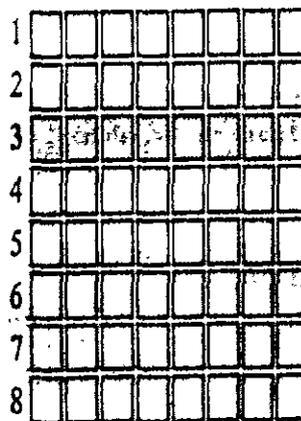
El arreglo de 64 celdas puede organizarse en varias formas basadas en unidades de datos. La Figura III.54 (a) muestra un arreglo de 8x8, que es una memoria de 64 bits o una de 8 bytes. La Figura III.54 (b) muestra un arreglo de 16x4 que es una memoria de 16 nibbles y la (c) ilustra un arreglo de 64x1, que es una memoria de 64 bits. Una memoria se identifica por el número de palabras que puede almacenar multiplicado por el tamaño de la palabra. Por ejemplo, una memoria de 16Kx4 puede almacenar 16 384 palabras de 4 bits cada una.

### III.4.B. DIRECCIÓN Y CAPACIDAD DE MEMORIA.

La capacidad de una unidad de datos en un arreglo de memoria se denomina dirección. Por ejemplo, en la Figura III.55 (a) la dirección de un bit en el arreglo se especifica por fila y columna. Como se muestra en la Figura III.55 (b), la dirección de un byte se especifica sólo por la fila. Como se puede observar, la dirección depende de cómo se organice la memoria en unidades de datos.



(a) La dirección del bit es fila 5, columna 4.

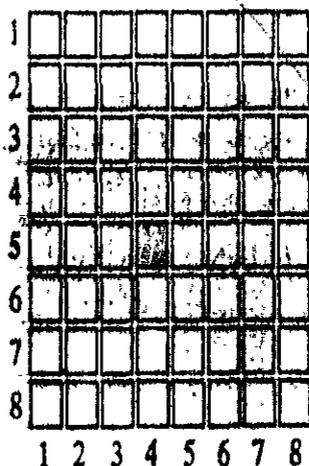


(b) La dirección del byte es fila 3.

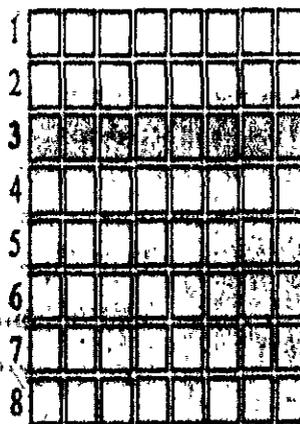
Figura III.55 Dirección de Memoria.

### III.4.B. DIRECCIÓN Y CAPACIDAD DE MEMORIA.

La capacidad de una unidad de datos en un arreglo de memoria se denomina dirección. Por ejemplo, en la Figura III.55 (a) la dirección de un bit en el arreglo se especifica por fila y columna. Como se muestra en la Figura III.55 (b), la dirección de un byte se especifica sólo por la fila. Como se puede observar, la dirección depende de cómo se organice la memoria en unidades de datos.



(a) La dirección del bit es fila 5, columna 4.



(b) La dirección del byte es fila 3.

Figura III.55 Dirección de Memoria.

FALTA PAGINA

No. 190

### III.4.D. OPERACIÓN ESCRITURA.

La operación de escritura básica se ilustra en la Figura III.57. Para almacenar un byte de datos en la memoria, se coloca un código sostenido en el registro de dirección en el bus de direcciones. Una vez que el código de dirección está en el bus, el decodificador de dirección proporciona la dirección y selecciona la localidad especificada en la memoria. La memoria adquiere enseguida una orden de escribir y el byte de datos mantenido en el registro se coloca en el bus de datos y se almacena en la dirección de memoria seleccionada, completando así la operación de escribir. Cuando se escribe un byte de datos en una dirección de memoria, el byte de datos almacenado en esa dirección se pierde.

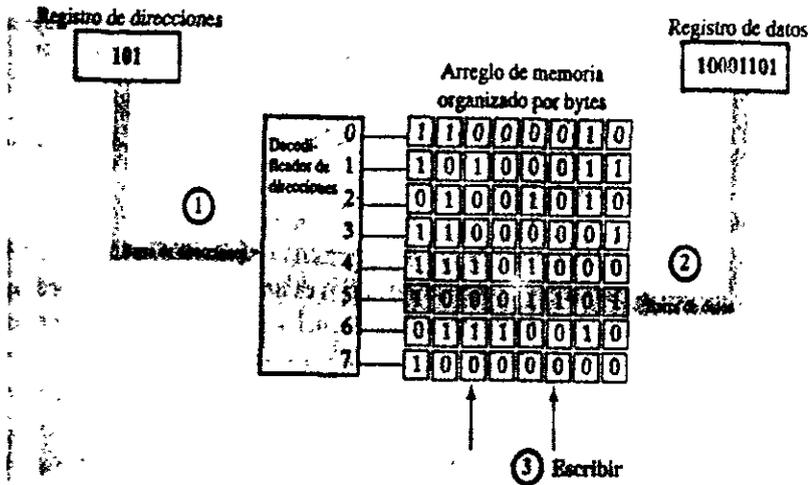


Figura III.57 Ilustración de la operación de escritura.

### III.4.E. OPERACIÓN DE LECTURA.

En la Figura III.58 se ilustra la operación de lectura básica. Nuevamente, en la barra se coloca un código mantenido en el registro de dirección. El decodificador de direcciones decodifica la dirección y selecciona la localidad especificada en la memoria. La memoria adquiere luego una orden de leer y se coloca en el bus de datos una copia del byte de datos, que está almacenado en la dirección de memoria seleccionada. El byte se carga temporalmente en el registro de datos, completando así la operación de lectura. Cuando se lee un byte de datos de una dirección de memoria, permanece almacenado en esa dirección y no se destruye.

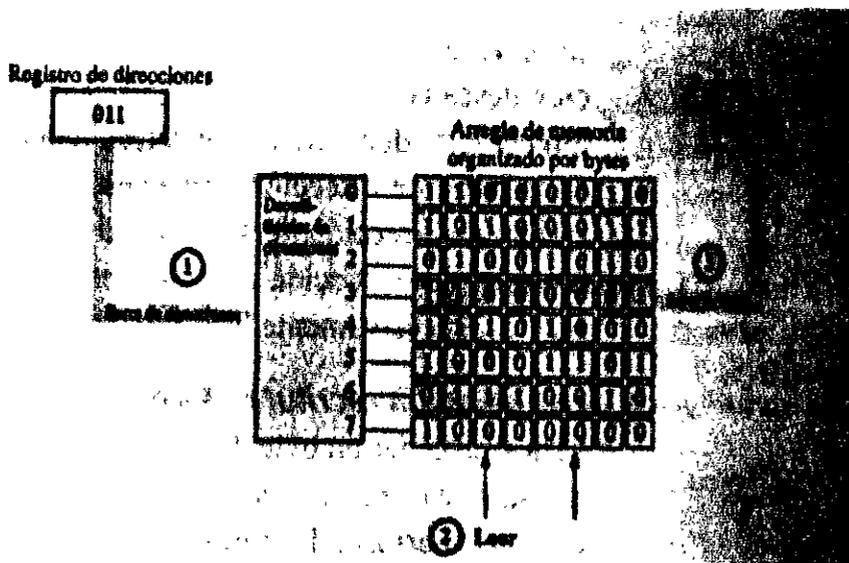


Figura III.58 Ilustración de la operación de lectura.

### III.4.F. RAM.

La RAM (memoria de acceso aleatorio) es un tipo de memoria en donde se puede tener acceso a todas las direcciones en un tiempo igual. Se pueden seleccionar en cualquier orden para una operación de lectura o escritura. Todas las RAM tienen la capacidad de leer y escribir. La RAM son memorias volátiles porque pierden datos almacenados cuando se interrumpe la energía eléctrica. En el diseño de la tarjeta SAD, es importante almacenar los datos digitales de la conversión analógica, pero que se debe pensar en un diseño que nos permita almacenar dichos primeros y, posteriormente, tener acceso a ellos leerlos.

Para el diseño del almacenamiento de datos dentro de la tarjeta SAD se selecciona una RAM estática, ya que sus características eléctricas son las más adecuadas para el tipo de datos digitales que se van a obtener de la tarjeta. Pero ¿cuáles son las características de la SRAM? Una vez que se almacena un bit de datos en una celda, este permanece ahí indefinidamente o al menos que se interrumpa la energía eléctrica o se escriba el nuevo bit de datos. Lo que interesa, es que la información este presente en el momento que la cámara esta "viendo" al objeto. Es por este motivo que elegimos la memoria SRAM. No importa retener los datos en la tarjeta, ya que se envía de forma serial la información obtenida a una PC, desde ahí, podemos interpretar y manipularlos más fácilmente que en una RAM dinámica. Anteriormente mencionamos dos dispositivos electrónicos que son importantes en la retención y salida de los datos digitalizados, por lo que, se encuentran relacionados. No se debe olvidar que el microcontrolador da las líneas de control a la memoria para que realice el ciclo de lectura/escritura. Para seleccionar alguno de

estas opciones, se observan las características eléctricas y el diagrama de tiempos de la memoria, ya que el fabricante (en este caso, HITACHI) nos proporciona todos los datos necesarios para elegir el ciclo de escritura/lectura y poder manipularlo de acuerdo a las necesidades del sistema.

MODO	CS	WE	OE	V <sub>CC</sub>	OPCION
SIN ENERGIA	ALTO	N. IMPORTA	N. IMPORTA	ALTA	
LECTURA	BAJO	ALTO	BAJO	Baja	LECTURA
ESCRITURA	BAJO	BAJO	ALTO	Baja	ESCRITURA
	BAJO	BAJO	BAJO	Baja	ESCRITURA

Tabla III.26 Modos de habilitación de la memoria.

## ■ PIN ARRANGEMENT

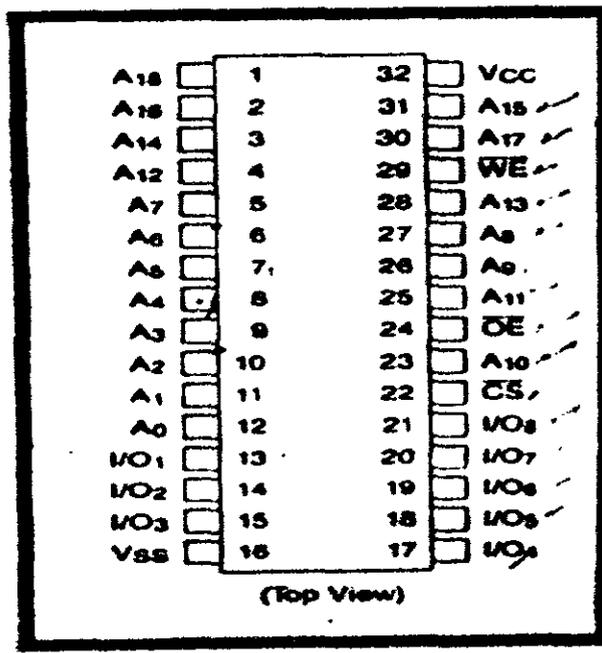


Figura III.59 Configuración de los pines de la Memoria HM66205

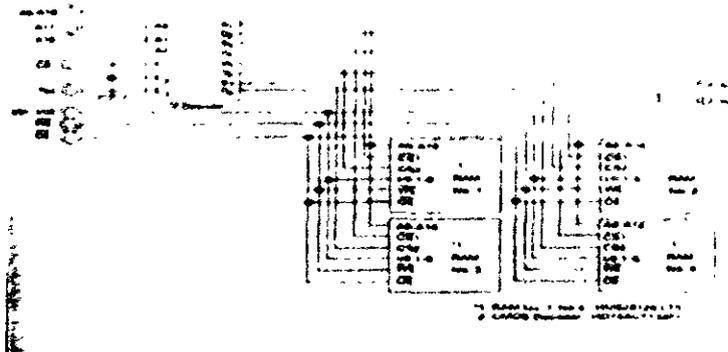


Figura III.60 Diagrama a bloques de la memoria.

• Write Cycle Timing (1) (OE Clock)

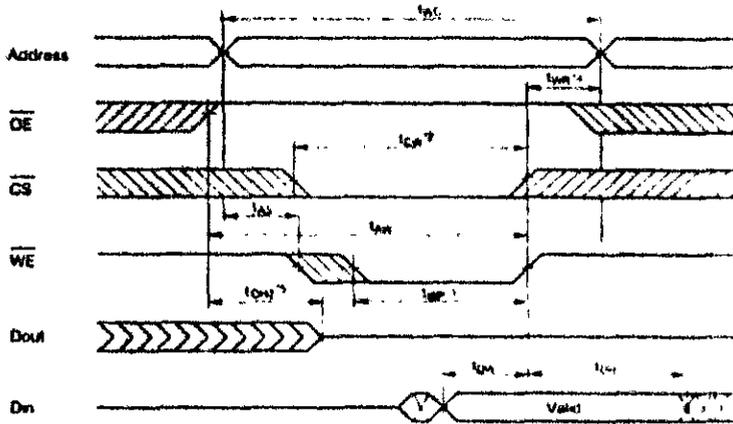
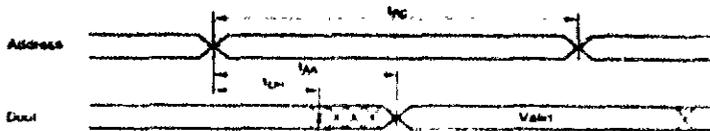
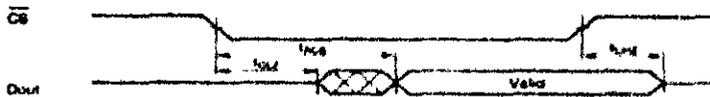


Figura III.61 Ciclo de Escritura.

• Read Cycle Timing (2) \*L \*A \*E



• Read Cycle Timing (3) \*L \*S \*E



- NOTES
- 1 OE is high for read cycle
  - 2 Dn is continuously updated. CS =  $t_{DQ}$
  - 3 Address should be valid prior or coincident with CS transition time
  - 4 DS =  $t_{DQ}$



Figura III.62 Ciclo de lectura.

Ya que se conoce de que forma se habilitan o inhiben los ciclos de lectura/escritura dentro de la memoria, esta información se proporciona al programa de microcontrolador.

Necesitamos ahora dar la dirección a los datos de la memoria. Debido a que la transferencia de datos de entrada/salida se tiene que realizar en el menor tiempo posible, se utiliza un arreglo de contadores en cascada para dar la dirección a la memoria, activar el arreglo de contadores y lograr que se encuentren sincronizados con la memoria y los demás elementos que se encuentran involucrados con ellos en la tarjeta SAD.

De acuerdo con lo anterior, podemos resumir los estados de nivel de voltaje alto o bajo que se necesitan para activar/desactivar las funciones de la memoria y del contador, así como su correspondiente salida del Puerto. En este caso, el Puerto C es el que se configura como Puerto de salida.

PUERTO C		NIVEL BAJO "0"	NIVEL ALTO "1"
	LOGIC DE LOS CONTADORES	ACTIVA	DESACTIVA
	HABILITACION DE LA CUENTA DE LOS CONTADORES	ACTIVA	DESACTIVA
C0	HABILITA LA MEMORIA	ESCRITURA	LECTURA
C1	SEÑAL DE RELOJ	NO IMPORTA	NO IMPORTA
C2	HABILITA LA SEÑAL DE RELOJ	LENT	RAPID
C3	HABILITA AL CONVERTIDOR	ACTIVA	DESACTIVA

Tabla III.27 Asignaciones al puerto del microprocesador.

### III.5. COMPARADOR.

El más común para abarcar la señal de voltaje en una entrada con un voltaje de referencia en la entrada de un amplificador operacional de propósito general, tiene la desventaja de que no posee suficiente rapidez en su respuesta. Sus cambios de salida se encuentran entre los límites fijados por los de saturación,  $+V_{sat}$  y  $-V_{sat}$ . Por lo tanto, su salida no puede activar dispositivos como el de línea finita TTL, que requiere niveles de voltaje de 0 y 5 V.

Estas desventajas se eliminan mediante el uso de un integrado que haya sido diseñado específicamente para actuar como un comparador (Amplificador Operacional de Colector Abierto "Open Collector").

El amplificador operacional de propósito general y el comparador no pueden operar correctamente si existe ruido en cualquier entrada. Con el fin de resolver este problema se utilizan técnicas para eliminarlo, por ejemplo, los comparadores con histéresis.

### III.5.A. COMPARADOR CON HISTÉRESIS.

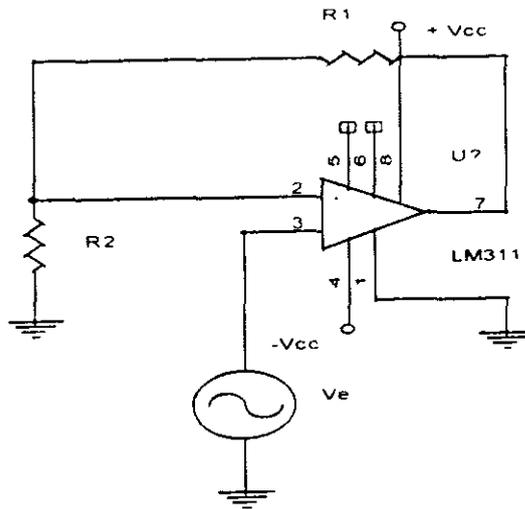


Figura III.63 Esquemático del circuito comparador con histeresis que utiliza el LM311.

Al analizar el circuito tenemos:

$$V_e = I_n \sin(\omega t) \dots \dots \dots \text{EC. III.10}$$

Si observamos el divisor de voltaje formado por el arreglo de R1 y R2, el voltaje de R2 =  $V_p$  del amplificador operacional, es:

$$V_p = \frac{R_2}{R_1 + R_2} I_s \dots \dots \dots \text{EC. III.11}$$

que sucede cuando:

$$V_S \begin{cases} +V \\ -V \end{cases}$$

EC. III.13

entonces se obtiene:

$$V_P \begin{cases} \frac{R_2}{R_1 + R_2} (+V) \\ \frac{R_2}{R_1 + R_2} (-V) \end{cases} \quad \text{EC. III.14}$$

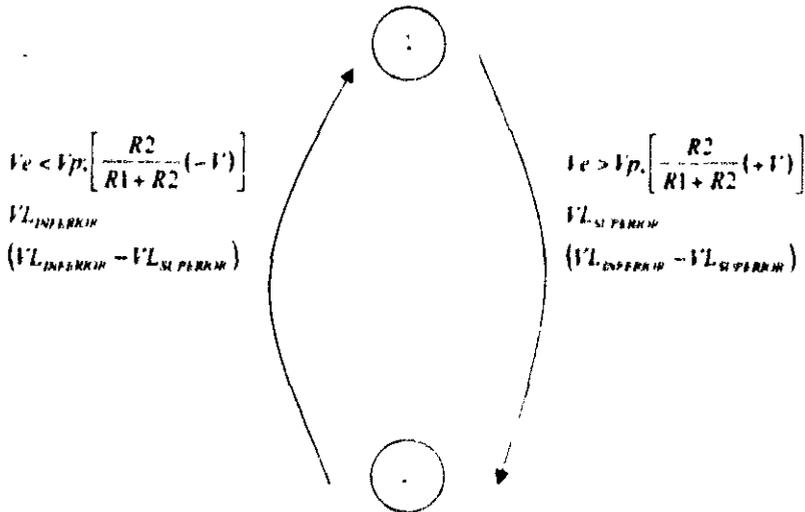


Figura III.64 Estados de histeresis para el comparador.

Por lo que el ciclo de histeresis para este tipo de análisis es:

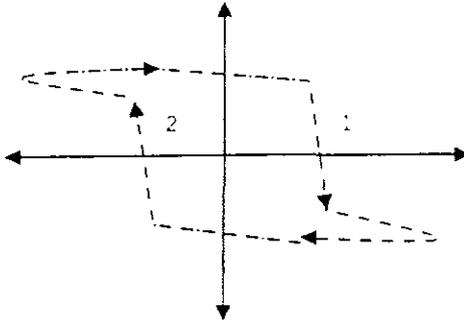


Figura III.65 Ciclo de histéresis.

$$V_H = VL_{SUPERIOR} - VL_{INFERIOR}$$

$$V_H = \frac{R2}{R1+R2} (+V) - \frac{R2}{R1+R2} (-V)$$

EC. III.14

$$V_H = \frac{2R2}{R1+R2} (V)$$

$$V_H = \frac{VL_{SUPERIOR} + VL_{INFERIOR}}{2} = 0$$

El comparador 111 o 311 es un CI que ha sido diseñado y mejorado para un rendimiento superior en las aplicaciones de detector de nivel de bajo voltaje. Un comparador debe ser de rápida respuesta a los cambios de entrada. El LM 311 es mucho más veloz que el 741 o 301. Por las características de la señal de control que maneja, este tipo de comparador cumple con los requisitos de costo y velocidad.

El comparador 311 es una apropiada elección por su versatilidad. Su salida está diseñada para no variar entre

IV sat, pero puede manipularse con gran facilidad. Al observar la salida del 311, se comporta como un interruptor conectado entre la terminal de salida y la terminal de entrada positiva. La terminal puede conectarse a cualquier voltaje de +Vref hasta de 40V mas positiva que la terminal de alimentacion -Vcc (terminal 4). Cuando la terminal de entrada positiva (terminal 2) es mas positiva que la entrada de terminal negativa (terminal 3), el interruptor de salida equivalente del 311 esta abierto.  $V_o$  se determina por el +Vref, que esta conectado a la terminal de salida.

Cuando la terminal de entrada positiva es menos positiva que la terminal de entrada negativa, el interruptor de salida equivalente del 311 se cierra y extiende la malla de tierra de la terminal de entrada positiva hasta la terminal de salida. Las resistencias de  $R_f$  y  $R_i$  adicionan cerca de 50 mV de histéresis para minimizar los efectos de ruido, de modo que la terminal de entrada positiva esta prácticamente en 0V. Las formas de onda para  $V_o$  y el  $V_{in}$  se muestran en la figura III.6. El  $V_o$  esta a 0V (interruptor cerrado) para los medios ciclos positivos de  $V_{in}$ .  $V_o$  esta al voltaje de  $V_c$  (interruptor abierto) para los medios ciclos negativos de  $V_{in}$ . Este es un circuito tipico de interfaz; esto es, los voltajes pueden variar entre los niveles de +Vcc y -Vcc, pero  $V_o$  esta restringido entre el voltaje de referencia y 0 volts. El voltaje de referencia (Vref) se puede fijar a 5 Volts y entonces se tendrían los niveles típicos de señal analógica a digital.

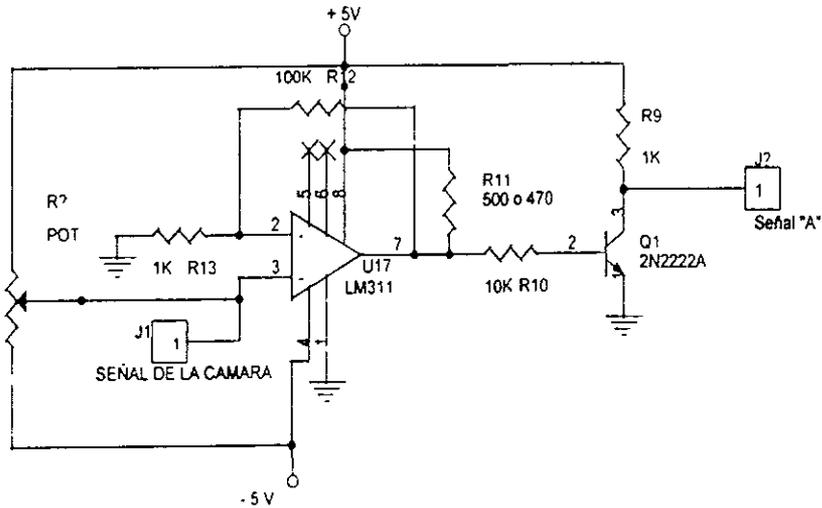


Figura III.66 Circuito comparador en el sistema de adquisición de datos.

En la explicación anterior se analizó el circuito comparador con histeresis. Ahora se obtendrán los valores de los elementos que forman la malla de retroalimentación del circuito.

El potenciómetro del circuito comparador se utiliza para darle un offset a la señal de video de la cámara. Esto es recomendable para obtener una referencia adecuada de la señal. En nuestro caso, se usa para quitar el pulso de sincronía de la señal de la cámara. Para tener un buen control, típicamente se recomienda un valor de entre 10 K $\Omega$  y 100 K $\Omega$ .

El comparador de potencial comparará el nivel de la señal de sincronía de video con el nivel de referencia.

El nivel de referencia se genera de la siguiente manera:

Entonces:

La salida del comparador es conectada al emisor del transistor 74LS120.

El nivel de referencia se genera de la siguiente manera:

$$V_e = I_{\text{suplementaria}} \cdot \frac{R_1 \cdot R_2}{R_1 + R_2} (+I_{\text{suplementaria}}),$$

donde

$$R_2 = \frac{R_1(I_{\text{suplementaria}})}{(+I_{\text{suplementaria}} - I_{\text{suplementaria}})} \quad (1)$$

sustituyendo

$$R_2 = \frac{50 \times 10^{-3} (100 \times 10^{-3})}{(5 - 50 \times 10^{-3})} = \frac{5000}{4,95} = 1010 \Omega \approx 1 \text{ k}\Omega$$

Entonces, que el valor de  $R_2$  es  $R_2 = 1 \text{ k}\Omega$ .

En la salida del comparador se encuentra conectado el transistor, la función del transistor es la de servir como interruptor que invierta la señal, alterando el nivel de la señal a nivel TTL, puesto que calculamos la salida con un nivel de 5V de DTL. El circuito comparador tiene la finalidad de quitar el pulso de sincronía de la señal de video e inhibir la alta impedancia del 74LS120, para dar paso a la señal de reloj conectada al convertidor. De otra manera, es lo

forma que se le indica al convertidor ADS901E cuando va a comenzar la digitalización.

Este tipo de arreglo permite que, cuando este el pulso de sincronía de la señal de video, el convertidor no detecte ningún tipo de señal de reloj. Entonces, el convertidor no opera y da un nivel de voltaje constante. Así, el software detecta un tren de datos constantes (se explicara esta característica en el desarrollo del software).

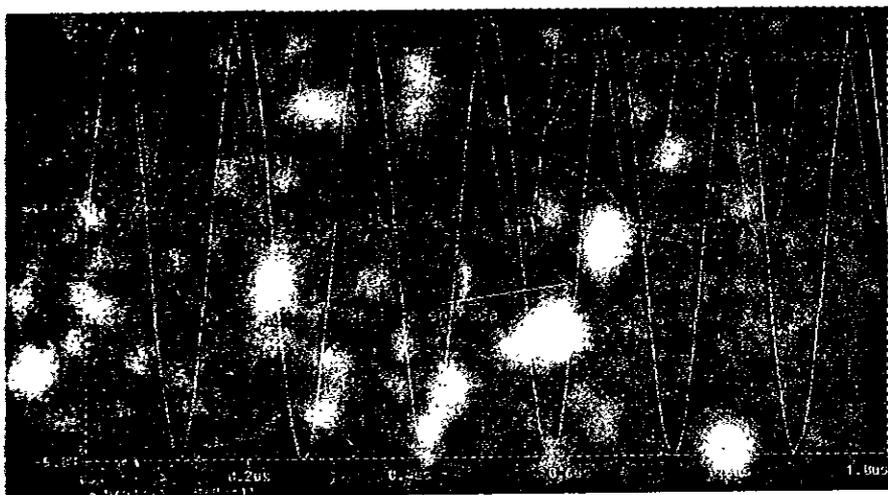


Figura III.67 Salida del comparador.

### III.6. REGIONES DE CORTE Y SATURACIÓN DEL TRANSISTOR.

Como sólo necesitamos que el transistor funcione como interruptor y éste invierta la señal que recibe del circuito comparador, solamente se hará el análisis en las regiones de corte y saturación del transistor.

Recordemos las regiones del transistor:

$V_{BE}$	$V_{BC}$	REGIÓN
<0	<0	Corte $I_B = I_C = I_E = 0$ $V_{CE} = +V_{CC}$
>0	<0	Activa directa $I_C = \beta I_B$ ; $I_E = I_C$ ; $I_C = \beta I_E$ $V_{BE} = 0.7V$
<0	>0	Activa inversa $I_E = I_C / \beta$
>0	>0	Saturación $I_{C(SAT)} = \beta I_{B(SAT)}$

Tabla III.28 Regiones de corte y saturación para el transistor.

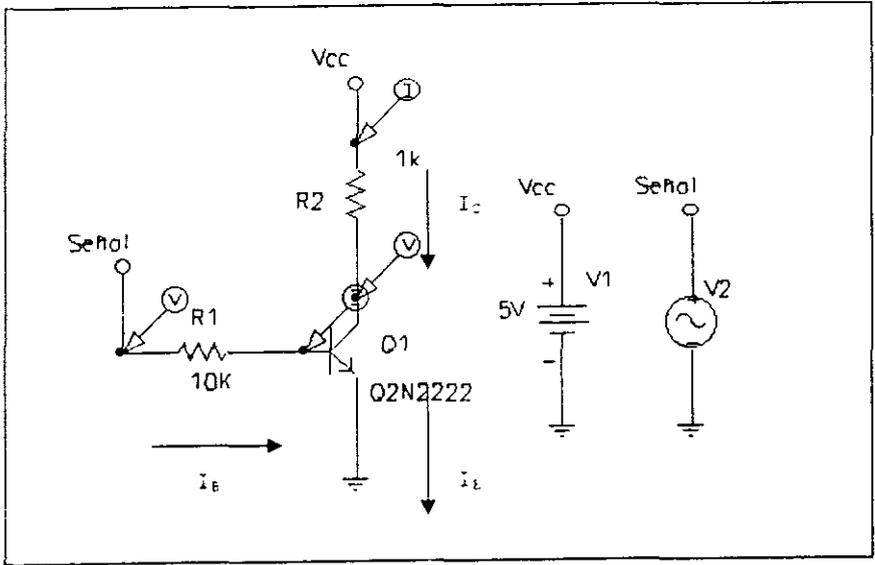


Figura III.68 Polarización del transistor en la configuración de emisor común.

Analizando las mallas del circuito obtenemos:

$$V_{cc} = I_c R_c + V_{cE}$$

$$V_{BES} = R_B I_B + V_{BE} \dots \dots \dots \text{Ec. III.18}$$

$$I_c = \beta I_B \cdot I_B = \frac{I_c}{\beta}$$

De los datos del fabricante:

**PN2222A**

**ELECTRICAL CHARACTERISTICS** ( $I_A = 20$  mA unless otherwise noted) (Continued)

Characteristic	Symbol	Min	Max	Unit	
<b>ON CHARACTERISTICS</b>					
DC Current Gain ( $I_C = 0$ mA; $V_{CE} = 10$ Vdc) ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc) ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc) ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $I_A = -50$ mA) ( $I_C = 150$ mA; $V_{CE} = 10$ Vdc) <sup>1</sup> ( $I_C = 150$ mA; $V_{CE} = 10$ Vdc) <sup>2</sup> ( $I_C = 500$ mA; $V_{CE} = 10$ Vdc) <sup>2</sup>	$\beta_{DC}$	30 50 100 100 100 100			
Collector-Emitter Saturation Voltage <sup>1</sup> ( $I_C = 150$ mA; $I_B = 15$ mA) ( $I_C = 500$ mA; $I_B = 50$ mA)	$V_{CE(sat)}$		0.5 1	Vdc	
Base-Emitter Saturation Voltage <sup>1</sup> ( $I_C = 150$ mA; $I_B = 15$ mA) ( $I_C = 500$ mA; $I_B = 50$ mA)	$V_{BE(sat)}$	0.6	1.2 2	Vdc	
<b>SMALL-SIGNAL CHARACTERISTICS</b>					
Current Gain Bandwidth Product <sup>2</sup> ( $I_C = 20$ mA; $V_{CE} = 20$ Vdc; $f = 100$ MHz)	$f_T$	300		MHz	
Output Capacitance ( $V_{CE} = 10$ Vdc; $f = 1$ to $10$ MHz)	$C_{out}$		8	pF	
Input Capacitance ( $V_{BE} = 0.5$ Vdc; $f = 0.1$ to $10$ MHz)	$C_{in}$		25	pF	
Input Impedance ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz) ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz)	$Z_{in}$	200 0.25	60 1.0	k $\Omega$ $\Omega$	
Voltage Feedback Ratio ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz) ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz)	$V_{fb}$		8 40	dB dB	
Small-Signal Current Gain ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz) ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz)	$\beta_{small}$	50 20	300 200		
Output Admittance ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz) ( $I_C = 10$ mA; $V_{CE} = 10$ Vdc; $f = 10$ MHz)	$Y_{out}$	5.0 20	30 20	$\mu$ S $\mu$ S	
Collector Base Time Constant ( $I_C = 20$ mA; $V_{CE} = 20$ Vdc; $f = 10$ MHz)	$\tau_{cb}$		1	ns	
Flare Figure ( $I_C = 100$ mA; $V_{CE} = 10$ Vdc; $f_c = 10$ kHz; $f = 10$ MHz)	$F$		6	dB	
<b>SWITCHING CHARACTERISTICS</b>					
Delay Time ( $V_{CC} = 5$ Vdc; $V_{BE(on)} = 2.0$ Vdc; $I_C = 150$ mA; $I_B = 15$ mA) (Figure 1)	$t_d$		1	ns	
Rise Time ( $V_{CC} = 5$ Vdc; $I_C = 150$ mA; $I_B = 15$ mA) (Figure 1)	$t_r$		20	ns	
Storage Time ( $V_{CC} = 5$ Vdc; $I_C = 150$ mA; $I_B = I_{CP} = 15$ mA) (Figure 2)	$t_s$		200	ns	
Fall Time ( $V_{CC} = 5$ Vdc; $I_C = 150$ mA; $I_B = 15$ mA) (Figure 2)	$t_f$		50	ns	

1 Pulse Test Pulse Width = 300 ns; Duty Cycle = 20%  
2  $f_T$  is defined as the frequency at which  $\beta_{DC}$  extrapolates to unity.

Figura III.69 Características electricas del transistor.

De donde obtenemos tanto los voltajes de saturación, como las corrientes de saturación.

De acuerdo a las ecuaciones anteriores, podemos calcular los valores de las resistencias del circuito del transistor.

Donde:

$$R_B = \frac{V_{BB} - V_{BE}}{I_B}$$

si

$$I_B = \frac{I_C}{\beta} \quad \dots\dots\dots\text{EC. III.19}$$

$$R_B = \frac{\beta(V_{BB} - V_{BE})}{I_C}$$

Sustituyendo

$$R_B = \frac{375(5 - 0.6)}{150 \times 10^{-3}} = 11.0 \text{ k}\Omega$$

Asi que el valor de la resistencia de base es 11.0kΩ.

De la misma manera se calcula la resistencia del colector:

$$R_C = \frac{V_{CC} - V_{CE}}{I_C}$$

Sustituyendo  $\dots\dots\dots\text{EC. III.20}$

$$R_C = \frac{(5 - 0.3)}{150 \times 10^{-3}} = 310 \Omega$$

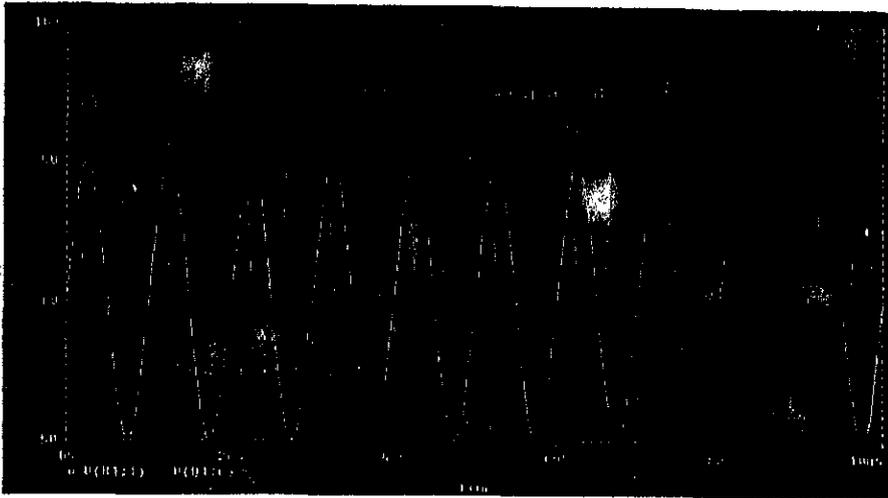


Figura III.70 Señal de entrada vs señal de salida del transistor.

El comportamiento del transistor se muestra en la figura III.70.

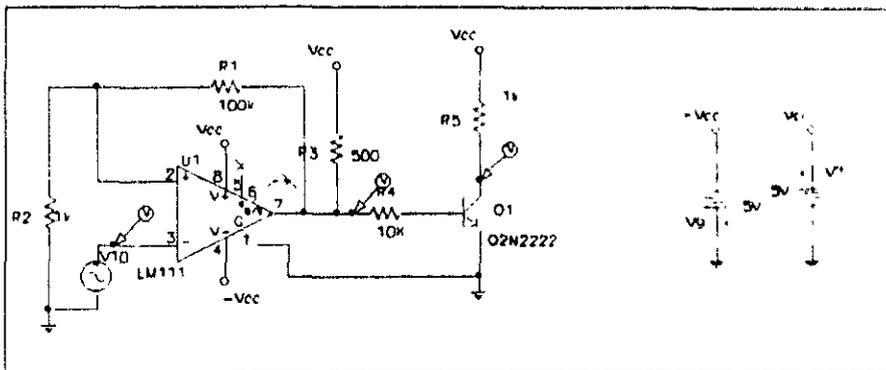


Figura III.71 Circuito del sistema de control (SC).

Finalmente, el comportamiento del circuito de la señal de control es similar a la gráfica del transistor en corte y saturación obtenida en la Figura III.70.

La señal que obtuvimos anteriormente se utiliza para manejar la alta impedancia del CI 74LS126. A la entrada de éste se encuentra la señal del oscilador, que nos proporciona una frecuencia de 20 MHz. Utilizamos esta señal como señal de reloj para el convertidor ADS901E. Según las especificaciones del fabricante, la digitalización comienza de acuerdo a los pulsos de reloj que detecta internamente. Si controlamos estos pulsos, podemos controlar el inicio de la conversión analógica a digital, lo que permite tener una cadena de valores constantes después de detectar la señal de video. Dicho de otra manera, quitamos los pulsos de sincronía, que no son de utilidad para nuestros propósitos, y solamente trabajamos la señal de video. El arreglo del comparador y el transistor detectan el cruce de la señal de video por cero, que realmente es el pulso de sincronía. Cuando termina el pulso de sincronía, el circuito de la señal de control (S7) da un pulso. El borde positivo es el que inhabilita la alta impedancia del 74LS126, permitiendo el paso de la señal del oscilador. La salida del CI tres estados va conectada a la entrada de la señal de reloj del convertidor ADS901E. Cuando el 74LS126 sale de la alta impedancia, el convertidor comienza a realizar la conversión.

### III.7. CIRCUITO INTEGRADO 74F126.

**Function Table MC54/74F126**

Inputs		Output
C	A	Y
H	L	L
H	H	H
L	X	Z

L = LOW Voltage Level  
 H = HIGH Voltage Level  
 X = Don't Care  
 Z = High impedance (off)

Tabla III.29 Tabla de verdad del circuito de tres estados.

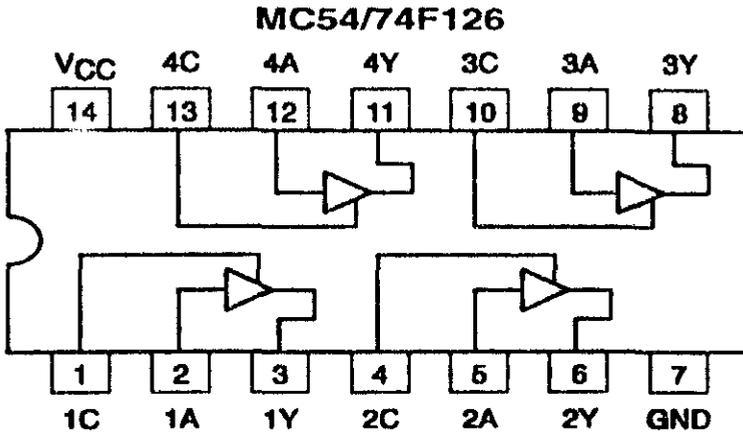


Figura III.72 Configuración de los pines del 74126.

De manera similar, las otras entradas del 74LS126 se utilizan para poner en alta impedancia a las señales que proporciona el microcontrolador. Las entradas se utilizan para evitar posibles errores en el bus de transferencia de datos, y de esta manera, dar protección a los dispositivos electrónicos contra señales que no se ocupan y se transmiten por el mismo bus. De esta forma, el microcontrolador envía las señales adecuadas para evitar conflictos en la

habilitación/innhabilitación en el funcionamiento de los dispositivos electrónicos que conforman la tarjeta.

PUERTO C		NIVEL BAJO "0"	NIVEL ALTO "1"	74LS126 (A, B, C)
C1	LOAD DE LOS CONTADORES	ACTIVA	DESACTIVA	Si es "1" se habilita (A)
C1	HABILITACION DE LA CUENTA DE LOS CONTADORES	ACTIVA	DESACTIVA	No importa
C2	HABILITA LA MEMORIA	ESCRITURA	LECTURA	En cualquier caso se habilita (B)
C3	SEÑAL DE RELOJ	NO IMPORTA	NO IMPORTA	Si es "1" se habilita (A)
C4	HABILITA LA SEÑAL DE RELOJ	LENTO	RAPIDO	No importa
C5	HABILITA AL CONVERTIDOR	ACTIVA	DESACTIVA	No importa

Tabla III.30 Señales de control que utiliza el microcontrolador.

De acuerdo a la tabla III.30 podemos introducir información al microcontrolador para que éste de las señales de control apropiadas a los dispositivos del sistema.

### III.8. LA WWW.

Por todos lados se habla de ella. En los periódicos, en la televisión, en los cafés. La Web está acaparando la atención de muchos sectores, pero, en realidad, ¿qué es la WWW?

La WWW, también conocida como 'La Web' ó 'W3', es la aplicación más popular de Internet y la abreviatura de World Wide Web, la 'telaraña' mundial de sitios gráficos y de multimedia vinculados en Internet. Es un gran conjunto de archivos de textos e imágenes relacionados y enlazados entre sí, de ahí el sobrenombre de 'telaraña'. El elemento o liga que mantiene unidas a estas páginas (que se conocen como Páginas Web) se denomina hipertexto. Precisamente este último concepto es la tecnología fundamental de la Web. Aunque el término general para la Web es 'hipertexto', es más exacto llamarlo hipermedios ('hipermedia') porque puede incluir cualquier tipo de objeto multimedia y no simplemente texto.

Efectivamente, una página Web puede contener texto, imágenes, sonido, video clips y, sobre todo, hipermedios, enlaces o vínculos con otras páginas. Dichos vínculos son más fáciles de ubicar, ya que siempre se presentan de un modo distinto al resto del texto. Por ejemplo, en color azul ó como texto subrayado. El hipermedio o vínculo permite que los archivos sean relacionados para que los usuarios de Internet puedan 'saltar' de una página a otra con solo hacer 'click' en las palabras o imágenes activas de la pantalla, sin necesidad de teclear las direcciones (URL's). Es decir, son los que finalmente nos permiten realizar exploraciones de manera sencilla.

### III.9. HTML Y EL AMPLIO MUNDO WEB.

Aun cuando el lenguaje HTML empezó como un experimento militar, sucesos recientes han transformado el amplio mundo de red de redes computacionales también conocido como Internet. Internet tiene un amplio y rápido crecimiento de información y de usuarios de computadoras en comunidades diversificadas. Actualmente, es posible hallar en Internet a usuarios de cualquier nacionalidad y que expresan las más diversas opiniones. Se pueden encontrar temas serios o frívolos, negocios y organizaciones no lucrativas, religión o pornografía. Los usuarios pueden crear documentos con lenguaje HTML y ponerlos a disposición del público en WEB.

### III.10. INTERNET.

Es una colección de redes, una red de redes de computadoras que comparten información digital mediante una vía común de protocolos de red y software. Actualmente, cualquier persona puede conectarse a Internet y comunicarse inmediatamente con otras computadoras y usuarios en la red.

¿Cuáles son las confusiones que han surgido en torno de Internet?  
¿Por qué se dice que es parecido a un bazar oriental? Las desventajas de Internet que se citan a menudo son:

- No está bien organizado.
- Tiene poca documentación para guiar al usuario.
- Puede llevar algo de tiempo usar por completo toda su potencial.

Internet empezó en los años sesenta como un experimento en el diseño de redes de computadoras robustas. La meta fue construir una red de computadoras que pudieran soportar la pérdida de varias máquinas sin perder la comunicación de las otras computadoras restantes. Esto se hizo en el Departamento de Defensa de los Estados Unidos, en donde existía un amplio interés en construir redes de información que pudieran soportar ataques nucleares. El resultado que obtuvieron de este nuevo desarrollo de redes fue maravilloso, pero tuvo sus limitaciones en tamaño y alcance. Solamente gente de la defensa e instituciones académicas podían tener acceso a lo que fue ARPANET (Advanced Research Project Agency Network of the Department of Defense).

Con la llegada de los modems de alta velocidad para comunicaciones digitales sobre líneas telefónicas, algunos individuos empezaron a conectarse y obtener ventajas de las avanzadas redes y las comunicaciones globales. Esto sucedió alrededor de 1993, año en que Internet realmente aumentó su poder. En los noventa, los negocios e individuos se vieron beneficiados por las comunicaciones digitales globales y, finalmente, presionaron a la mayor red de computadoras del gobierno de los Estados Unidos para que fundara Internet y abriera sus sistemas a un tráfico de información sin restricciones. Resulta interesante recordar que la red no fue diseñada para enviar información de tipo comercial; tal actividad fue prohibida en las computadoras de las universidades.

En Internet se puede intercambiar y compartir información. Muchos de los miembros originales de Internet continúan colaborando con su colección electrónica de documentos, que están disponibles para los nuevos usuarios sin ningún costo. Esto es muy atractivo,

ya que las comunicaciones globales ofrecen información y gran variedad de software sin costo.

En Internet se pueden usar varias herramientas de software si están instaladas previamente en las computadoras. Esto podría dar una tecnología insuperable a mucha gente. Sin embargo, el software que puede obtenerse en Internet no es como los paquetes de servicios en línea, como los que ofrecen America On Line, Prodigy o CompuServe. Internet está demasiado desorganizado, es externo al gobierno e instituciones académicas y poca gente sabe como usar el misterioso software, además de que no cuenta con el tiempo suficiente para explotar los documentos en su beneficio.

Cuando Internet se abrió a los negocios, algunas organizaciones de físicos como CERN y European Particle Physics Laboratory, liberaron un lenguaje y un sistema de distribución desarrollado por ellos para crear y compartir multimedia y documentos electrónicos a través de Internet. Así fue como nació **Hypertext Markup Language (HTML)** o lenguaje de marcación de hipertexto. Este lenguaje es utilizado para la creación de documentos de hipertexto e hipermedia; es el estándar empleado en el amplio mundo Web.

### **III.10.A. DIBUJOS DE BROWSER Y EL LENGUAJE HTML.**

Los browser son programas que despliegan el código HTML y que permiten acceder al amplio mundo Web. Aunque el primer navegador fue el NCSA Mosaic, hoy en día Microsoft Explorer y Netscape son, por mucho, los mejores navegadores (véase figura III.73). Los autores distribuyen sus trabajos como conjuntos de fragmentos, sonidos o texto y HTML une estos elementos. En el amplio mundo

Web existe un hypertext linking, gracias al cual los documentos automáticamente hacen referencia a otros documentos, localizados en cualquier lugar alrededor del mundo.

Algunos estudiantes del National Center for Supercomputing Applications (NCSA), la Universidad de Illinois y Urbana-Champaign escribieron un **WEB BROWSER** llamado mosaico. Al principio fue diseñado par ver documentos HTML. Posteriormente el software pudo construir herramientas para tener acceso a más recursos en Internet, tal como FTP.

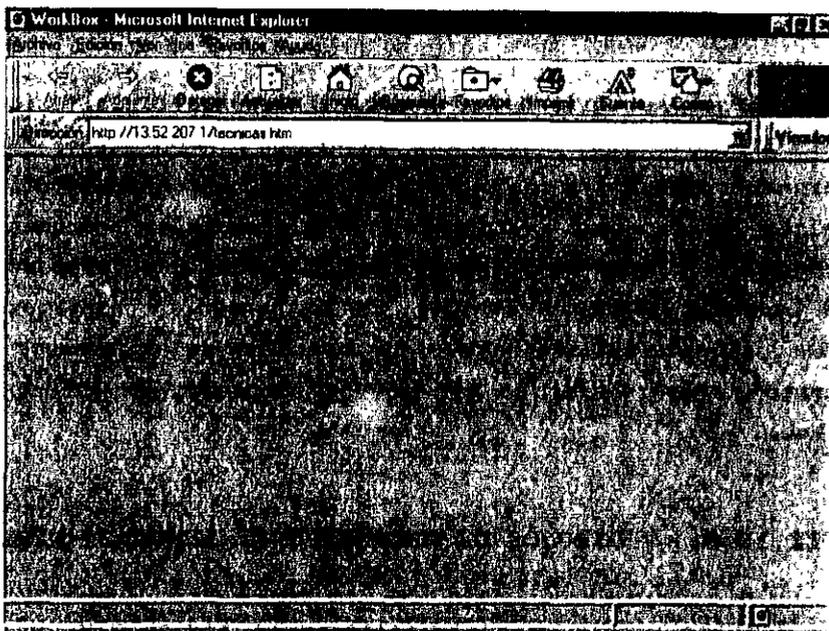


Figura III.73 Browser de Microsoft Explorer.

Millones de usuarios usan este software sin costo. Así, el desarrollo de Internet ha traído nuevos medios para el

intercambio de información y el comercio. Por ejemplo, los negocios pueden producir catálogos para millones de lectores alrededor del mundo.

Los creadores del mosaico dejaron la compañía NCSA y formaron Netscape Communications para producir los browsers:

- **Netscape Navigator.**
- **Microsoft Explorer.**

Este software es usado en Internet (véase figura III.73). Los empresarios y la mercadotecnia se han fortalecido en Internet y han crecido. De acuerdo a encuestas recientes elaboradas por ActuMedia Inc. (Peterborough, NH), la mitad de las empresas en Internet han tenido excelentes incrementos a un año de su lanzamiento.

Asimismo, Internet ocupa el primer lugar como medio para difundir cualquier información ya sea de tipo social o publicitaria. El poder de HTML y la distribución de información por las redes va más allá de la mercadotecnia. La educación se vio enriquecida porque los estudiantes tienen acceso a esta gran biblioteca mundial.

### **III.10.B. CÓMO SE CONECTA INTERNET.**

Cada computadora conectada a Internet tiene una dirección única IP (Identification Address), es decir, un número que es un formato definido por el Protocolo de Internet (IP). El protocolo es el estándar que define cómo son enviados los mensajes de una máquina a otra en la red. Una dirección IP está constituida por 4

numeros menores que 255 y unidos por puntos, por ejemplo: 13.52.207.1 o 13.52.207.79.

Mientras las computadoras se distribuyen solamente con numeros, la gente prefiere nombres. Por esta razón, cada computadora en Internet tiene un nombre propio otorgado por su propietario. Existen varios millones de maquinas en la red. Esto podria ser un problema para hallar el nombre de una maquina. La red de redes está dividida en grupos conocidos como dominios (**domains**), que a su vez se dividen en uno o mas subdominios (**subdomains**). Podemos escoger un nombre comun para nuestra computadora, pero todas las maquinas tienen un nombre de dominio como un sufijo separado, lo cual genera un fully qualified domain name (nombre de dominio calificado). Por ejemplo, el nombre de dominio calificado www.ora.com se traslada a la maquina llamada 'www', que es parte del dominio conocido como 'ora', el cual, a su vez, es parte de una rama comercial (com) de Internet. Otras ramas de Internet incluyen instituciones educacionales (edu), organizaciones no lucrativas (org), instituciones de gobierno (gov) y proveedores de servicios de Internet (net). Las computadoras y redes fuera de los Estados Unidos tienen un dominio propio que incluye dos letras de su nombre al final del dominio: por ejemplo, 'ca' para Canada, 'jp' para Japón, 'uk' para United Kingdom etc.

Las computadoras especiales, conocidas como servidores (servers), guardan tablas con los nombres de las máquinas y su dirección asociada numerica IP. El nombre del dominio debe ser registrado como una organización no lucrativa InterNIC. Una vez registrada, el propietario del nombre del dominio le envia con sus direcciones a otro domain name servers alrededor del mundo. Los dominios y subdominios tienen un nombre de servidor asociado.

Así, cada máquina es conocida únicamente por su nombre y su dirección IP.

### III.10.C. CLIENTES, SERVIDORES Y BROWSERS.

Internet se conecta con dos clases de computadoras:

- **Servidor en una red:** estación host de datos que proporciona servicios o aplicaciones a otras estaciones.
- **Cliente:** son computadoras que consumen o utilizan el contenido y la información ofrecida por los servidores. A estos equipos se les conoce como computadoras cliente.

En conjunto, esta relación se denomina Sistema Cliente/Sistema Servidor. Un programa cliente es el que un usuario ejecuta en su computadora para tener acceso a los recursos de Internet. Cuando el programa cliente necesita algo de Internet, lo busca en los programas del servidor. Un programa servidor recibe la petición del programa cliente y envía de regreso una respuesta. Por ejemplo, el navegador Web es un programa cliente y cada servidor Web de Internet es un programa servidor. Para acceder y desplegar información HTML, se ejecutan programas llamados browser en la propia computadora del cliente. Estos browsers llaman a un software especial del web server sobre Internet para acceder y obtener documentos electrónicos. Varios browsers para Web están disponibles sin ningún cargo y cada uno ofrece diferentes características. Por ejemplo, un browser como Lynx se ejecuta en clientes basados en caracteres y despliega documentos solamente en texto. Otros se ejecutan en clientes con despliegado de gráficos y sus documentos usan fonts proporcionales y gráficos de

color en 1024 X 768, 24-bit-per-pixel display. Algunos más, como Netscape Navigator, Microsoft Internet Explorer, NCSA Mosaic, Netcom's WebCruiser e Inter Con's NetShark, tienen características especiales que permiten obtener y desplegar una variedad de documentos electrónicos sobre Internet, incluyendo audio y multimedia.

### **III.10.D. EL FLUJO DE INFORMACIÓN.**

Todas las actividades en Web empiezan en el lado del cliente, cuando los usuarios inician su browser. El browser empieza ejecutando la homepage HTML (documento de HTML), ya sea para almacenamiento local o de un servidor sobre alguna red, tal como Internet. En el último caso, el browser del cliente consulta primero el nombre del dominio del servidor. Luego traslada a la home page HTML el nombre del servidor (por ejemplo, [www.ora.com](http://www.ora.com)) con una dirección IP, antes de enviar una solicitud al server sobre la red. Esta solicitud (y la respuesta del server) es formateada de acuerdo a lo que dicta el estándar HyperText Transfer Protocol (HTTP). Un servidor pasa la mayor parte de su tiempo atendiendo las solicitudes de documentos con una única dirección de este servidor. Una vez que recibe algo, el servidor verifica que la solicitud del browser sea autorizada para extraer documentos. Si esto ocurre, el servidor verifica el documento solicitado. Por lo general, el servidor transporta la petición hacia la dirección de la computadora del cliente en el menor tiempo posible, regresa al browser y actualiza el documento que le fue solicitado. Los directorios de documentos son tratados como documentos planos, aunque la mayoría de los browser gráficos muestran iconos de folders, que pueden ser seleccionados por el usuario para ver su contenido como subdirectorios. La mayor parte

de los browsers muestran un documento especial que aparece como un archivo de texto plano y contiene texto y marcas especiales llamadas **tags** (instrucciones del lenguaje HTML que van entre los simbolos < >). El browser procesa estos documentos de HTML, formatea el texto basado en los tags y muestra un archivo especial como imagenes. Los siguientes son ejemplos de tags Title y Head:

```
<TITLE> Titulo que se quiere ver </TITLE>
<HEAD> Encabezado que se quiere ver </HEAD>
```

El usuario lee el documento y selecciona otro documento como un hyperlink. El documento es mostrado cuando uno se posiciona en un texto hyperlink en donde aparece la figura de una mano. Se le da click, lo cual significa que tiene una liga hacia otro documento

### III.11. **BENEFICIOS DE LA WORLD WIDE WEB.**

Los browsers y los servidores HTTP no necesitan ser parte de www para funcionar. En realidad, el usuario no necesita estar conectado a Internet ni otra red para escribir documentos en lenguaje HTML y poder operar un browser. El usuario puede cargar y desplegar en su browser de cliente los documentos almacenados localmente en HTML, así como acceder a los archivos directamente. Esta separación es buena, ya que sirve para que se pueda probar un documento y distribuirlo más adelante. Los autores de HTML escriben y prueban localmente sus documentos antes de liberarlos para su distribución general, con el fin de evitar falsas hyperlinks (ligas). Las diversas organizaciones conectadas a Internet y www tambien tienen páginas privadas y una colección de

documentos para distribuir a sus clientes en su red local, o intranet (red interna).

Con documentos en HTML, los negocios y otras empresas pueden tener bases de datos personales, con fotografías, libros o colección de partes heliográficas, que se manejan de manera sencilla por usuarios autorizados y despliegan los datos en una computadora local.

### **III.12. ¿QUÉ ES HTML?**

HTML es un lenguaje planeado para documentos y tiene ligas para hacer referencias a otras ligas (hyperlink). HTML tiene una sintaxis con un orden especial: las direcciones no son desplegadas por el browser, pero le dicen a éste como desplegar el contenido del documento, incluyendo texto, imágenes y multimedia. También acepta documentos interactivos a través de textos especiales (hypertext). Estos conectan el contenido de sus documentos con otros documentos, ya sea a su computadora o alguna otra, así como con otros recursos de Internet -por ejemplo, el FTP.

#### **III.12.A. ESTÁNDARES DE HTML Y EXTENSIONES.**

La sintaxis y la semántica básicas de HTML son definidas en el estándar HTML versión 3.2 como un lenguaje joven. En la actualidad, se encuentra la tercera edición. No sería extraño que apareciera otra versión antes terminar de leer este texto. Uno nunca sabe cuándo una nueva versión estándar irá a tener frutos. Los creadores de los browsers confían en el estándar HTML para

programar el software que formatea y despliega documentos comunes de HTML.

Los autores usan un estándar para asegurar que están escribiendo documentos correctos en HTML. HTML empezó como una especificación informal, siendo usado por poca gente. Más y más autores empezaron a usar este lenguaje, lo cual ocasionó que se hiciera estándar para HTML.

### **III.13. EL CONSORCIO DE WWW.**

El consorcio del amplio mundo Web W3C (The World Wide Web Consortium) se formó para definir el estándar de versiones de HTML. Los miembros son responsables de revisar y modificar los estándares para ponerlos en circulación, creados específicamente para Internet.

El W3C tiene una amplia responsabilidad de estandarización para cualquier tecnología relacionada con el amplio mundo Web. Ellos manejan el estándar HTTP y desarrollan los estándares para proporcionar las direcciones de los documentos en el Web. Se encargan de solicitar los estándares actuales para realizar las extensiones de las tecnologías que ya existen en la Web, así como de la internacionalización del estándar HTML.

En la dirección <http://www.w3.org/> se puede encontrar una ruta de creadores de tecnología HTML. Otras direcciones son:

- [com.infosystems.www](http://com.infosystems.www)
- [com.infosystems.www.authoring.html](http://com.infosystems.www.authoring.html)
- [com.infosystems.www.authoring.images](http://com.infosystems.www.authoring.images)

### **III.14. LAS FUERZAS DE INGENIERÍA EN INTERNET.**

La responsabilidad que tiene la Internet Engineering Task Force (IETF) es definir y manejar cada aspecto de la tecnología de Internet en el amplio mundo Web. La IETF define todo lo relacionado con las tecnologías de Internet por medio de documentos especiales conocidos como Request for Comment o RFCs. Estos son numerados de manera individual para hacer referencia a ellos fácilmente. Cada dirección RFC tiene una tecnología específica en Internet. Para conocer más acerca de IETF y seguir el progreso de RFCs, se puede consultar la dirección <http://www.ietf.org/>.

### **III.15. CARACTERÍSTICAS DE HTML.**

Es importante hablar de las limitaciones de HTML. No es una herramienta para ser usada como un procesador de palabras, no proporciona soluciones de publicidad ni es un lenguaje de programación. Su propósito fundamental es definir la estructura y apariencia de los documentos para que sean enviados rápida y fácilmente en la red.

Antes de que el usuario pueda apreciar el lenguaje y comience a crear documentos con HTML, se deben tener algunas reglas: HTML está diseñado para estructurar documentos y hacer su contenido más accesible, no para formatear documentos y desplegarlos. HTML proporciona diferentes maneras para definir la apariencia de los documentos. La apariencia es muy importante (véase figura III.74) porque determina la manera en que los usuarios tienen acceso y utilizan esta información en sus documentos de HTML.

HTML presenta varias formas para organizar la estructura de los documentos sin considerar la apariencia final. Por ejemplo:

- Sección de encabezados
- Listas estructuradas
- Párrafos
- Reglas
- Títulos

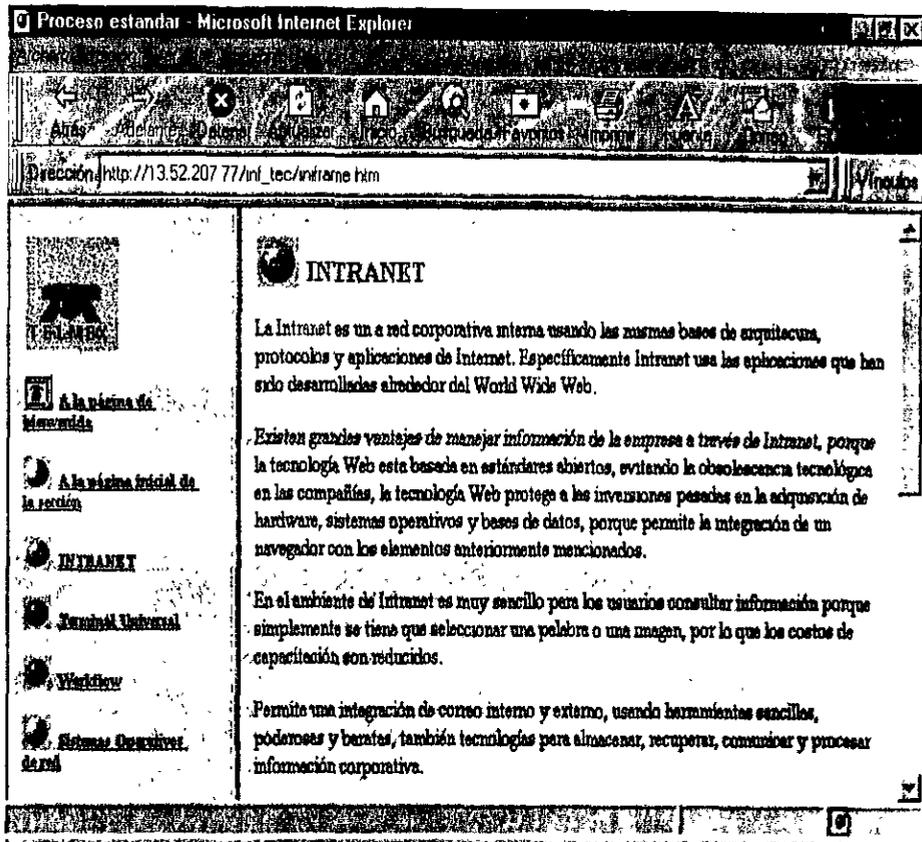


Figura III.74 Ejemplo de una forma HTML.

Se puede mostrar la información en diferentes formas. Aquí se ejecuta la forma HTML llamada iniframe.html. Se muestra el resultado de diferentes tags (instrucciones de HTML). Esta pantalla se encuentra dividida en dos partes por medio de tags llamados frames. Tanto el frame izquierdo como el derecho hacen llamadas a las siguientes formas HTML:

- Imágenes
- Ventanas
- Ligas

Todas ellas son definidas sin importar como son manejadas por el browser. Si el usuario trata al HTML como una herramienta de generacion de documentos, no lo podra explotar como es correcto ; por lo tanto, no tendra la oportunidad de construir documentos que puedan tener herramientas parecidas a Frame Work.

Es importante recordar estas limitaciones para no desperdiciar el tiempo forzando a HTML a llevar a cabo funciones para las que nunca fue desarrollado. En vez de esto, se puede usar HTML para los fines que fue diseñado, indicando la estructura de los documentos para que el browser pueda suministrar su contenido apropiadamente. HTML cuenta con abundantes tags (instrucciones), por ejemplo, para colocar titulos y encabezados que permiten indicar la semántica del contenido de su documento. Con esto los documentos se verán mejor y los lectores serán beneficiados.

### **III.15.A. LIMITACIONES ESPECÍFICAS DE HTML.**

En las versiones de HTML 3.2, algunos tags de HTML pueden ser usados por varios browsers como:

- Documentos de frame (un documento es dividido en cuadros .
- Documentos dinamicos scripted.
- Movimiento de texto.
- Texto absoluto y posicionamiento de imagenes.
- Ventanas.

Otros tags inexistentes en las versiones de HTML son:

- Notas al pie de página.
- Contenido de tablas automáticas.
- Índices.
- Encabezados.
- Tabuladores y otros caracteres.
- Anidación.
- Composición matemática.

Si dos usuarios usan la misma versión de browser en sus computadoras pueden reconfigurar el software; esto hará que sus documentos con HTML se vean diferentes. Por ejemplo, la apariencia de los documentos depende de la configuración de la pantalla que se tenga. En una configuración de 640 X 480 pixeles aparecen las flechas en la parte inferior para desplazarse hacia la derecha o hacia la izquierda, arriba o abajo. En monitores de más pulgadas, estas barras no aparecen porque todo se ve perfectamente en la pantalla.

El browser de Internet Explorer fue desarrollado en colaboración con Spyglass (de la tecnología NCSA) y se maneja con Windows 95 o 98. Netscape también se usa con Windows 95 o 98. Algunos tags corren sin problema en Netscape pero no sucede lo mismo en Internet Explorer.

### III.15.B. PRO Y CONTRA DE LAS VERSIONES.

Los vendedores buscan que sus productos sean diferentes y mejores que los de la competencia. Netscape e Internet Explorer son dos ejemplos de ello. Muchos autores se sienten seguros usando estos browsers. Las extensiones de HTML hechas por los parientes de Netscape o Microsoft arrastran palabras francesas a pesar de los esfuerzos de la Academia Francesa por evitar el uso indiscriminado de estos vocablos. La realidad es que los browsers están dejando de ser estándares. La W3C no se ha hecho cargo de este problema, por lo que los competidores de browser están desarrollando muchas extensiones para HTML sin importarles la calidad del producto.

### III.15.C. EVITANDO LAS EXTENSIONES.

En general, se recomienda al usuario que se resista a usar extensiones de HTML o en último caso, que las utilice muy poco en sus documentos, ya que corre el riesgo de perder una porción sustancial de lectores y se estará excluyendo a varios millones de personas que no utilicen Netscape o Internet Explorer para visualizar sus páginas.

No es posible excluir por completo las extensiones, ya que existen grados de dependencia de las extensiones de HTML. Por ejemplo, en una versión la regla que aparece en la pantalla se encuentra en posición horizontal al ejecutar la página HTML. En otra versión puede aparecer vertical. Lo mismo sucede con los fonts, ya que su tamaño pueden variar y al colocar un texto junto a un dibujo, se ven a los lados del dibujo o arriba de éste, dependiendo de la versión del browser.

### **III.15.D. HERRAMIENTAS PARA DISEÑAR FORMAS HTML.**

En la actualidad, existen herramientas que permiten diseñar páginas HTML para personas que nunca han trabajado con este tipo de software. El usuario coloca a su gusto el título, los dibujos, las cajas, los botones, las ligas etc. Algunos ejemplos de software para desarrollar formas en HTML son CLARIS PAGE en el sitio [claris.com/support/products/claris.page](http://claris.com/support/products/claris.page). El programa se puede "bajar" y puede obtenerse una licencia por un mes gratis.

- **CONDICIONES ESENCIALES.**

Se necesita un editor para colocar las instrucciones de HTML, un browser para verificar el trabajo y una conexión para tener acceso a Internet.

### **III.15.E. PROCESADOR DE PALABRAS O EDITOR HTML.**

Algunos autores utilizan procesadores de palabras, como el notepad, para escribir los tags (instrucciones) de HTML. Después, insertan este código dentro de las formas HTML. Los procesadores de palabras tienen mejores herramientas para escribir el texto, ya que cuentan con funciones específicas como las sangrías, fuentes de letra, márgenes, justificaciones del texto, estilos del texto etc. Es necesario tener cuidado porque una forma HTML no debe contener caracteres de control ni acentos en las palabras. EL notepad es una buena herramienta para colocar instrucciones en las formas de HTML. Si quiere utilizar un editor en particular, primero debe examinarlo y realizar pruebas para no tener problemas.

### III.16. ¿QUÉ ES UNA BASE DE DATOS?

Es un lugar centralizado donde se almacena información de todas las clases. Las bases de datos simplifican la colección y mantenimiento de datos, facilitando las búsquedas y desplegando datos en una amplia variedad de formatos.

La información en una base de datos se almacena dentro de **TABLAS** con las características especiales que la relacionan con un tópico en particular. Cada tabla contiene muchos campos que agrupan cierto tipo de información, como apellidos o el nombre de una ciudad. Mediante las tablas, la información es relacionada de manera simple o compleja para ser organizada de manera... Cada renglon en una tabla representa un registro.

Por ejemplo, muchas escuelas tienen una base de datos de todos sus alumnos gracias a la que es posible saber el grado escolar cursado por cada alumno. Esta base de datos tiene una tabla que puede llamarse 'alumno' y contiene campos como número de cuenta, nombre, apellido paterno, dirección, teléfono, etc.

REGLON	No_cta	Nombre	Ape_patern	Dirección	teléfono
REGLON	9752632	Angel	Arias	Viella 9	5652318
REGLON	9652488	Lorena	Marzi	Pragua 6	4252521
REGLON	9125252	Ana	Cedillo	Vigo 9	4523698

Tabla III.31 Ejemplo de una base de datos.

Cada renglon representa un registro y cada columna es un campo de la tabla.

### III.16.A. STRUCTURED QUERY LANGUAGE (SQL).

**Lenguaje estructurado para consultas.** Es un lenguaje estandar que se usa para interactuar con bases de datos relacionales. Las características avanzadas de SQL son extremadamente poderosas y aprender SQL es relativamente simple. Se puede aprender SQL sin saber cómo usar Cold Fusion (software para aplicaciones Web). Raramente se necesitan escribir oraciones complicadas en SQL, pues cuenta con herramientas que permiten hacer consultas de base de datos como Access, Foxpro, y colocar estos resultados en otras aplicaciones como Cold Fusion.

#### III.16.A.1. TRAER DATOS (SELECT).

Las oraciones de SQL consisten de operaciones, seguidas por un conjunto de cláusulas y argumentos que especifican como sacar los datos. La operación más importante es **SELECT**, usada para seleccionar datos en los registros de las tablas en la base de datos.

##### **Sintaxis**

**SELECT fields FROM table**

Los campos (fields) representan las columnas de la tabla separados por una coma. La tabla (table) es el nombre de la tabla.

Por ejemplo, para seleccionar los campos del nombre, apellido paterno y teléfono de la tabla *alumnos*, tendremos:

```
SELECT ape_paterno,telefono FROM alumnos
```

Si se quieren seleccionar todos los campos de una tabla se usa \*:

```
SELECT * FROM table
```

### III.16.A.2. SELECCIONAR REGISTROS ESPECÍFICOS (WHERE).

Para seleccionar información que cumpla con ciertas especificaciones, se usa la clave **WHERE** dentro del SELECT.

#### **Sintaxis**

```
SELECT fields FROM table WHERE criterio
```

**CRITERIO:** Representa oraciones de comparación (=, >, <, etc.) y operadores booleanos (AND, OR, NOT, etc.) que determinan qué registros deberán ser regresados de la consulta.

Por ejemplo, para seleccionar información cuyo apellido paterno sea Maldonado y tenga número de cuenta mayor de 79000000:

```
SELECT * FROM alumnos  
WHERE no_cuenta > 79000000 AND ape_paterno='Maldonado'
```

Para seleccionar información de los alumnos cuyo apellido paterno sea 'Jimenez ' o 'Cedillo ':

```
SELECT nombre,ape_paterno,teléfono  
FROM alumnos  
WHERE ape_paterno='Jiménez ' OR ape_paterno='Cedillo '
```

### III.16.B. ORDENAMIENTO DE LAS CONSULTAS (ORDER BY).

En muchas ocasiones, se requiere que el resultado de la selección esté en un orden específico. Para esto se usa la cláusula ORDER BY dentro de SELECT.

#### Sintaxis.

```
SELECT fields FROM table
WHERE criterio
ORDER BY FieldOrder
```

Donde **FieldOrder** representa los campos de las columnas en las tablas, que se ordenan separadas por comas especificando si es en orden ascendente (ASC) o descendente (DESC). Se puede usar ORDER BY, ya sea que esté o no presente la cláusula WHERE.

Ejemplo: seleccionar de la tabla alumnos toda la información ordenada por apellido paterno:

```
SELECT * FROM alumnos
ORDER BY ape_paterno
```

En forma descendente sería:

```
SELECT * FROM alumnos
ORDER BY ape_paterno DESC
```

SQL también soporta un amplio rango de otras aplicaciones para base de datos como: borrar datos, crear tablas, borrar tablas, insertar datos y otras funciones.

### III.17. PLATAFORMA DE NETSCAPE Y JAVASCRIPT.

Este capítulo es un resumen de la plataforma de Netscape y JavaScript, la cual está basada en un lenguaje script de objetos para aplicaciones de cliente servidor. JavaScript tiene objetos predefinidos del lenguaje core (common object request).

JavaScript permite crear aplicaciones que corren en Internet. Por ejemplo, se puede observar su uso en las páginas dinámicas de HTML, que procesan la información del usuario y mantienen los datos usando objetos especiales de Javascript. A través de funciones de JavaScript como LiveConnect, se puede acceder a las aplicaciones de objetos distribuidos como Java y CORBA.

JavaScript comparte las mismas características del lenguaje del lado del servidor (server-side) y del cliente (client-side). Este lenguaje corresponde al ECMA-262, es un lenguaje script adecuado a los estándares de Europa y contiene un conjunto de objetos tales como arreglos de ordenamientos por fecha, por mencionar algunos. Asimismo, cuenta con características para ejecutar expresiones, oraciones y operaciones. Aunque en el server-side y en el client-side JavaScript usa las mismas funciones, en algunos casos tienen una aplicación diferente. Los componentes de JavaScript se ilustran en la figura III.75.

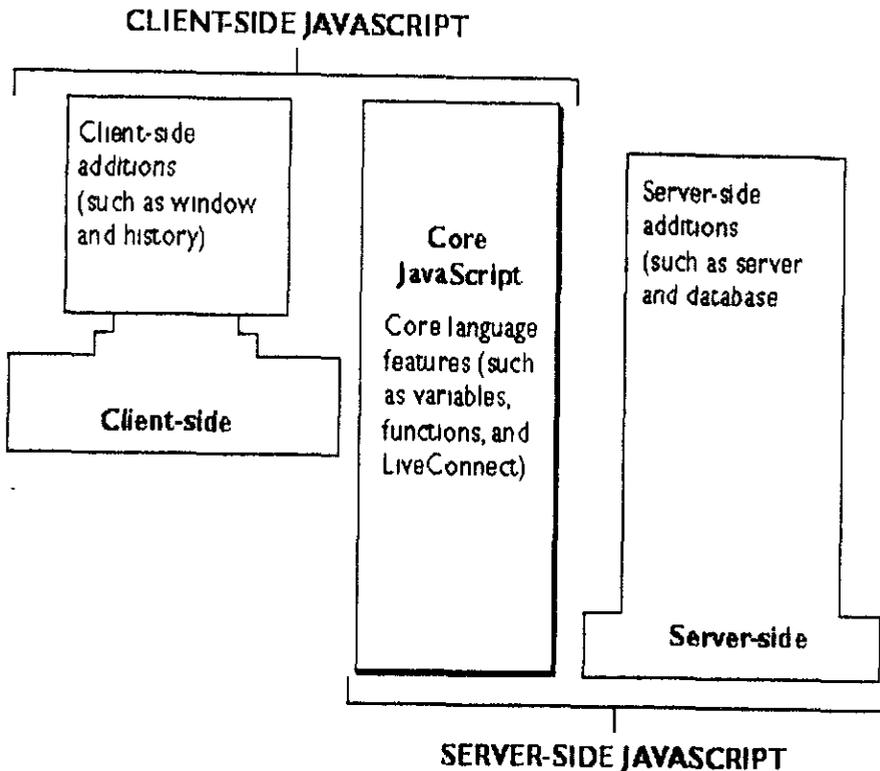


Figura III.75 Componentes de JavaScript.

En el client-side JavaScript (o navegador de JavaScript) se encuentra el lenguaje core con mas objetos para su funcionamiento. Tales objetos son funciones predefinidas que solamente pueden ser ejecutadas por JavaScript en el lado del servidor. Simultáneamente, el lenguaje core es incrustado directamente en páginas HTML para ser interpretado por el browser en cierto tiempo. Las aplicaciones de JavaScript son compiladas antes de ser ejecutadas. En la siguiente sección se describe como trabaja JavaScript en el client-side y en el server-side.

### III.18. EL CLIENT-SIDE DE JAVASCRIPT.

Los Web browser tales como Netscape Navigator 2.0 (y últimas versiones) pueden interpretar el código colocado del client-side JavaScript en una página HTML. Cuando el browser (o cliente) solicita tal página, el servidor le envía el contenido completo del documento, incluyendo el código de HTML y JavaScript sobre la red. El cliente lee toda la página, desplegando los resultados HTML, y ejecuta el código de JavaScript. Este proceso produce el código que se muestra a continuación.

\*\*\*\* ARCHIVO QUE SE EJECUTA CUANDO SE LLAMA A LA FUNCIÓN EmitError: \*\*\*\*

```
function EmitError(Message)
{
    write("<H1>Error:</H1>" + Message);
}
```

\*\*\*\*\* ESTE ARCHIVO CONTIENE FUNCIONES QUE DEBEN EJECUTARSE EN EL SERVIDOR y SE ENCUENTRA ENTRE LOS TAGS <SERVER> y </SERVER>, ASIMISMO AL FINAL REALIZA UNA LLAMADA A LA FUNCIÓN EmitError: \*\*\*\*\*

```
<HTML>
<HEAD>
<TITLE>Procesar Archivo para identificar el color</TITLE>
</HEAD>
<BODY>
<h1> Proporcionar datos para procesar </h1>
<SERVER>
    if (!arch_ent.eof()) {
        for (i=0; i < max_serie ; i++)
```

```

    arreglo[i] /= num_series;
    database.connect("ODBC","ensayo", "admin", "admin","");
    if(!database.connected())
    {
        EmitError("Error al intentar conectar la base de datos");
    } else {
        c_curva = database.cursor("SELECT color, indice, valor from
curva", true);
        c_curva.color=8;
        for (j=0; j < max_serie; j++) {
            c_curva.indice = j;
            c_curva.valor = arreglo[j];
            c_curva.insertRow("curva");
        }
        c_curva.close();
    }
    database.disconnect();
    arch_sal = new File("salida.txt");
    if (arch_sal.open("w")) {
        for (j=0; j < max_serie; j++)
            arch_sal.writeln(String(arreglo[j]));
        arch_sal.close();
    }
    delete arch_sal;
}

var ErrorText = "";
arch_ent = new File("c:/tesis/leer/entrada.m");
if (arch_ent.open("r")) {
    procesa_datos();
} else {

```



HTML (que contienen código JavaScript tanto en el client-side como en el server-side) y archivos JavaScript. Así se compilan estos archivos, quedando un archivo ejecutable. Las instrucciones que se encuentran dentro de los tags <server> </server> son ejecutadas en el servidor; las demas se ejecutan en el lado del cliente.

```
Function Substitute (guess, word, answer) {  
    Var result = "";  
    Var len= word.length;  
    Var pos= 0;  
    While (pos < len ) {  
        Var word_char = word.substring( pos, pos +1);  
        Var answer_char = answer.substring( pos, pos + 1 );  
        If (word_char == guess) result = result + guess ;  
        Else result = result + answer_char;  
        Pos = pos + 1;  
    }  
    return result;  
}
```



Se compilan las instrucciones  
Quedando en un archivo  
Ejecutable

```

<HTML> <HEAD> <TITLE> hangman </TITLE></HEAD>
<BODY> <H1> hangman </H1>

<SERVER>
if client.gamemo == null {
    client.newgame = 'true'
}

</SERVER>
Usted ha usado las siguientes letras hasta ahora:
<SERVER> write 'client.used' </SERVER>
<FORM METHOD="POST" ACTION="hangman.html">
<P>
Cual es su invitado ?
<INPUT TYPE="TEXT" NAME="guess" SIZE="1">
</BODY> </HTML>

```

### III.20. ARQUITECTURA DE APLICACIONES JAVASCRIPT.

En la sección anterior se vio que las aplicaciones de JavaScript tienen partes que se ejecutan en el cliente y en el servidor. En general, muchas aplicaciones de JavaScript usan Live Wire Database Service para conectar las aplicaciones a una base de datos que las relacionan de alguna manera. Debido a lo anterior, se puede pensar que las aplicaciones JavaScript tienen tres "capas" en la arquitectura de cliente servidor, como se ilustra en la figura III.76.

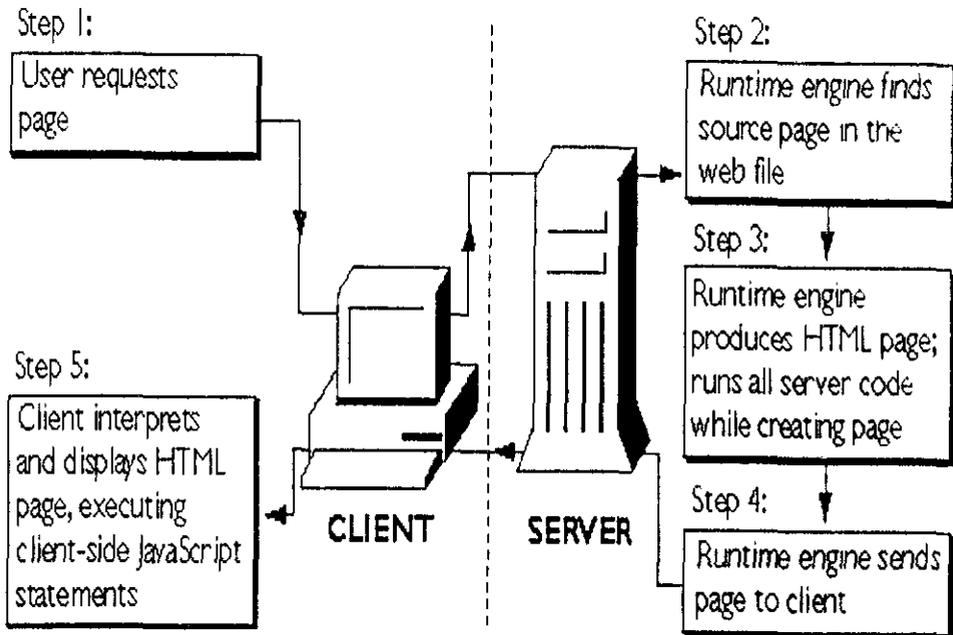


Figura III.76 Arquitectura cliente servidor para un medio ambiente de aplicaciones JavaScript.

Las capas son:

- **WWW Clientes** (Por ejemplo clientes Netscape Navigator). Esta capa provee una interface a un usuario final en una plataforma cruzada (que puede ser de varios tipos). Contiene aplicaciones lógicas como las reglas de validación de datos desarrollados en el client-side JavaScript. Los clientes pueden estar dentro o fuera de firewall (sistema de protección para las conexiones de Internet).
- **Cliente Netscape WWW server/database client.** Consiste de un Server Netscape, con JavaScript habilitado en el lado del

servicios. Contiene aplicaciones lógicas, maneja la seguridad, el control de accesos de múltiples usuarios a la aplicación, usando JavaScript en el server-side. Permite acceso a aplicaciones. El servidor Web también actúa como cliente para permitir servidores instalados en la base de datos.

- **Servidores de base de datos.** Por sus características, este nivel se encuentra en el apartado siguiente

### III.20.A. SERVIDORES DE BASE DE DATOS

Esta capa consiste en servidores de base de datos que se ejecutan en estaciones de trabajo. Contiene todas las bases de datos y reglas de integridad requeridas por las aplicaciones. Este nivel está dentro del sistema firewall, y puede proporcionar un nivel de seguridad aparte de los ya proporcionados por el servidor WWW. Netscape Enterprise Server soporta el uso de los siguientes servidores de base de datos:

- ODBC
- DB2
- INFORMIX
- ORACLE
- SYBASE

El medio ambiente client-side JavaScript corre como cliente dentro de WWW, mientras que el medio ambiente server-side JavaScript corre como servidores web de Netscape con acceso a uno o más servidores de bases de datos.

La figura III.77 muestra más detalles sobre cómo el medio ambiente server-side JavaScript cabe perfectamente dentro de un servidor web de Netscape.

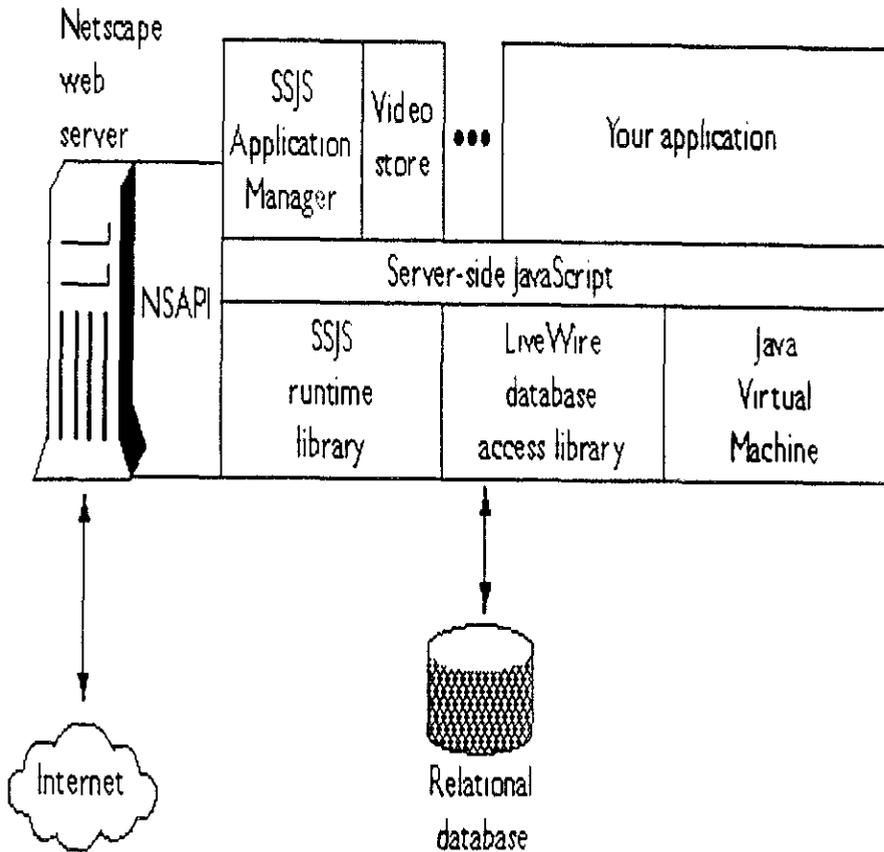


Figura III.77 El server-side JavaScript cabe perfectamente dentro de un servidor web de Netscape.

El JavaScript Application Manager corre en la parte superior del server-side JavaScript. Dentro del servidor web, el server-side JavaScript runtime esta formado por tres componentes:

- **JavaScript library.** Esta biblioteca ofrece componentes basicos de JavaScript. Un ejemplo es Session Management Service, que proporciona objetos predefinidos para ayudar a manejar su aplicacion , compartir informacion entre el cliente y el servidor.
- **LiveWire database access Library.** Estos componentes se orientan hacia funciones de server-side JavaScript con clases , objetos que proporcionar acceso a servidores externos de base de datos.
- **Java virtual machine.** No se usa solamente para JavaScript; cualquier aplicacion de Java que corra en el servidor usa esta maquina virtual. Esto permite aplicaciones para acceder a clases de Java, usando las funciones de JavaScript's LiveConnect.

Cuando se crean las aplicaciones, las paginas con código HTML pueden contener al código JavaScript en el server-side y en el client-side. Dentro del código fuente de HTML, el client-side JavaScript es delimitado por el tag SCRIPT y el server-side JavaScript por el tag SERVER. Si un archivo contiene código HTML y JavaScript en el servidor JavaScript se puede compilar dentro de un simple archivo ejecutable. Este archivo ejecutable es llamado archivo web, con la extension .web.

El compilador de JavaScript regresa el código fuente HTML dentro de una plataforma independiente de código de bytes, verificando y compilando las oraciones de server-side JavaScript.

### III.21. CONSTRUCCIÓN DE UNA APLICACIÓN JAVASCRIPT.

Se necesitan seis pasos básicos para construir una aplicación JavaScript:

- Crear el archivo fuente. Los archivos fuente pueden ser archivos HTML con código de JavaScript. Estos archivos también contienen el código de JavaScript o archivos Java.
- Construir la aplicación usando el compilador de la aplicación de JavaScript para crear el archivo ejecutable con extensión .web. Los archivos fuente de Java se compilan dentro de archivos clase.
- Colocar el archivo web, cualquier clase de archivo compilado en Java, imágenes, etc. Es necesario colocarlo en su directorio dentro del servidor.
- Instalar las aplicaciones por primera vez, usando JavaScript Application Manager.
- Correr la aplicación dando click en el botón Run del Application Manager o cargando la aplicación URL en el browser. Por ejemplo, para correr World, cargamos <http://server.domain/world/>.

- Después de que se ha examinado la aplicación, es necesario dar los permisos de acceso para los usuarios. Véase la figura III.7b.

Antes de construir una aplicación en JavaScript se necesita habilitar el motor runtime en el servidor y se debe proteger el JavaScript Applications Manager contra accesos no autorizados.

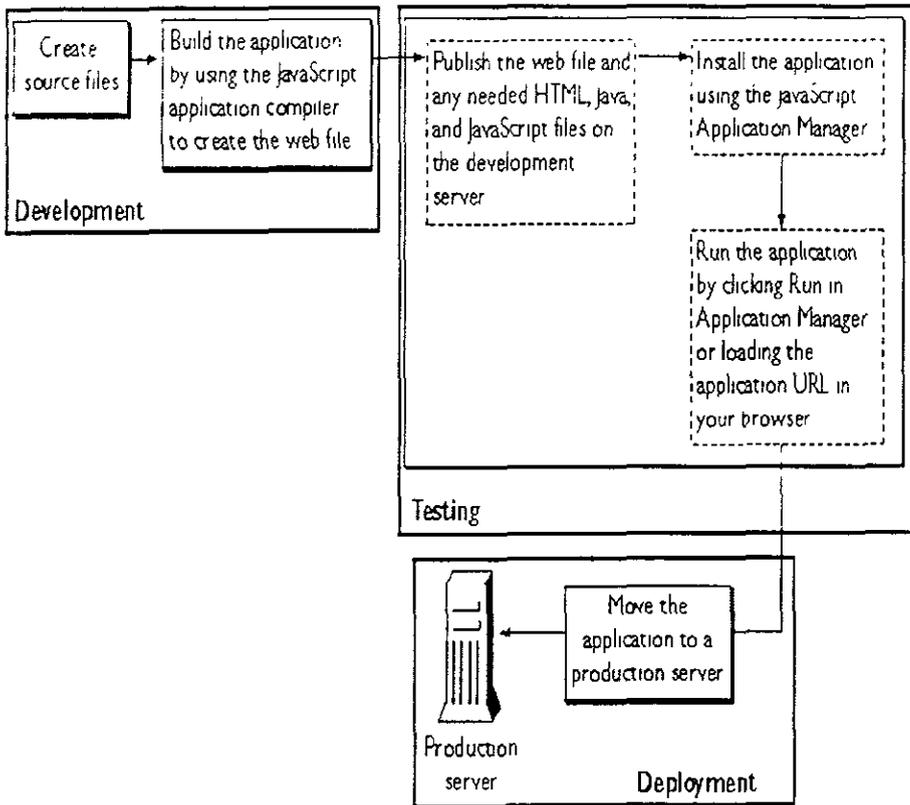


Figura III.7a Construcción de una aplicación en JavaScript.

En resumen, para crear una aplicación JavaScript, se requiere familiarizarse con el JavaScript Application Manager y realizar las siguientes tareas:

- Agregar una aplicación nueva de JavaScript.
- Compilar la aplicación.
- Ejecutar los siguientes pasos en este orden: Stop, start and restart en una aplicación instalada.
- Ejecutar y verificar la aplicación activa.
- Quitar e instalar la aplicación. Esto se hace cuando se realizan cambios. Vease la figura III.79.

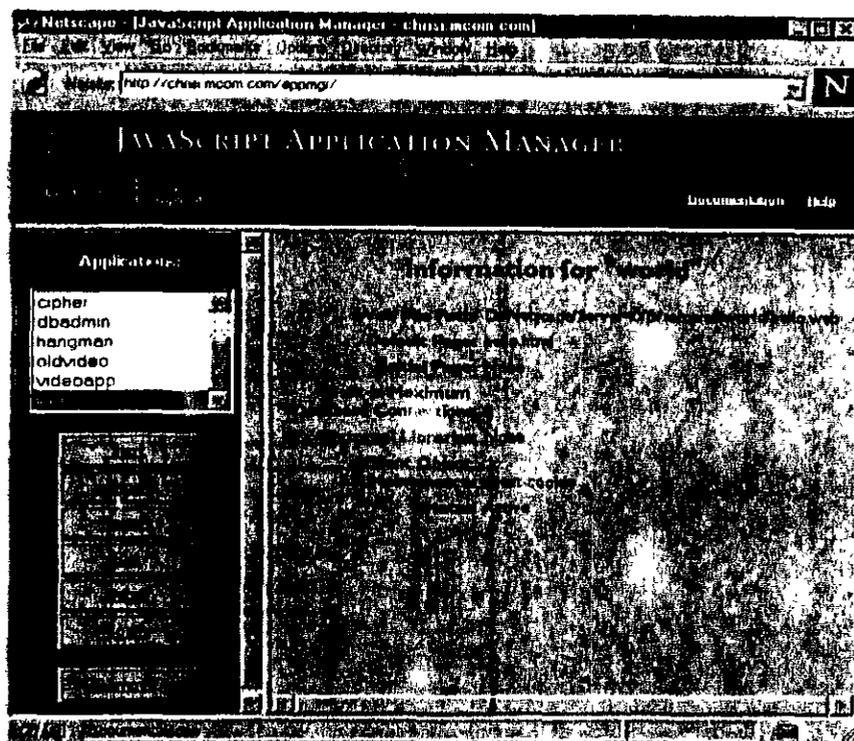


Figura III.79 Aplicación de JavaScript.

### III.22. RESUMEN DE RUNTIME PROCESSING.

Una vez que el usuario ha instalado e iniciado una aplicación en JavaScript, el usuario puede comenzar a trabajar. El procedimiento a seguir es como se indica a continuación:

- Un usuario puede tener acceso a la aplicación URL con algún web browser, por ejemplo Netscape Communicator. El web browser envía una petición del cliente hacia el servidor, que nos proporcionará la página de la aplicación solicitada.
- Si la petición es una página bajo la aplicación URL, el motor de JavaScript runtime corre en el servidor y busca información en el archivo web correspondiente a esa URL.
- El runtime construye una página HTML para enviarla al cliente como respuesta. El runtime regresa el código de bytes asociado con los tags del servidor del código original del archivo fuente de HTML, creando una página HTML basada en este código de bytes y cualquier otro HTML encontrado en el original.
- El motor de runtime envía al cliente la nueva página HTML (la cual contiene oraciones de client-side JavaScript).
- El motor JavaScript runtime dentro del web browser interpreta cualquier oración de client-side JavaScript y los formatos de salida de HTML. Además, despliega los resultados al usuario.
- Después de haber desarrollado y revisado la aplicación, se necesita caracterizar el runtime para que sólo los usuarios responsables tengan acceso a él. Véase la figura III.80.

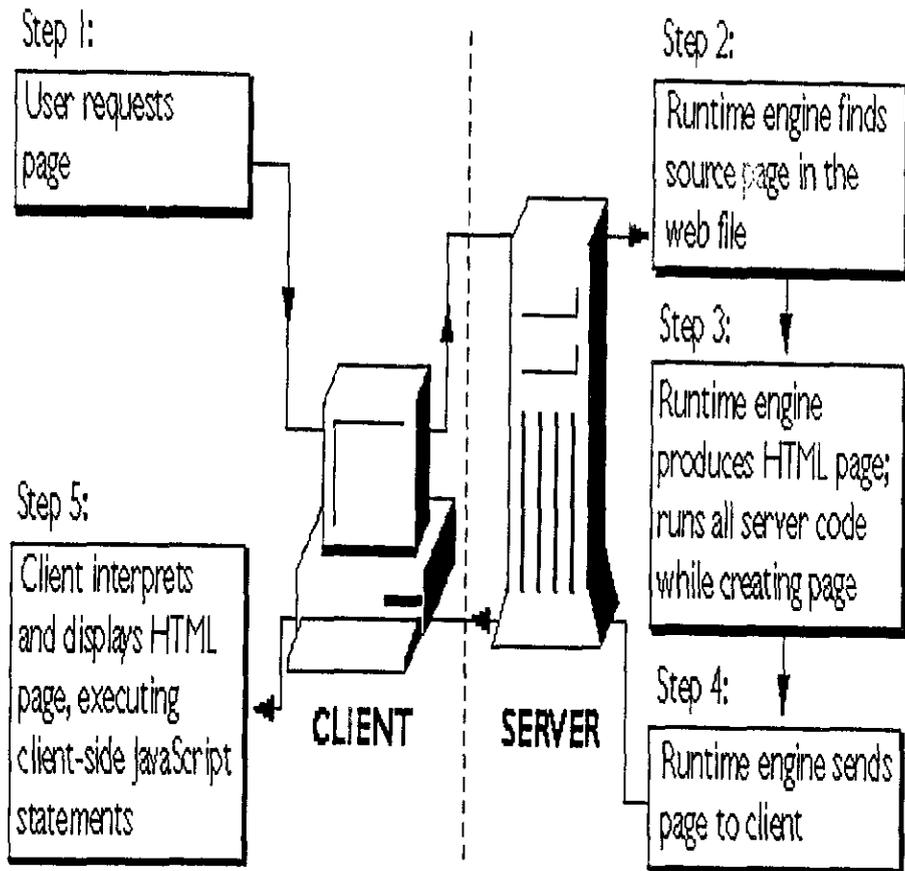


Figura III.80 Resumen de Runtime processing.

#### IV RESUMEN Y CONCLUSIONES.

En este ultimo capitulo analizaremos las características del trabajo desarrollado en la parte de hardware y software. Comentaremos el analisis de resultados de la señal que nos sirve para caracterizar al color que obtiene la cámara, la manera en que se logra la interpretación de los datos a través de la computadora.

En el apéndice C se da una explicación del diseño de fuentes de acuerdo a los dispositivos de regulación de voltaje de tres terminales LM 317, 7800, 7900..

##### IV.1.A. Especificaciones

La cámara de video proporciona las siguientes características eléctricas. El estándar de la señal de video que maneja la cámara es el de 1 Vpp 75 (EIA RS- 330). Si se digitaliza la señal con los pulsos de sincronía horizontal, se presenta un problema: no se tiene control sobre lo que realmente es la caracterización del color en el software. Al realizar la conversión de analógico a digital, el pulso de sincronía aparece en la reconstrucción de la señal como una señal, tal cual, pero de otra manera, el software no la detecta como una señal meramente analógica que se utiliza para asegurar que la imagen se establezca en el receptor (monitor) a partir de la detección del pulso de sincronía.

Para el software necesitamos solamente la señal de video sin el pulso de sincronía horizontal. Esta característica se contempló en el diseño electrónico de la tarjeta.

Al quitar la señal de sincronía logramos dar una cadena de caracteres constantes, lo cual ayuda a reconocer la señal de

video, el software se ve limitado por la señal analógica, ya que se trabaja con la señal digitalizada de video y cuando detecta la cadena de caracteres constantes nos puede indicar cuándo empieza una señal y cuándo termina está.

Otro punto importante en el diseño de la tarjeta de adquisición de datos que influyó directamente en la elección del microcontrolador fue la velocidad de operación. En este caso se seleccionó la Familia de PICS, que son microcontroladores de Arquitectura RISC Harvard fabricados por Microchip Technology. De acuerdo a las características del diseño, el que presentó mayor velocidad de operación fue el PIC 17C44, velocidad que es la más adecuada para nuestras aplicaciones.

Los microcontroladores de Microchip Technology de la serie PIC16XX/PIC17XX proporcionan una velocidad de ejecución significativamente mayor que cualquier otro microcontrolador de 8-bits. El desempeño de los PIC16CXX/17CXX puede atribuirse principalmente a su arquitectura Harvard (el espacio de programa de memoria y el espacio de memoria de datos de 8-bits se encuentran separados). También utilizan dos estados de entubamiento (pipelining) para almacenar la instrucción y realizar la ejecución. Todas las instrucciones son ejecutadas en un solo ciclo (120 ns @ 33 MHz de reloj) salvo las derivaciones del programa, que toman dos ciclos. Sólo hay que recordar la existencia de 33 a 58 instrucciones para programar los PIC's dependiendo del microcontrolador. El de mayor número de instrucciones es el PIC17C44.

La estructura y organización interna del PIC permiten que la instrucción sea optimizada para cualquier tamaño. Por

ejemplo, se pueden cargar 8-bits de datos en un ciclo. Esto se logra porque no hay ningún conflicto entre la instrucción de captura y el dato a ser capturado, la diferencia de la arquitectura de Von Neumann, y porque la instrucción es bastante amplia para soportar los 8-bits de datos.

Pero tales características deben ser justificadas respecto a otros dispositivos, por lo que se realizaron comparaciones entre el PIC 17C44X/XX @ 20 Mhz y los siguientes microcontroladores de su genero:

- SGS-Thomson ST62 @ 8 Mhz
- Motorola MC68HC05 @ 4.1 Mhz
- Intel 8051 @ 10 Mhz
- Zilog Z86CXX @ 12 Mhz
- Nacional COP800 @ 20 Mhz

Se consideraron varias rutinas de ejecucion de comandos para efectuar las comparaciones.

#### **IV.1.A.1. Cambio de Nibbles**

En esta rutina se toman dos bytes. Se intercambian el nibble inferior y el nibble superior. Ambos se encuentran en el mismo registro en algun espacio de RAM. Una vez cambiados, vuelven a ocupar la misma localidad de RAM o el mismo registro.

<b>PIC16C5X/XX</b>				<b>COP800</b>			
		Byte/Words	Cycles			Byte/Words	Cycles
SWAPF	REGHI,W	1	1	X	A,[B+]	1	2
IORWF	REGLO	$\frac{1}{2}$	$\frac{1}{2}$	SWAP	A	1	1
				OR	A,[B]	1	1
				X	A,[B]	$\frac{1}{4}$	$\frac{1}{5}$
			0.4 $\mu$ s				5 $\mu$ s
				B is pointing to the higher BCD digit initially. After auto-increment it points to the lower BCD digit.			
<b>ST62</b>				<b>MC68HC05</b>			
		Byte/Words	Cycles			Byte/Words	Cycles
LD	A,REGHI	2	4	LD	REGHI	2	3
RLC	A	1	4	ROLA	A	1	3
RLC	A	1	4	ROLA	A	1	3
RLC	A	1	4	ROLA	A	1	3
RLC	A	1	4	ROLA	A	1	3
ADD	A,REGLO	2	4	ADD	REGLO	2	3
LD	REGLO,A	2	4	STA	REGLO	2	4
		$\frac{10}{10}$	$\frac{28}{28}$			$\frac{10}{10}$	$\frac{22}{22}$
			45 $\mu$ s				10 $\mu$ s
REGHI and REGLO are registers addressable by short direct addressing mode							
<b>Z86CXX</b>				<b>8051</b>			
		Byte/Words	Cycles			Byte/Words	Cycles
SWAP	REGHI	2	8	MOV	A,Rx	1	1
OR	REGHI,REGLO	$\frac{2}{4}$	$\frac{8}{14}$	SWAP	A	1	1
				ORL	A,Ry	1	1
			5.33 $\mu$ s	MOV	Ry,A	$\frac{1}{4}$	$\frac{1}{4}$
							2.4 $\mu$ s
REGHI and REGLO are addressable via the working register addressing mode				Register Rx contains higher BCD digit Ry holds lower BCD digit			

Ejemplo IV.1.A.1 Cambio rotación del nibble más significativo al menos significativo, almacenado nuevamente en el mismo registro.

#### IV.1.A.2. Ciclo de Control

Este es un ejemplo sencillo de una malla de control en donde un registro contiene un ciclo de conteo que es disminuido hasta cero. El registro presenta una bifurcación para continuar con el ciclo.

PIC16C5X/XX				COP800			
	Byte/Words	Cycles			Byte/Words	Cycles	
DECFSZ	COUNT	1	1.2	DRESZ	COUNT	1	3
GOTO	BEG LOOP	1	2	JP	BEG LOOP	1	3
		$\frac{2}{}$	$\frac{3.2}{}$			$\frac{2}{}$	$\frac{6}{}$
			0.6 $\mu$ s/0.4 $\mu$ s	COUNT is Register (RAM F0h-FFh)			6 $\mu$ s
ST62				MC68HC05			
	Byte/Words	Cycles			Byte/Words	Cycles	
DEC	X	1	4	DECX		1	3
JFZ	BEG LOOP	1	2	BEQ	BEG LOOP	2	3
		$\frac{2}{}$	$\frac{6}{}$			$\frac{3}{}$	$\frac{6}{}$
			9.75 $\mu$ s				2.86 $\mu$ s
Z86CXX				8051			
	Byte/Words	Cycles			Byte/Words	Cycles	
DJNZ	COUNT, BEG_LOOP	2	10/12	DJNZ	Rx, BEG_LOOP	2	2
			1.67 $\mu$ s/2.0 $\mu$ s				1.2 $\mu$ s

Ejemplo IV.1.A. 2 Ciclo de control.

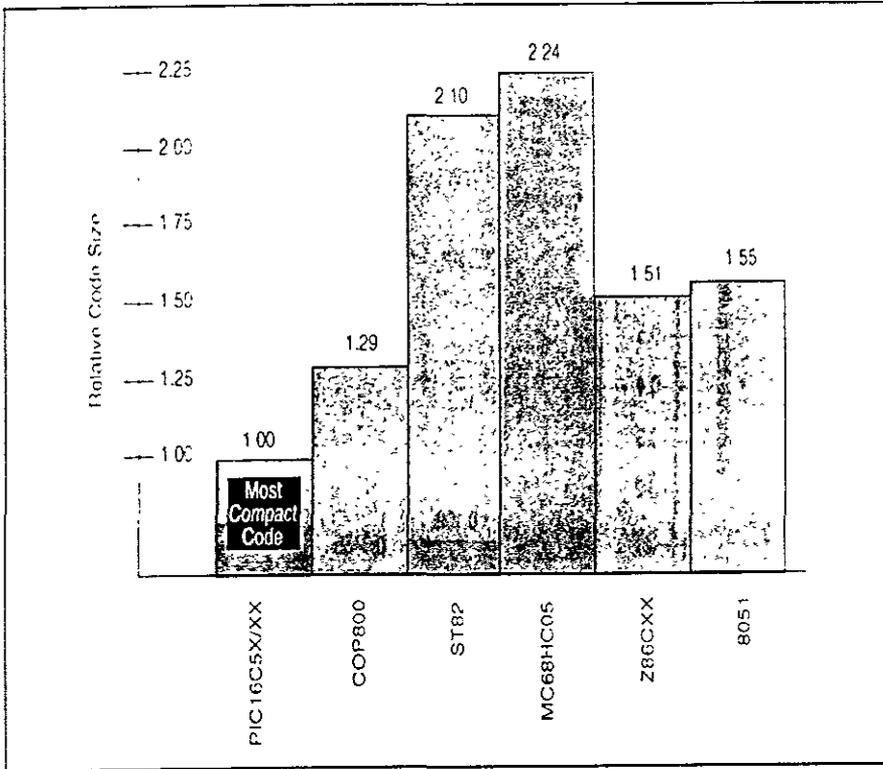
Este ejemplo prueba un bit sencillo en un registro o en una localidad de RAM y realiza una bifurcación condicionada. Nosotros asumimos que el MSB es probado y una bifurcación se realiza si el bit es puesto.

PIC16C5X/XX				COP800			
BTFSCL	REG. 7	Byte/Words	Cycles	IFBIT	7 [B]	Byte/Words	Cycles
GOTO	NEWADD	1	1/2	JP	NEWADD	1	1
		$\frac{1}{2}$	$\frac{2}{3}$			$\frac{1}{2}$	$\frac{3}{4}$
0.6 $\mu$ s/0.4 $\mu$ s				8 points to the memory location under test			
4 $\mu$ s							
ST62				MC68HC05			
JRR	7 NEWADD	Byte/Words	Cycles	BRCLR	7 NEWADD	Byte/Words	Cycles
		3	5			3	5
8.125 $\mu$ s				2.38 $\mu$ s			
Z86CXX				8051			
BTJRT	NEWADD, REG 7	Byte/Words	Cycles	MOV	A @ R <sub>x</sub>	Byte/Words	Cycles
		3	16/18	JB	A 7 NEWADD	1	1
						$\frac{3}{4}$	$\frac{2}{3}$
2.67 $\mu$ s/3.0 $\mu$ s				1.8 $\mu$ s			
				Registers R <sub>x</sub> is assumed to be pointing to the memory location under test			

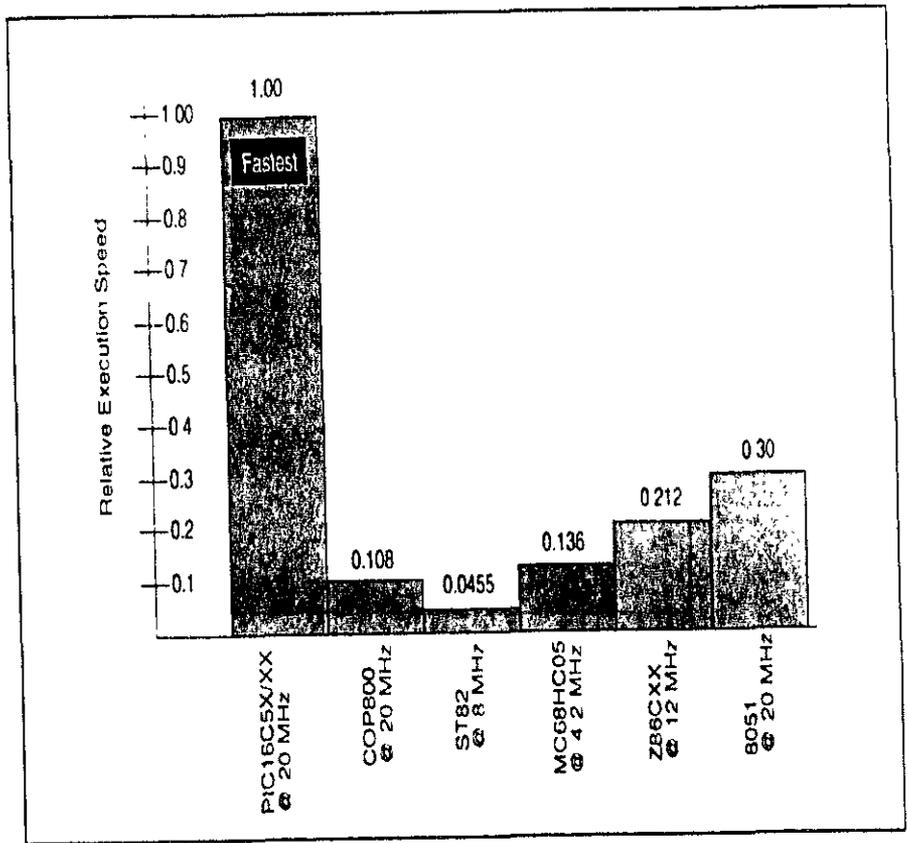
Ejemplo IV.1.A. 3 Bit de prueba y bifurcación

Device	Packing BCD	Loop Control	Bit Test & Branch	8-Bit Sync Transmission	10 ms Software Timer	Overall
COP800	4	2	2	16	8	1.29
	2.00	1.00	1.00	1.46	1.00	
ST62	10	2	3	19	10	2.10
	5.00	1.00	1.50	1.73	1.25	
MC68HC05	10	3	3	20	11	2.24
	5.00	1.50	1.50	1.82	1.38	
Z86CXX	4	2	3	21	9	1.51
	2.00	1.00	1.50	1.9	1.125	
8051	4	2	4	15	11	1.547
	2.00	1.00	2.00	1.36	1.375	
PIC16C5X/XX	2	2	2	11	8	1.00

Tabla IV.1.A. 1 Resumen de los ejemplos.



Gráfica K.1.1.A. : Ejecucion de código.



Grafica IV.1.A. 2 Ejecución relativa de velocidades.

Estos criterios se consideraron para la elección del microcontrolador, ya que se trabajó con señales de video que tienen una frecuencia aproximada de 6.25 MHz.

La frecuencia de la señal es el parámetro que se considera de mayor importancia. Las señales de frecuencias altas deben ser tratadas de una manera diferente de las de señal pequeña, ya que se tiene una influencia de elementos parásitos como las

inductancias y las capacitancias, que afectan la señal de manera considerable. También se debe reducir al máximo la señal de "ruido" o señales no deseadas, que son generadas por diferentes fuentes del medio ambiente y de las cuales desconocemos su modelo matemático.

Por estos factores, deben seguirse técnicas especiales en la fabricación de las tarjetas electrónicas. Algunas técnicas de manufactura son:

- Debe cubrirse la mayor área de "tierra" que sea posible en la tarjeta.
- No es recomendable tener pistas en el circuito de la tarjeta a 90°, sino a 45°.
- Los dispositivos electrónicos deben ser agrupados a la menor distancia posible.
- Las terminales de los componentes deben de ser lo más corta posible, pues de lo contrario, pueden realizar la función de antenas y captar señales no deseadas.
- Los componentes no deben de estar demasiado separados del circuito impreso.
- En tarjetas de doble cara o más, se debe de tener especial cuidado en que las pistas inferiores y superiores no sean paralelas, ya que esta posición produce capacitancias parasitas. Las pistas deben ser perpendiculares entre sí.
- Los dispositivos a polarizar deben estar cercanos a las fuentes de alimentación.
- Los elementos pasivos deben ser de especificaciones especiales para trabajar a frecuencias altas (elementos militares de preferencia ya que se fabrican con

técnicas especiales que minimizan la generación de elementos parásitos.

La selección de los dispositivos electrónicos se debe realizar en función de sus especificaciones de fabricación, según el uso específico que se le dará en el sistema.

Actualmente existe en el mercado una gran cantidad de convertidores analógicos a digitales. El criterio de selección fue de nuevo la frecuencia de operación del dispositivo electrónico, el costo y el tiempo de conversión de la señal de analógica a digital, ya que se debe de tomar en cuenta el criterio de Nyquist sobre el teorema de muestreo para elegir correctamente la frecuencia de operación del convertidor. El ADS901E es un convertidor de alta velocidad de muestreo, relativamente más económico y más sencillo de manipular que otros convertidores de su género. Este convertidor tiene la característica de manejar salidas de niveles lógicos de voltaje TTL. Además, en las aplicaciones que recomienda el fabricante se puede encontrar la opción para señales de video entre otras.

En un diseño anterior al que se obtuvo finalmente, el microcontrolador proporcionaba el bus de dirección de datos de la memoria, la habilitación de los modos de escritura/lectura, e indicaba las condiciones de inicio de la memoria, así como la transferencia de datos de forma serial al dispositivo que los procesaría y manipularía una vez ya procesados los datos. El problema principal que se presentó fue el siguiente: al realizar el cálculo de los ciclos de ejecución por instrucción, se obtuvo un muestreo de 1.03 MHz, lo que no cumple con el teorema del muestreo de Nyquist. La señal que se generó fue realmente ruido.

Esto lo observamos al introducir una señal sinusoidal de frecuencia y amplitud conocida:

Frecuencia 500 kHz.

Amplitud 1 Vpp.

Como la frecuencia de muestreo se encontraba por debajo de la señal a digitalizar, realmente no se obtuvo su digitalización.

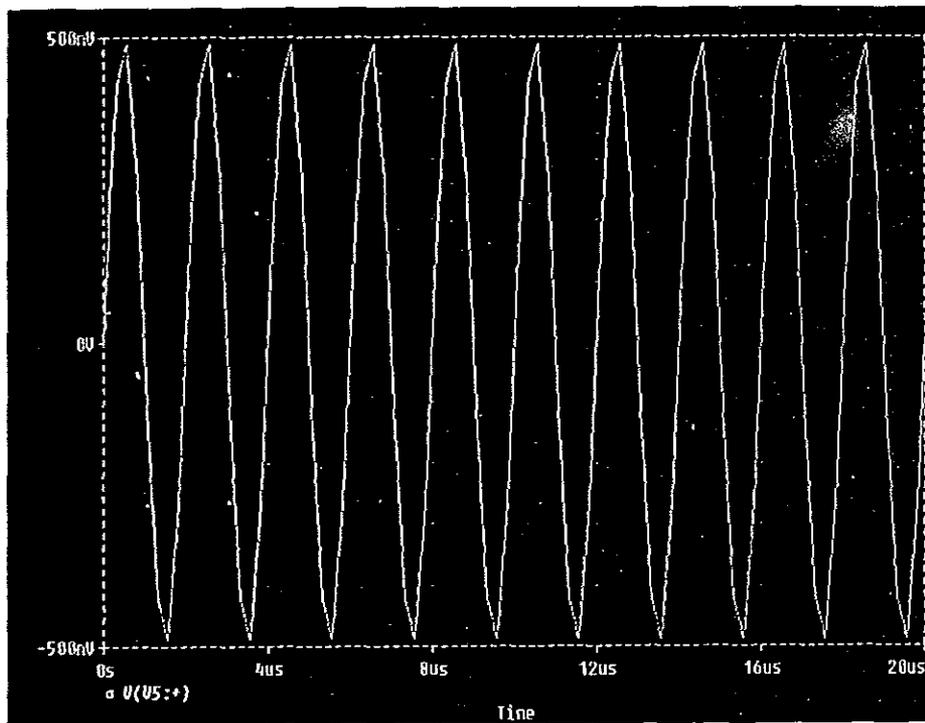


Figura IV.1.A. 1 Señal de entrada.

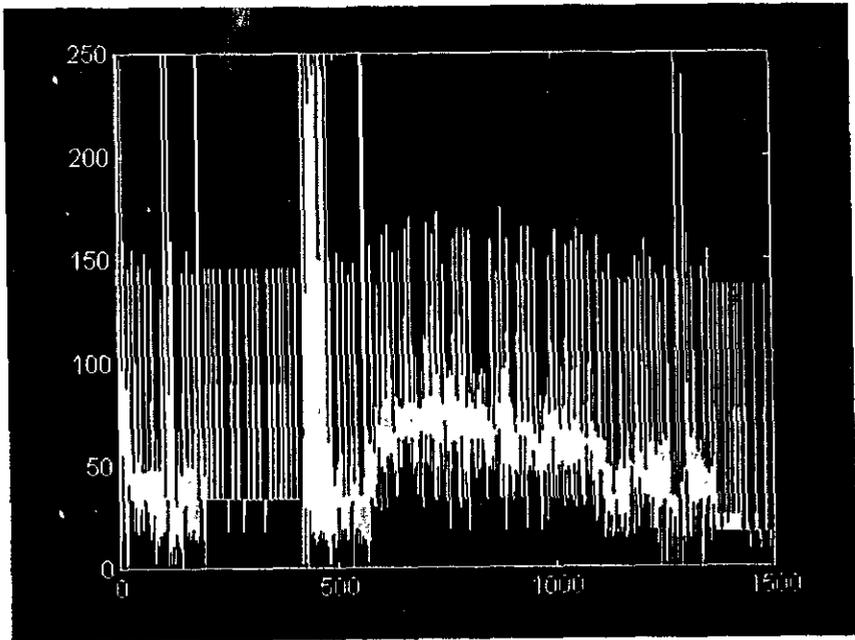


Figura IV.1.A. 2 Señal de salida.

Al observar la señal digitalizada, comprobamos que no se obtuvo nada que nos permitiera trabajar con el software para interpretar la señal de video digitalizada, de hecho nunca se obtuvo tal digitalización. Una forma de resolver este problema era proporcionar a la memoria un direccionamiento rápido y eficaz sin que afectara a los demás dispositivos. Se llegó a la conclusión de que una mejor manera de proporcionar las direcciones de las localidades de la memoria era por medio de contadores de alta velocidad. Con esto, se logró que realmente se realizara el muestreo a la frecuencia de reloj del convertidor. El oscilador que sirve para darle la señal de reloj al convertidor es de una frecuencia de 20 MHz, por

lo que a esta frecuencia se puede asegurar un muestreo de por lo menos 3 veces la frecuencia de la señal a digitalizar.

El banco de contadores nos garantiza una rápida transferencia de datos, ya sea del ciclo de lectura o de escritura de la memoria.

Una vez que el programa del microcontrolador detecta que la memoria esta completamente llena, por medio del puerto del USART comienza a transmitir de forma serial en el formato EIA - RS 232 hacia el equipo de recepcion que va a usar esta informacion -en este caso se trata de una PC. Una de las ventajas que posee el PIC, es que se puede configurar la velocidad de transmision en un amplio rango, dependiendo de la señal de reloj que esté manejando el microcontrolador, como se mencionó en la descripción del PIC.

Como se dijo anteriormente, es necesario quitar el pulso de sincronia horizontal, ya que para el software es mas complicado detectar el inicio y el final de la señal de video.

De acuerdo a las especificaciones del convertidor AD3 901E, si no le proporcionamos una señal de reloj, no puede comenzar la digitalizacion de la señal. Teniendo en cuenta esta característica, se propuso que mediante un circuito comparador con histeresis, detectáramos cuando cruza por cero la señal. El pulso de sincronia se puede manipular por medio de un potenciómetro exterior que le proporciona una señal de offset, cuya funcion en este caso es manipular el nivel de referencia de la señal de video y poner al pulso de sincronia por un nivel menor que cero. Esto es de gran utilidad, ya que

al pasar por cero la señal de video, el circuito comparador da a la salida un pulso de duración aproximadamente igual al tiempo que dura la señal de video hasta que tiene una caída que es físicamente el pulso de sincronía de la señal, pero en este caso como el pulso de sincronía se encuentra en un nivel menor que cero, no da la habilitación a la señal de reloj del convertidor, por lo que en ese instante el software detecta una cadena de caracteres constantes, lo que a nosotros nos permite detectar el inicio y final de la señal digitalizada, empleando un transistor que se encuentra en corte o en saturación, y que se activa por el pulso que recibe del circuito comparador esta es una manera de proteger al convertidor contra posibles niveles de voltajes que no sean TTL, al asegurar la saturación del transistor a 5 Volts. Cuando el transistor se pone en saturación envía un pulso, activando la señal de reloj del convertidor. De esta manera es como controlamos la digitalización de la señal.

Otro punto crítico que se debe tomar en cuenta en la caracterización de los colores es considerar una fuente externa de iluminación constante. Esto nos garantizará que los datos que se obtengan de la tarjeta de adquisición de datos no sean afectados por las características de iluminación del medio ambiente.

Finalmente, después de realizar la digitalización de la señal analógica a digital por medio de la tarjeta de adquisición de datos, se procede a la transmisión. Ésta se efectúa por medio del USART contenido en el microcontrolador, pero no se puede conectar directamente a la PC. Por ello, necesitamos una interface que nos proporcione los niveles de RS 232 en la

transmisión del microcontrolador y, a la vez, los niveles lógicos de recepción en el PIC. Afortunadamente, existen dispositivos electrónicos dedicados a estas funciones. Uno de los más populares y fáciles de manejar es el MAX 232, que es la parte con la que concluimos el diseño de la Tarjeta de Adquisición de Datos.

#### IV.1.B. INTERPRETACIÓN DE LOS DATOS A TRAVÉS DEL SOFTWARE.

La cámara digital va generando datos como se muestra en la Figura IV.1.A.4, con el programa LEER.C éste lee del puerto serial la información que envía el PIC.

Cada color tiene su curva característica, la idea es almacenar los datos de dicha curva, cada una es de aproximadamente 1,000 datos.

Este programa configura el puerto serial en modo de interrupción para no perder datos, y envía la información a un archivo binario.

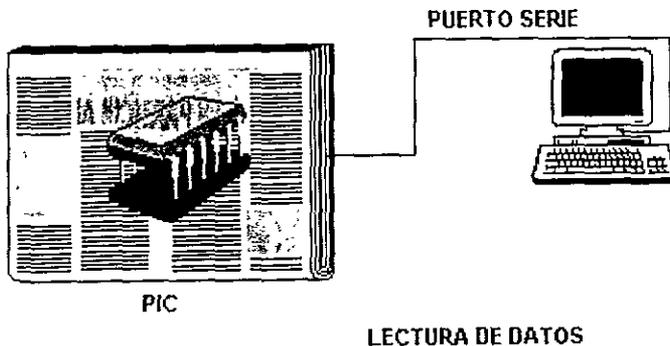


Figura IV.1.A. 3 LECTURA DE DATOS

Asimismo le debemos pasar como parámetro el número de datos a recibir ya que cada curva es de aproximadamente 1,000 datos, cuando se desea caracterizar un color se toman 24,000 datos para asegurar al menos 20 curvas y cuando únicamente se muestrea se toman 12,000 datos para asegurar tener 10 curvas.

A continuación se listan los procedimientos que determinan el proceso de caracterización del color y la designación de que color se esta muestreando.

Para delimitar cada curva se tienen pulsos constantes al inicio y al final, ver Figura IV.1.A.5

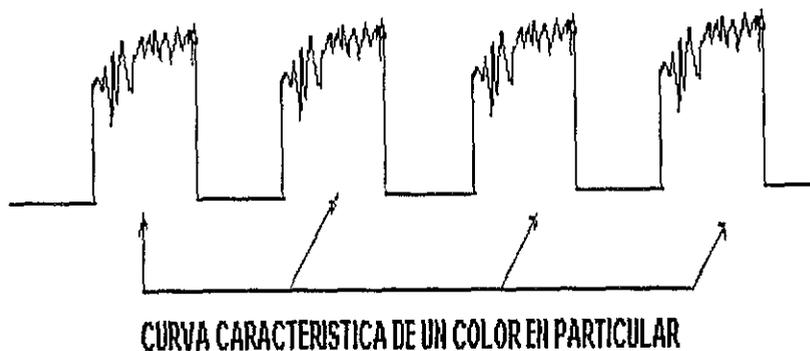
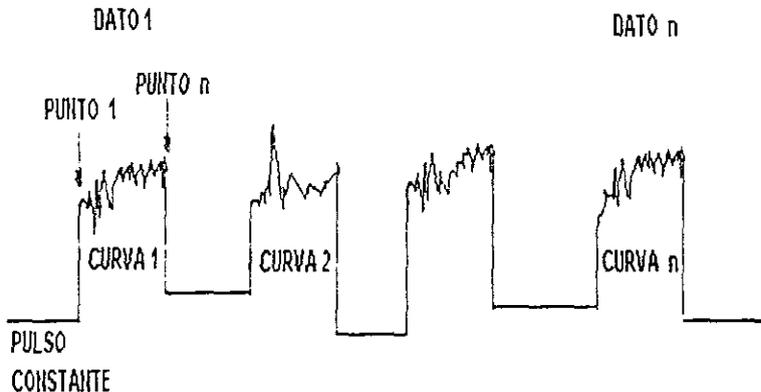


Figura IV.1.A. 4 Datos que envía la cámara de video



EJEMPLO DE PULSOS QUE ENVIA EL PIC

Figura IV.1.1.A. b). Señal de video a través del PIC16C44

#### IV.1.B.1. CARACTERIZACION.

El objetivo es obtener la curva característica para cada color y almacenar sus datos, para que posteriormente se pueda comparar con la muestra que se tome. Se tienen los programas: `CARACTERIZAR_COLOR.HTML` y `CARACTERIZAR.JS`, estos dos programas se compilan con Netscape Server para generar el programa ejecutable llamado `CARACTERIZA.WEB`

Se da el nombre del color a caracterizar, se obtiene su componente RGB con la página `wheel.html`.

NOTA. Para tener el color en RGB, se utiliza un programa llamado `COLOR_WHEEL`, lenguaje JavaScript y HTML, el cual muestra un círculo con diferentes tonos, y la intensidad de cada color uno lo selecciona hasta que nos da el tono que queremos.

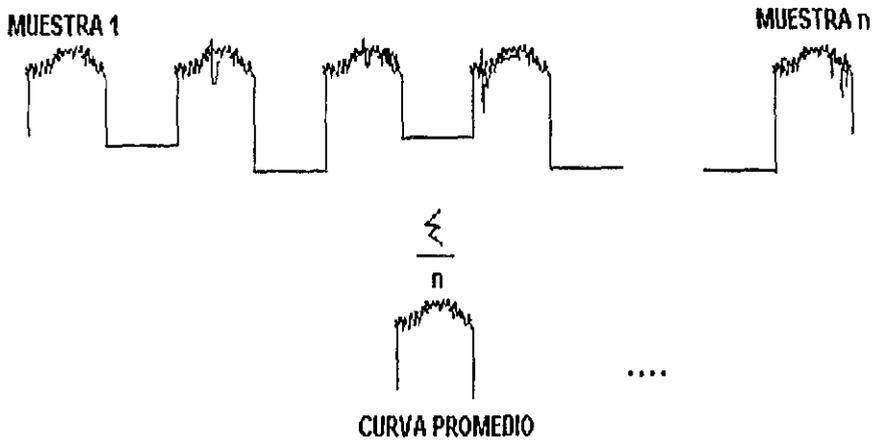


Figura IV.1.A. 6 Curva promedio

Se toman 20 curvas y se eliminan los picos para poder obtener así la curva promedio. En una tabla cabecera de datos llamada COLORES, se guarda nombre del color, código RGB y una llave primaria artificial llamada ID, en otra tabla llamada CURVA se almacena propiamente los datos de la curva y contiene la llave ID, y valores de cada punto.

ID	MOMBRE_COLOR	CODIGO_RGB
1	ROJO CARMESI	FF0005

TABLA IV.1.A. 2 TABLA COLORES

	ID	INDICE	VALOR
	1	1	60
	1	2	63
	1	3	68
	1	4	69
	1	5	64
	1	6	62

TABLA IV.1.1.A.1. TABLA CURVA

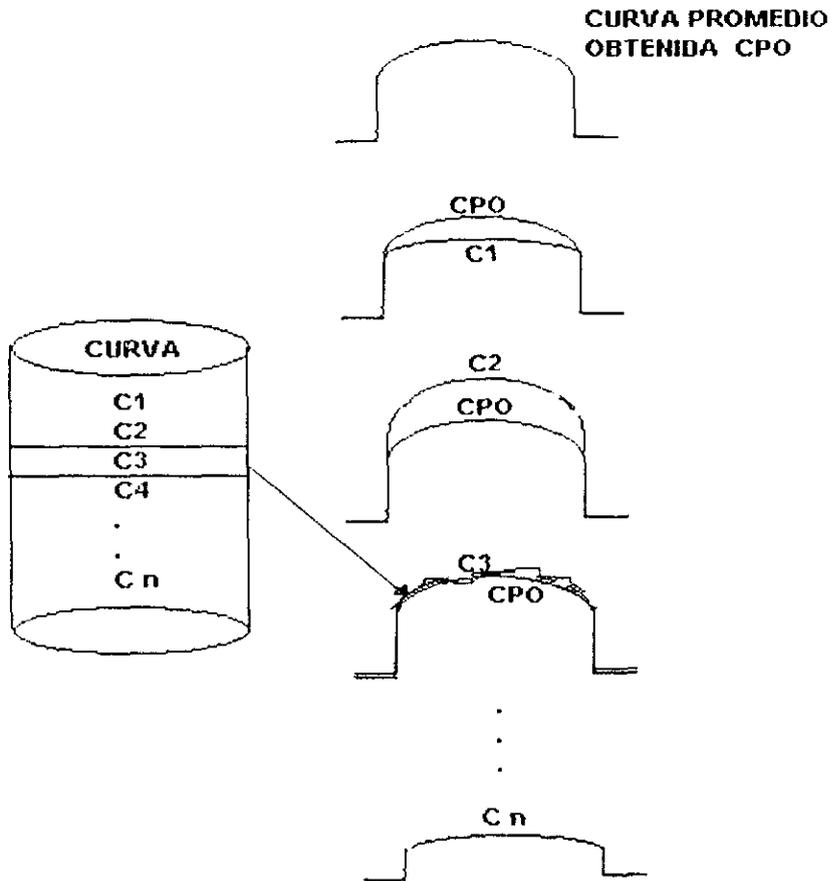
#### IV.1.B.2. MUESTREO.

Al igual que en la caracterización, se realiza el mismo proceso, solo que aquí se toman 12,000 datos.

El programa se llama MUESTRA.HTML, y CARACTERIDA.JS (lenguaje JavaScript, HTML y SQL, estos dos programas se compilan bajo Netscape Server para generar el programa ejecutable llamado MUESTRA.WE).

Se obtiene una curva promedio y se va comparando con cada una de las curvas almacenadas en la tabla llamada CURVA, se toma con correctitud aquella donde el número de puntos que coinciden sea mayor, ver figura IV.1.A.7.

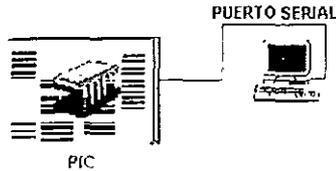
En la siguiente figura se muestra un ejemplo de una curva obtenida al muestrear un color.



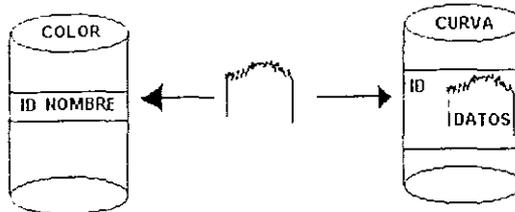
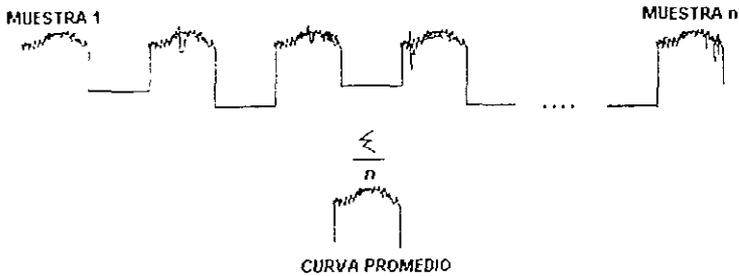
**LA CURVA PROMEDIO OBTENIDA SE COMPARA CON LA INFORMACION ALMACENADA EN LA TABLA CURVA, LA CUAL CONTIENE LAS CURVAS CARACTERIZADAS DE CADA COLOR**

Figura IV.1.7. - Analisis de los datos obtenidos.

CARACTERIZACION DE UN COLOR  
 INFORMACION QUE ENVIA LA CAMARA DE VIDEO VISTA DESDE UN OSCILOSCOPIO



LA SEÑAL DEL COLOR A MUESTREAR QUE LE LLEGA AL PIC SE ENVIA A LA COMPUTADORA



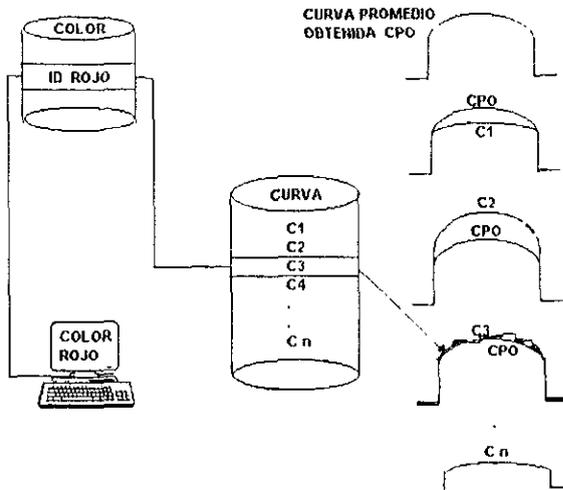
LOS DATOS DE LA CURVA PROMEDIO SE ALMACENAN EN UNA TABLA LLAMADA CURVA, Y EL ENCABEZADO QUE CORRESPONDE AL NOMBRE DEL COLOR Y SU CODIGO RGB SE ALMACENA EN LA TABLA LLAMADA COLOR.

Figura 1.1.1.4. Caracterización de un color

**MUESTREO DE UN COLOR**  
**INFORMACION QUE ENVIA LA CAMARA DE VIDEO VISTA DESDE UN OSCILOSCOPIO**



LA SEÑAL DEL COLOR A MUESTREAR QUE LE LLEGA AL PIC SE ENVIA A LA COMPUTADORA



SE OBTIENE UNA CURVA PROMEDIO Y SE VA COMPARANDO CON CADA UNA DE LAS CURVAS ALMACENADAS EN LA TABLA LLAMADA CURVA, Y SE TOMA COMO CORRECTA AQUELLA A DONDE EL NUMERO DE PUNTOS QUE COINCIDEN SEA MAYOR.

Figura IV.1.A. 9 Muestreo de un color

#### IV.1. CONCLUSIONES

En la actualidad es muy importante desarrollar tecnología con los elementos y herramientas que contamos en el país. Es una forma de comprometernos como profesionales con nuestro país, México. Al diseñar y construir sistemas que demanda la industria, y no solamente limitarnos a productos extranjeros que existen en el mercado a un alto costo, y limitando a la mediana y micro empresa a contar con tecnología que las haga fuertes y competitivas dentro de este mundo de comercio globalizado y con marcada tendencia neoliberal, que cada vez marcha más a los países de bajos recursos (países en vías de desarrollo). Le ahí que exista nuestro compromiso de aplicar los conocimientos de ingeniería que se han estudiado a través de la licenciatura en la facultad de Ingeniería.

Sería muy atrevido decir que todos los dispositivos empleados en este proyecto de tesis son nacionales, pero lo que sí es fruto de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México son nuevos ingenieros con conocimientos sólidos y muy bien fundamentados, que son aplicados a problemas reales, con soluciones acertadas y funcionales, a no detenernos por las adversidades que se presentan en cualquier campo de nuestro entorno.

Este proyecto de tesis "Diseño y construcción de un colorímetro" Es un diseño muy bien desarrollado que puede competir con dispositivos electrónicos de su género de propósito general y se deja abierta su aplicación, ya que la Colorimetría se emplea tanto en el campo de la investigación científica como en el de las aplicaciones industriales. La ventaja que se tiene al utilizar esta herramienta para

desarrollar satisfactoriamente las actividades de la gente dedicada a la ciencia o a la industria, es el apoyo que surge con las nuevas tecnologías para desarrollar cada vez procesos más complejos y así entender los enigmas de la ciencia y las necesidades de la población.

Estamos concientes de las necesidades del país, que cada vez demanda gente que emprenda proyectos en nuevas tecnologías y que sean aplicadas al apoyo de la industria y de la población.

Este proyecto es una muestra de lo que se puede lograr en la UNAM, con sus estudiantes y profesores.

El empeño y constancia de los alumnos, apoyados por la disposición, experiencia y conocimientos de los profesores de la Facultad de Ingeniería, promueven la elaboración de trabajos y proyectos que apoyen a la industria y a la ciencia, para dar soluciones acertadas, reales y concretas, a problemas complejos.

La ciencia y la tecnología se desarrollan, y muy bien es: las instituciones de investigación como es la UNAM.

El éxito de cualquier empresa es muy difícil lograrlo, y más si se trata de realizar de manera individual, por lo que al trabajar en grupo se pueden aportar un mayor número de ideas y tener diversos caminos para obtener la solución de los problemas que se presenten.

Para realizar proyectos de investigación se debe de tener la disposición y el empeño de lograr el objetivo planteado. Ya

que en muchas ocasiones, mas que la falta de recursos economicos para concretar un ideal, es mas complicado si no se cuenta con el compromiso total de la gente que se encuentra involucrada directamente en la elaboracion del proyecto.

Uno de los objetivos mas importantes de la facultad de Ingenieria, es promover y fomentar la academia, tener la inquietud de crear y lograr disenos de ingenieria. Este objetivo se cumple satisfactoriamente en este proyecto de tesis.

Realmente nos sentimos comprometidos con la sociedad y con nuestro pais. Queremos aportar todos nuestros conocimientos intelectuales y nuestra sensibilidad humana en dar nuestro mayor esfuerzo como profesionales.

## BIBLIOGRAFÍA.

- ARRL Hand Book for the radio amateur 1990, USA, 1990
- Beiser, A. Física aplicada, Schaum McGraw Hill, México, 1989.
- Belove, Schilling, Circuitos Electrónicos, Alfaomega, México 1991.
- Bustamante L. Diseño de una cámara para un sistema de percepción remota en base de un CCD. Posgrado de Ingeniería UNAM, Mexico 1998.
- Burr - Brown IC Data Book, USA, 1998.
- Coughlin, Driscoll, Circuitos Integrados lineales y amplificadores operacionales, Prentice Hall, México 1991.
- Floyd, Fundamentos de Electrónica Digital, Limusa, México, 1996.
- Harold K. Color: Origen, metodología, sistematización y aplicación, Lectura, México, 1988.
- Hayt, Kemmerly, Análisis de circuitos en Ingeniería, McGraw Hill, México 1988.
- Hebert Schildt, Turbo C, Mc Graw Hill, México 1990.
- Hitachi HM66205 Series, USA, 1990
- Graham, Thompson, Óptica, Limusa, México 1992.
- Microchip, Embedded control handbook, vol.1, USA, 1997
- \_\_\_\_\_, PIC17C4X Data Sheet, USA, 1997.
- Morris M., Lógica Digital y diseño de computadoras, Pretince Hall, México, 1988.
- Motorola Analog IC Device Data, USA, 1998
- \_\_\_\_\_, Optoelectronics Device Data, USA, 1996
- \_\_\_\_\_, HC11 Reference Manual, USA, 1991
- \_\_\_\_\_, TTL & CMOS Device, USA, 1991
- Philips Semiconductors Devices, USA, 1996
- Texas Instruments Semiconductors Devices, USA, 1992

**APENDICE A.****A. CÓMO ENTRAR AL BROWSER DE MICROSOFT INTERNET EXPLORER****A.1. ABRIENDO EL EXPLORADOR DE INTERNET EXPLORER**

Para ejecutarlo se puede dar click al icono de internet explorer como se muestra en la figura A.1, o desde windows realizar la siguiente secuencia con las ventanas que se despliegan:

Inicio

Programas

Internet Explorer

Se puede observar esta secuencia en la figura A.2. Cualquiera de estas dos formas se resumen en la figura A.3.



Figura A. 1

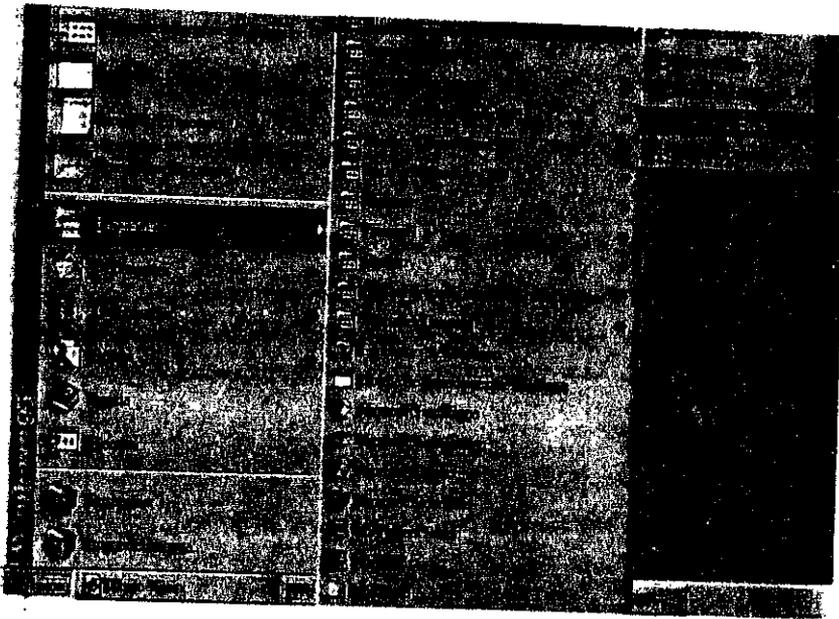


Figura A. 2

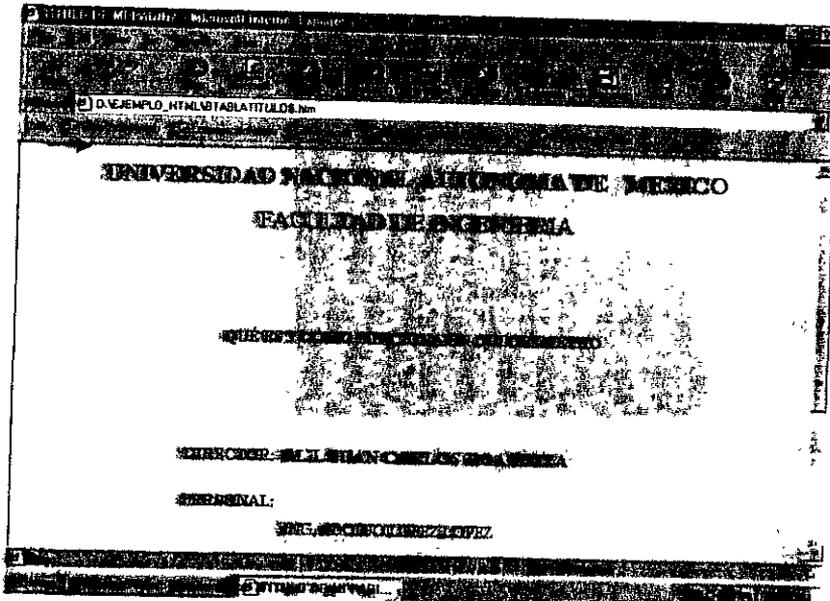


Figura A. 3

## A.2. EJECUTAR EL PROGRAMA.

En la figura A.3, en la parte superior existe una ventana en blanco etiquetada como Address, hasta este momento ya abrimos la pantalla del browser. Ahora supongamos que necesitamos ejecutar el programa BTABLATITULOS, el cual contiene código de HTML y se encuentra en la unidad D en la siguiente trayectoria D:\EJEMPLO\_HTML. Para ejecutarlo se le da toda la dirección como sigue:

D:\EJEMPLO\_HTML\btablatitulos.htm                    y enter

Al darle esta instrucción nos muestra el contenido que se ilustra en la figura A.3.

## A.3. HERRAMIENTAS PARA GENERAR CÓDIGO HTML

Se puede escribir código HTML en varios procesadores o herramientas como son:

- WordPad
- Block de Notas
- Microsoft FrontPage
- Allain: homeSite 4.1

## A.4. FRONTPAGE EDITOR

En fin, existen bastantes herramientas para poder escribir código HTML. Los dos primeros se tienen que conocer el código para poder escribir. Los dos últimos manejan ventanas, donde podemos seleccionar por ejemplo que inserte una tabla, un cuadro, una imagen, etc., y automáticamente escribe el código HTML sin necesidad de recordar su código. En la figura A.4 se muestra un ejemplo de la herramienta FrontPage

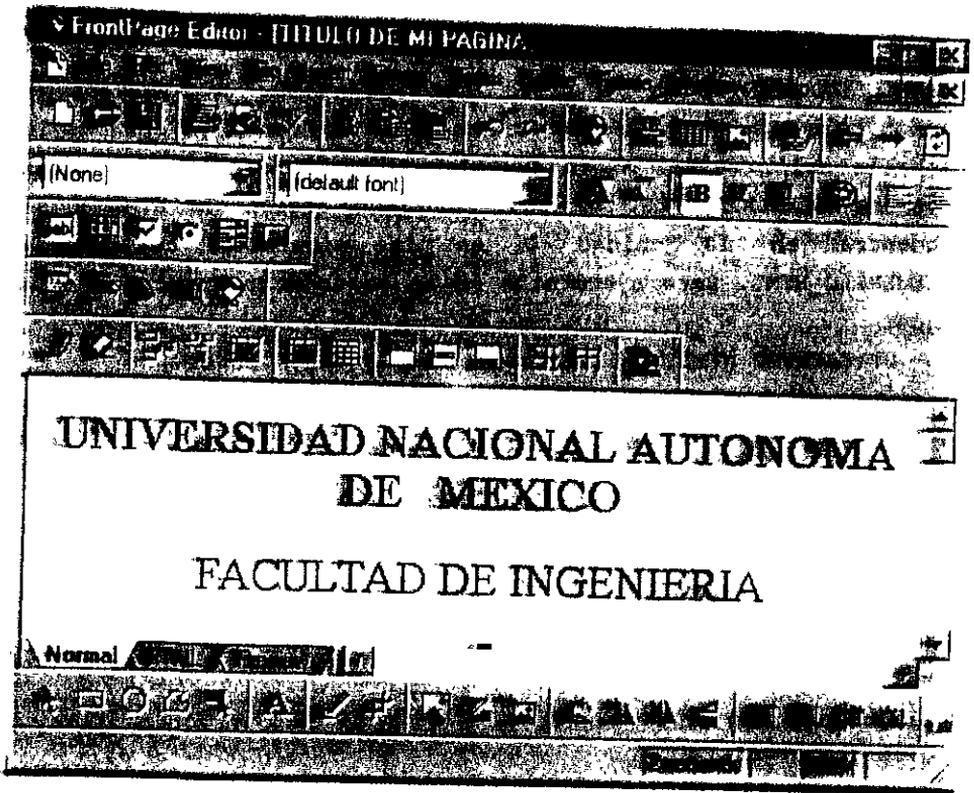


Figura A. 4

En la figura A.4 se puede observar en la pantalla inferior la pestañas que dicen Normal, HTML y Previews. En la pantalla normal se puede colocar el código HTML que se necesita. Si se necesitan ver el código HTML seleccionando la pestaña HTML se nos muestra el código, ver figura A.5.

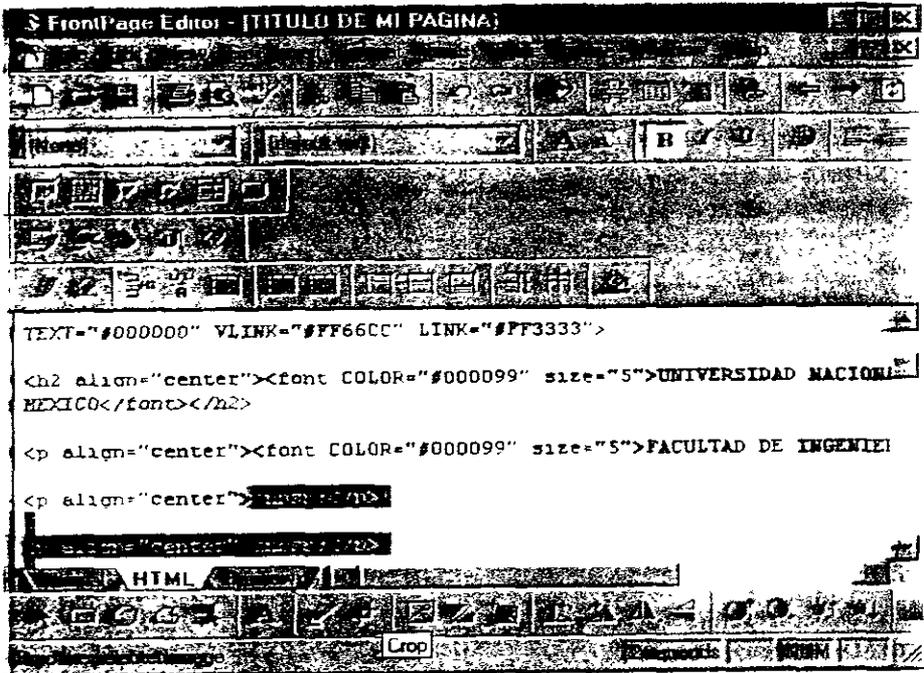


Figura A. 5

La ventana preview, nos mostrara una vista previa de como se veria al ser ejecutada en el browser de Internet Explorer, ver figura A.1

**A.5. EJECUCIÓN DE UN PROGRAMA CON CÓDIGO HTML**

**EJEMPLO A.1**

Las siguientes instrucciones son un ejemplo de:

- tres títulos de diferente tamaño
- un fondo gris que corresponde al archivo CHALK.JPG
- una tabla con dos columnas y doce renglones.



```

<td width="33%"><td>
<td width="67%"><td>
<td>
<td>
<td width="33%"><td>
<td width="67%"><td>
<td>
<td>
<td width="33%"><td>
<td width="67%"><td>
<td>
<td>
<td width="33%"><td colspan="2" style="text-align:center">SINODALES</td>
<td width="67%"><td>
<td>
<td width="33%"><td>
<td width="67%">ING. ADOLFO LOPEZ LOPEZ</td>
<td>
<td>
<td width="33%"><td>
<td width="67%">ING. PATRICIA LOPEZ LOPEZ</td>
<td>
<td>
<td width="33%"><td>
<td width="67%">ING. JUAN LOPEZ LOPEZ</td>
<td>
<td>
<td width="33%"><td>
<td width="67%">ING. JOSE LOPEZ LOPEZ</td>
<td>
<td>
<td width="33%"><td>
<td width="67%">ING. CARLA PEREZ PEREZ</td>
<td>
<td>
<td width="33%"><td>
<td width="67%"><td>
<td>
<td>
<td width="33%"><td>
<td width="67%">MAYO 1999</td>

```

```

        <td width="67%"></td>
    </tr>
</table>

<tr style="text-align:center"><td>

<p><nbsp;</p>
</td>
</tr>
</html>

```

#### EJEMPLO A.1

Otro ejemplo con marcos sería el siguiente. Suponga que se requiere una pantalla dividida en dos partes, de tal manera que el marco izquierdo muestre un menú, y cada vez que se seleccione una opción de este lo muestre en el marco derecho.

Para realizar lo anterior se necesita la instrucción **FRAME**. En la figura 6, se observa que la ventana está dividida en dos cuadros, para esto llamaremos cuadro derecho al de la derecha y así respectivamente. El cuadro izquierdo es más pequeño que el de la derecha, cada tema que está seguido de un círculo apunta a un archivo HTML, cuando lo seleccionemos dándole un click nos mostrará su contenido correspondiente en el cuadro izquierdo.



**Código del programa `frameset.html`**

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN//2.0">
<html>
<!-- Sample Template for a HTML Document created by Aardvark Pro Version 2 -->

<head>
<title>QUE ES Y COMO FUNCIONA UN COLORIMETRO</title>
</head>

<frameset cols="30%,*">
  <frame SRC="fra_left.htm" name="izquierda">
  <frame SRC="fra_right.htm" name="derecha">
  <noframes>
  <body>
  </body>
  </noframes>
</frameset>
</html>
```

Como se observa, éste hace llamado en el lado izquierdo al archivo llamado `fra_left.htm`, y en el lado derecho al archivo llamado `fra_right.htm`

**Código de `fra_left.htm`**

El siguiente código es el que se muestra en el lado izquierdo de la figura A.6.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN//2.0">
<html>
<!-- FRA_RIGHT.HTM Sample Template for a HTML Document created by Aardvark
Pro Version 2 -->

<head>
<title>Gestion de Trafico</title>
</head>

<body BACKGROUND="Chalk.jpg" BGCOLOR="#ffffff" TEXT="#000000" VLINK="#FF6600">
```





```

<table border="1" cellpadding="2" cellspacing="5" width="100%">
  <tr>
    <td width="33%"><p align="right"><strong>DIRECTOR:</strong></td>
    <td width="67%"><p align="left"><font size="4"><strong>M. I. JUAN CARLOS
ROA BELIZA</strong></font></td>
  </tr>
  <tr>
    <td width="33%"></td>
    <td width="67%"></td>
  </tr>
  <tr>
    <td width="33%"><p align="right"><strong>SINODALES:</strong></td>
    <td width="67%"></td>
  </tr>
  <tr>
    <td width="33%"></td>
    <td width="67%">ING. ADOLFO MILLAN NAJERA</td>
  </tr>
  <tr>
    <td width="33%"></td>
    <td width="67%">ING. LUCILA PATRICIA ARELLANO MENDOZA</td>
  </tr>
  <tr>
    <td width="33%"></td>
    <td width="67%">ING. JUAN LOPEZ LOPEZ</td>
  </tr>
  <tr>
    <td width="33%"></td>
    <td width="67%">ING. JOSE CARMONA PEREZ</td>
  </tr>
  <tr>
    <td width="33%"></td>
  </tr>

```

```

    <td width="67%">ING. CARLA PÉREZ PEREZ</td>
  </tr>
  <tr>
    <td width="33%"></td>
    <td width="67%"></td>
  </tr>
  <tr>
    <td width="33%">MAYO 1999</td>
    <td width="67%"></td>
  </tr>
</table>
<p align="center">&nbsp;</p>
<p>&nbsp;</p>
</body>
</html>

```

### Programa introduc.htm

```

<html>
<head>
<title>I</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1252">
<meta NAME="GENERATOR" CONTENT="Microsoft FrontPage 3.0">
</head>

<body BACKGROUND="Chalk.jpg" BGCOLOR="#ffffff" TEXT="#000000" VLINK="#FF6600"
LINK="#FF3333">
<font FACE="Courier" SIZE="4">

<p ALIGN="CENTER"><font face="Arial" size="4"
color="#000000"><strong>EL</strong></font></p>

<p ALIGN="CENTER"><font face="Arial" size="4" color="#000000"><strong>MILENIO</strong></font></p>

<p ALIGN="CENTER"><font face="Arial" size="4" color="#000000"><strong>LAS</strong></font></p>

<p ALIGN="CENTER"><font face="Arial" size="4"
color="#000000"><strong>RESPONSABILIDADES</strong></font></p>

```

<strong>DE  
</strong></p>

<strong>TECNOLOGIA</strong></p>

&nbsp;</p>

Muy lejos  
están los días en que Ray  
Bradbury y otros escritores de ciencia ficción inundaban la  
imaginación popular con descripciones apocalípticas del futuro.  
Las máquinas &quot;inteligentes&quot;, la  
robótica, la <b>tecnología</b> de la información en general,  
constituyen el pan de cada  
día y prosiguen su camino perfeccionándose  
constantemente.</p>

Como todo  
aquello que ha creado el ingenio humano, las computadoras no  
están libres de detractores. Muy por el contrario,  
hay quienes las comparan con la explosión de la revolución  
industrial que dio al traste  
con millones de empleos y provocó una crisis económica, social y  
cultural en el siglo  
XVIII.</p>

Lo cierto  
es que, bien mirado, la computación como la electrónica tienen  
también una responsabilidad social que cumplir.  
La <b>educación</b>, la <b>comunicación</b> y la difusión de  
<b>nuevas ideas</b> deben  
ser compromiso intrínsecos del avance tecnológico.</p>

```
<p ALIGN="JUSTIFY"><font face="Arial" color="#0000FF">Con la
programacion de Internet y
World Wide Web, la globalización de la información y las
posibilidades prácticamente
ilimitadas que ofrecen las redes hoy en día, la tecnología de la
información ha
contribuido como en ninguna otra época al desarrollo de todos los
ambitos, desde la
medicina hasta los negocios, pasando por el entretenimiento y el
contacto humano.</font></p>
<p ALIGN="JUSTIFY"><font face="Arial" color="#0000FF">Como
ingenieros de aplicaciones de
punta en la tecnología latinoamericana tenemos, de igual manera,
la obligación de seguir de cerca los progresos
tecnológicos.</font></p>
<font SIZE="4"><b>
<p>&nbsp;</p>
<a HREF="fra_RIGHT.htm">
<p align="center"><img SRC="goback.gif" align="middle"
ALT="Regresa al Índice" ismap
width="44" height="43"> </a></p>
</b></font>
</body>
</html>
```

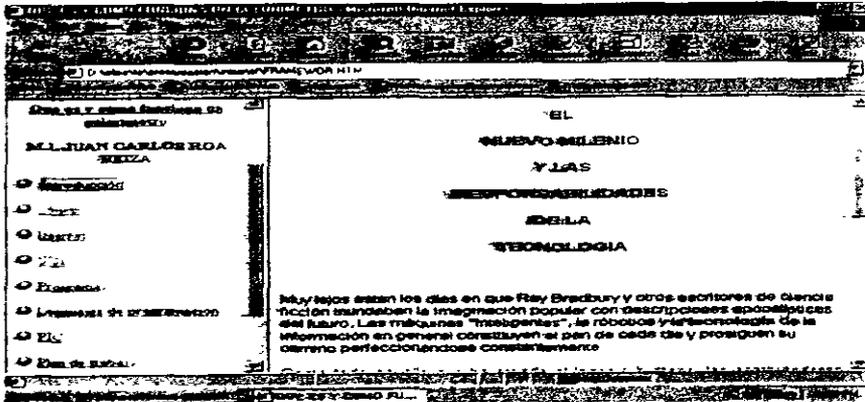


Figura A. 7

La figura A.7 nos muestra el contenido del programa `introduc.htm`, el cual corresponde al menú del cuadro izquierdo que está etiquetado como `Introducción`.

Programa `Internet.htm` (al ser ejecutado)

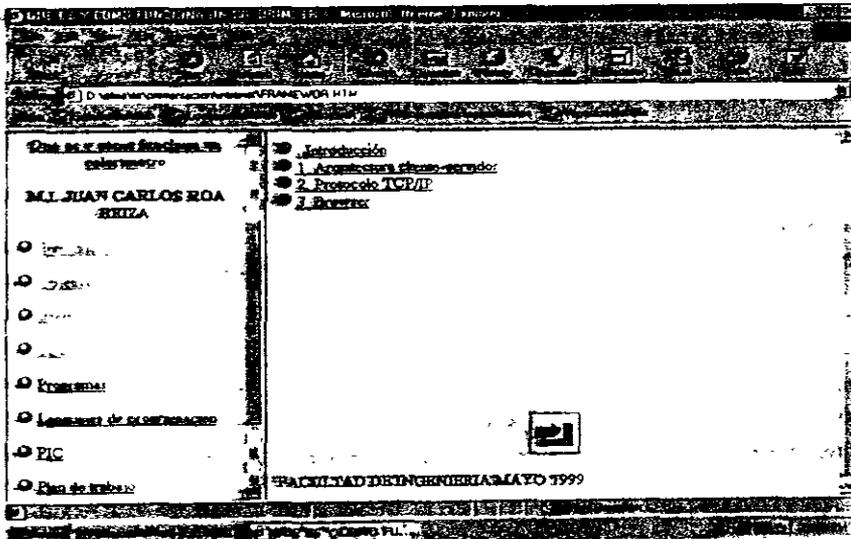


Figura A. 8

La figura A.8 nos muestra el contenido del programa internet.html, el cual corresponde al menú del cuadro izquierdo que esta etiquetado como Internet.

Código de internet.htm.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN//2.0">
<html>
  <!-- Sample Template for a HTML Document created by Aardvark Pro Version 1.0 -->
  <head>
    <title>eSPECIFI</title>
  </head>
  <body BACKGROUND="Chalk.jpg" BGCOLOR="#ffffff" TEXT="#000000" VLINK="#FF6600"
LINK="#FF3333">

  <p align="center"><img SRC="http://www.ingenieria.com/imagenes/leimex.gif"
  ALT="Teime">
  width="121" height="62"> </p>
  <h4 align="center"><u><font COLOR="#0333FF">PROGRAMACION</font></u></h4>
  <p><img SRC="99-00-20.gif" ALT="punto rojo" width="14" height="14"> <a
  href="internet_introd.html" target="derecha"><font color="#0000FF">
  Introducción</font></a><br>
  <img SRC="99-00-20.gif" ALT="punto rojo" width="14" height="14"> <a
  href="cliente-servidor.html" target="derecha"><font color="#0000FF">1. Arquitectur
  cliente-servidor </font></a><br>
  <img SRC="99-00-20.gif" ALT="punto rojo" width="14" height="14"> <a
  href="protocolotop.html" target="derecha"><font color="#0000FF">2. Protoco
  TCP/IP </font></a><br>
  <img SRC="99-00-20.gif" ALT="punto rojo" width="14" height="14"> <a
  href="browser.htm"
  target="derecha"><font color="#0000FF">3. Browser </font></a><br>
  <br>
  <br>
  </p>
  <p>&nbsp;</p>
  <a href="internet.htm" target="derecha">
  <p align="center"><img SRC="goback.gif" align="middle" ALT="Regresa al Inicio"
  image
  width="44" height="43"> </a></p>
  <p>FACULTAD DE INGENIERIA MAYO 1999</p>
  </body>
```



La figura A.9 nos muestra el contenido del programa Internet\_introd.html, el cual corresponde al menú de la figura A.8 del cuadro derecho etiquetado como Introduccion.

En la figura A.9 se muestra un dibujo que es una llamada dentro del programa Internet\_introd.html. El archivo que corresponde al dibujo se llama INTERNETSERV.GIF

#### Código de Internet\_introd.html

```
<html>
<head>
<title>I</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1252">
<meta NAME="GENERATOR" .CONTENT="Microsoft FrontPage 3.0">
</head>
<body BACKGROUND="Chalk.jpg" BGCOLOR="#ffffff" TEXT="#000000" VLINK="#FF66CC"
LINK="#FF3333">
<b><font SIZE="4">
<p ALIGN="JUSTIFY"><font color="#0000FF">INTERNET.</font></p>
</font></b>
<p ALIGN="JUSTIFY">Internet es la base de datos más grande del mundo con una
característica adicional: casi toda la información es gratuita, lista para ser
consultada por sus cientos de miles de usuarios.</p>
<p ALIGN="JUSTIFY">Sus principales servicios y/o herramientas cada día
aumentan y se vuelven más amigables para el usuario. Así como el <b>comercio
electrónico</b> el cual va a ser la vanguardia para las empresas de aquí en
adelante.</p>
<p ALIGN="JUSTIFY">Esta red se basa en la configuración <b>cliente-
servidor</b>. Por ejemplo, cada uno de los equipos de cómputo que forman parte
de Internet recibe el nombre de computadora host (huésped). Algunos host
hospedan contenidos o las aplicaciones , que sirven a otras computadoras , por
lo que se les denomina servidores. Otras computadoras, como la de la gran
mayoría de los usuarios de Internet, consumen o utilizan el contenido y la
información ofrecida por los servidores. A estos equipos se les conoce como
computadoras cliente. En conjunto, esta relación se denomina arquitectura
cliente-servidor.</p>
<p ALIGN="JUSTIFY">Un <b>programa cliente</b> es aquel que el usuario ejecuta
en su
```

computadora para acceder a recursos de Internet. Cuando el programa cliente necesita a.c.

de Internet, busca un programa servidor.</p>

```
<b ALIGN="JUSTIFY">Un <b>programa servidor</b> recibe la petición del programa cliente y
```

```
envía de regreso una respuesta. Por ejemplo, su navegador Web es un programa cliente ,
```

```
cada servidor Web de Internet es un programa servidor.</p>
```

```
<p ALIGN="JUSTIFY">Para que los usuarios finales puedan navegar en Internet, se necesitan <b>lenguajes de programación</b> como JavaScript, Java, Corba, HTML etc., y manejadores de base de datos relacional como SQL, así como protocolos de comunicación como TCP/IP etc.</p>
```

```
< < ALIGN="JUSTIFY">&nbsp;</p>
```

```
<b>&nbsp;</p>
```

```
<b></b>
```

```
<p>&nbsp;</p>
```

```
<a href="internet.htm" target="derecha">
```

```
<p align="center"><img SRC="goback.gif" align="middle" ALT="Regresa al Índice" ismap
```

```
width="44" height="43"> </a></b>
```

```
</body>
```

```
</html>
```

El dibujo (figura A.10) se llama goback.gif, se encuentra al final de todos los cuadros del lado derecho, cuando se le da click, nos regresa al menú inicial de donde se hizo el llamado. Ejemplo, en la figura A.9 cuando se le da click a este dibujo (goback.gif), nos regresa al menú de la figura 8 y toda la información se muestra dentro del cuadro derecho.



Figura A. 10 Archivo goback.gif





Programa protocoloTCP.html al ser ejecutado.

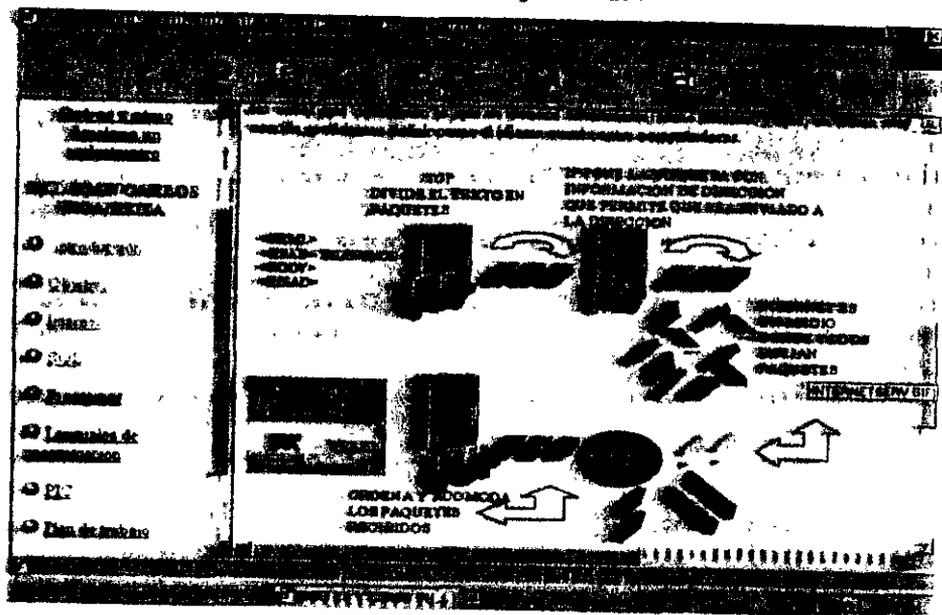


Figura A. 12

La figura A.12 nos muestra el contenido del programa protocoloTCP.html, el cual corresponde al menu de la figura A.8 del cuadro derecho etiquetado como 2.Protocolo TCP/IP.

Código protocoloTCP.html

```

<html>
<head>
<title>I</title>
<meta HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1252">
<meta NAME="GENERATOR" CONTENT="Microsoft FrontPage 3.0">
</head>
<body BACKGROUND="Chalk.jpg" BGCOLOR="#ffffff" TEXT="#000000" VLINK="#FF6600"
LINK="#FF3333">
<div>
<p ALIGN="JUSTIFY"><font color="#0000FF">PROTOCOLO TCP / IP</font></p>
<p ALIGN="JUSTIFY">(Transmission Control Protocol/ Internet Protocol)</p>
<p ALIGN="JUSTIFY">&nbsp;</p>

```





**APÉNDICE B****B.1. MANUAL DE OPERACIÓN DEL COLORIMETRO.**

**A CONTINUACION SE MUESTRAN LOS PASOS PARA CARACTERIZAR Y MUESTREAR UN COLOR:**

La información para caracterizar y muestrear a los colores se encuentra en un servidor al cual le llamamos ZEUS bajo la plataforma NETSCAPE COMMUNICATOR V4.0 y NETSCAPE ENTERPRISE SERVER V3.0.

**CARACTERIZACION:**

- 1.- Primero se debe tener un objeto a caracterizar.
- 2.- Ejecutar el programa en lenguaje C para caracterizar, este se encuentra como un icono llamado LEE\_COLOR.BAT, ver figura uno.



Figura B. 1

Al terminarse de ejecutar este programa se abre una pantalla de MSDOS, y se tiene que cerrar dando click en la esquina superior derecha en el signo de X.

- 3.- Entrar al Navegador de Netscape Server, esto se realiza seleccionando la trayectoria Inicio, Programas, Netscape

Communicator Professional Edition, Netscape Navigator, como se muestra en la figura B.2.

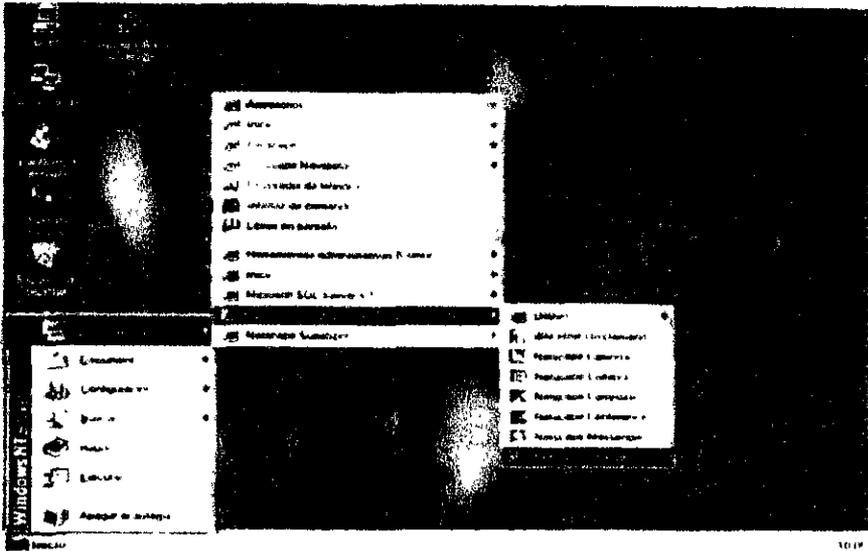


Figura B. 2

Después de ejecutar el **Navegador de Netscape** se muestra la figura B.3.



Figura B. 3

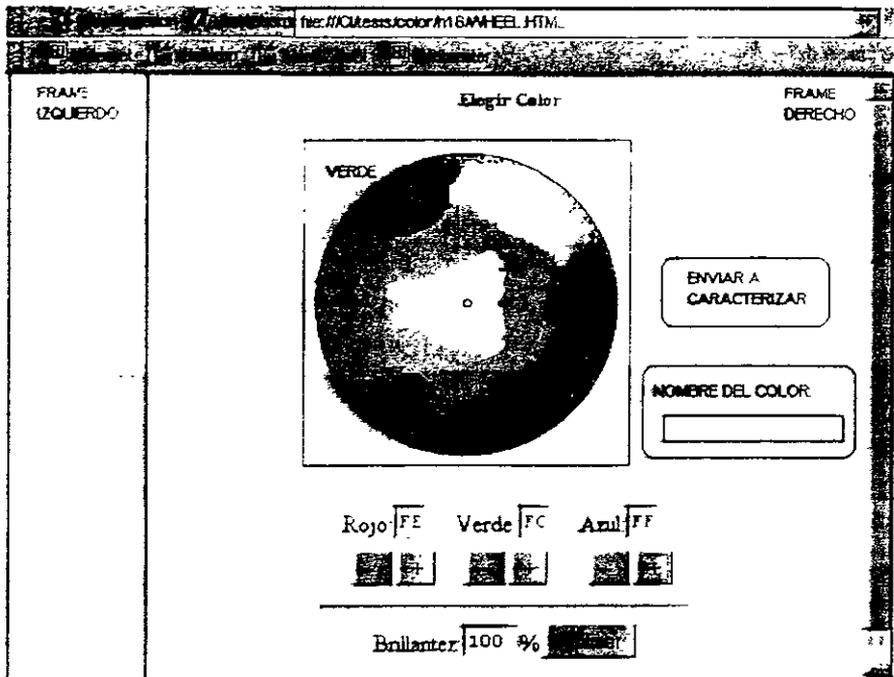


Figura B. 4

Ejecutar en el Navegador de Netscape el programa WHEEL.HTML, este se encuentra en la trayectoria <http://zeus.com> como se muestra en la figura B.4

La figura B.4 esta dividida en dos partes, el lado izquierdo y lado derecho. En el lado derecho vamos a seleccionar los componentes RGB (Red, Green y Blue) del color a muestrear. Por ejemplo, si tuvieramos un color rojo, seleccionamos con el "mouse" un punto de la zona del color rojo, y en la pantalla del lado izquierdo se ilumina de color rojo, si lo deseamos más intenso el color o más claro, vamos seleccionando con los signos de + y - en los cuadros etiquetados como rojo, verde y azul así como el brillo de los colores, le damos click al botón llamado

APLICAR, y en el lado izquierdo nos da el tono de color que escogimos.

El campo que esta etiquetado como NOMBRE DEL COLOR, coloca el nombre de éste, ejemplo ROJO BANDERA, ROJO CARMESI etc.

En el lado derecho tenemos un botón que dice ENVIAR A CARACTERIZAR, damos click a éste botón y nos muestra la figura B.5.

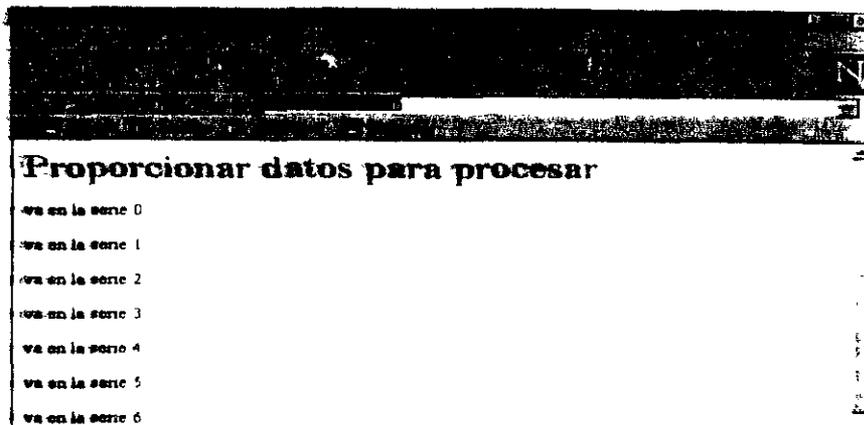


Figura B. 5

El programa de la figura B.5 (CARACTERIZA.WEB) va a caracterizar el color, el cual obtiene una curva característica promedio y la almacena en la tabla CURVA. Va desplegando en que numero de curva se encuentra. Al finalizar se encuentra un hipertexto que dice GRAFICAR LA CURVA PROMEDIO DEL COLOR (figura B.6), esto sirve para mostrar los datos en forma gráfica. Si seleccionamos esta opción nos muestra una gráfica, ver figura B.7.

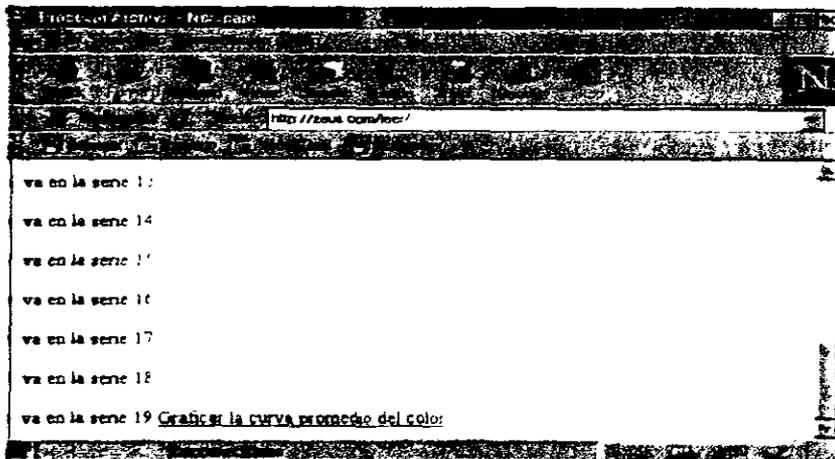


Figura B. 6

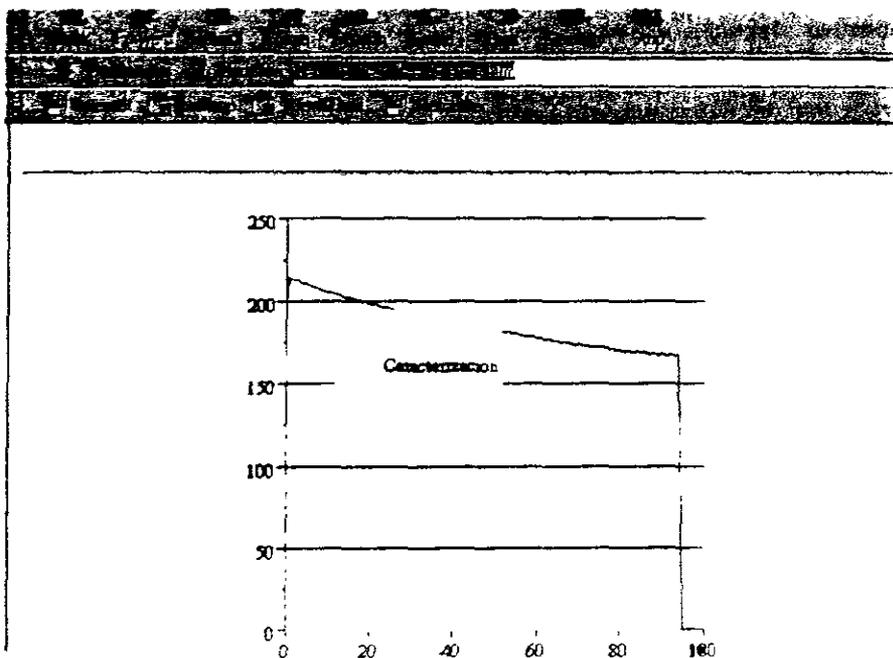


Figura B. 7

**B.2. MUESTREO.**

1. Primero se debe tener un objeto a muestrear.
2. Ejecutar el programa en lenguaje C para muestrear, este se encuentra como un icono llamado LEE\_MUESTRA.BAT, ver figura ocho.



Figura B. 8

Al terminarse de ejecutar este programa se abre una pantalla de MSDOS, y se tiene que cerrar dando click en la esquina superior derecha en el signo de X.

3. Para saber que color estamos muestreando ejecutamos bajo el navegador de Netscape el programa MUESTRA.WEB que se encuentra en la trayectoria <http://zeus.com/muestra/>, como se indica en la figura B.9.



Figura B. 9

El valor que se aproxima al valor muestreado es desplegado en la pantalla, así como su código en RGB.

Si deseamos graficar los datos del color muestreado, con los valores de la curva almacenados, se da click en el hipertexto que dice GRAFICAR, este se muestra en la figura B.10.

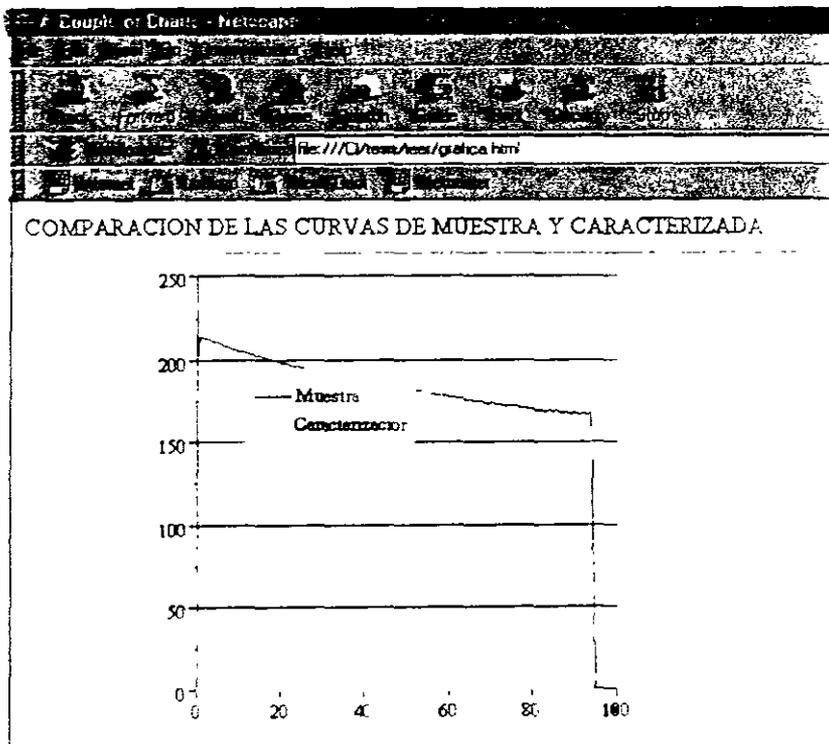


Figura B. 10

## APENDICE C

### C.1. FUENTES DE VOLTAJE

La mayoría de los dispositivos electrónicos requieren voltajes de CC para operar. Son útiles las baterías en dispositivos de baja potencia o portátiles, pero el tiempo de operación es limitado a menos que se reemplacen o se recarguen las baterías. Las fuentes de alimentación disponibles que se tienen comúnmente son las de CA de 60 hertz a  $110 \pm 10\%$ . Pero este tipo de alimentación de nada sirve si no está rectificadas, esto es si no obtenemos puramente la señal de DC.

Afortunadamente existen técnicas y dispositivos electrónicos que nos permiten la rectificación y regulación de voltaje de CA.

Comúnmente se tiene una fuente de alimentación de voltaje de CA a 60 hertz,  $110 \text{ RMS} \pm 10\%$  Volts. Lo que necesitamos es realizar una reducción de este voltaje a un nivel de amplitud que nosotros lo podamos manejar de acuerdo a nuestras necesidades. Para realizar la reducción de voltaje generalmente ocupamos un transformador, al utilizar el transformador una de las ventajas que nosotros tenemos es que el sistema se encuentra protegido contra posibles cortos de alimentación de la fuente de CA.

El valor pico del voltaje se obtiene de la siguiente manera

$$V_m = \sqrt{2} \text{ V RMS} \dots \dots \dots \text{Ec. C. 1}$$

Por ejemplo si tenemos un voltaje de 12 V RMS de AC su valor de voltaje pico sería:

$$V_m = 1.4 \times 12 = 16.8 \text{ Volts}$$

Una vez que se obtiene la reducción de voltaje de AC, lo que necesitamos es rectificar la señal. La rectificación la realizamos por medio de un puente rectificador de onda completa. Pero el voltaje de AC no es puro ya que presenta unos pulsos a la salida, para eliminar estos pulsos se colocan entre las terminales de salida un capacitor.

Para seleccionar el capacitor adecuado que proporcione un filtrado correcto en las terminales de salida de la fuente, nosotros debemos primero calcular el valor de la capacitancia.

Para calcular la reactancia de diversos elementos pasivos como pueden ser los inductores y/o capacitores principalmente, se pueden recurrir a diversas técnicas, nosotros optamos por la carta de la reactancia para el cálculo de estos valores. Partiendo de las ecuaciones de la ley de Ohm que también son aplicables para encontrar la reactancia de los elementos de un circuito tenemos:

$$I = \frac{E}{X} \dots\dots\dots \text{Ec. C. 2}$$

despejando esta ecuación obtenemos:

$$E = IX \dots\dots\dots \text{Ec. C. 3}$$

$$\lambda = \frac{E}{I} \dots \dots \dots \text{Eq. C. 1}$$

Donde:

- E= EMF en Volts
- I= Corriente en Amperes
- X= reactancia en Ohms

Estas formulas como se menciono anteriormente, son aplicables para encontrar el valor de las reactancias, pero como herramienta se tiene a la carta de reactancias que podemos obtener mas rapidamente los valores para diferentes capacitancias que van de 10F a 0.1 µF, y tambien para diferentes inductancias que van de 0.1 mH a 10 H henrys, y para diferentes frecuencias entre 100 hertz a 100 Megahertz.

Por ejemplo si queremos calcular el valor de un capacitor de polyster a una frecuencia de 100 Hz, de acuerdo a la carta de reactancias tenemos un valor de reactancia de 19 ohms y una capacitancia de 100 µF. Por lo general las capacitancias grandes eliminan los pulsos de AC de las frecuencias bajas.

La forma de regulacion mas economica y sencilla es la que se realiza por med. de circuitos reguladores de voltaje de tres terminales por ejemplo el LM 317, LM 7800, LM 7900, estos dos ultimos se utilizan para valores fijos y el primero puede manipularse para valores de fuentes variable.

Para el calculo de los componentes externos que se necesitan para la correcta operacion de estos dispositivos recurrimos a las hojas de especificacion del fabricante así como sus recomendaciones para cada dispositivo.

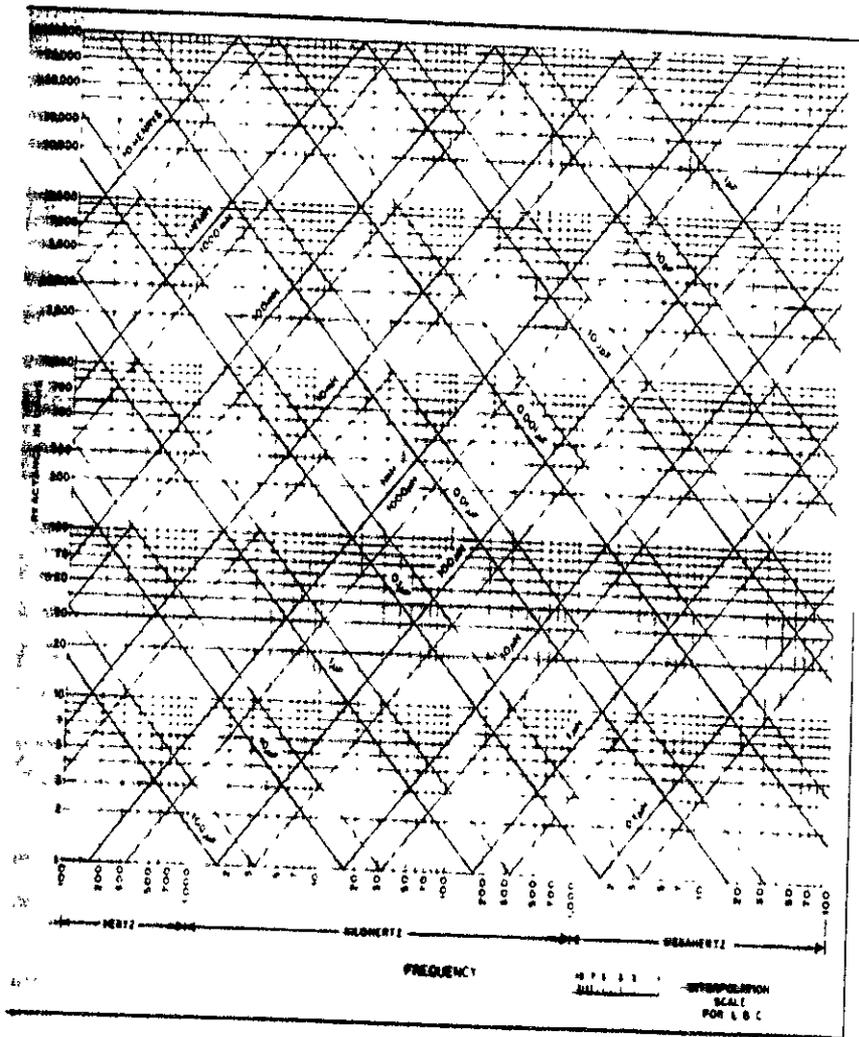


Figura C. 1 Carta de Reactancias para inductancias y capacitancias

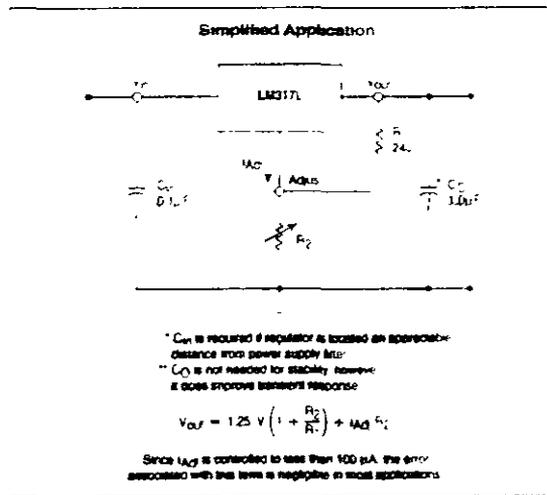


Figura C. 2 Cálculo para el voltaje de salida del LM 317.

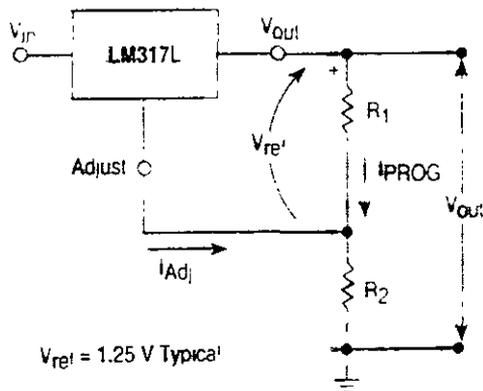


Figura C. 3 Configuración típica del LM 317.

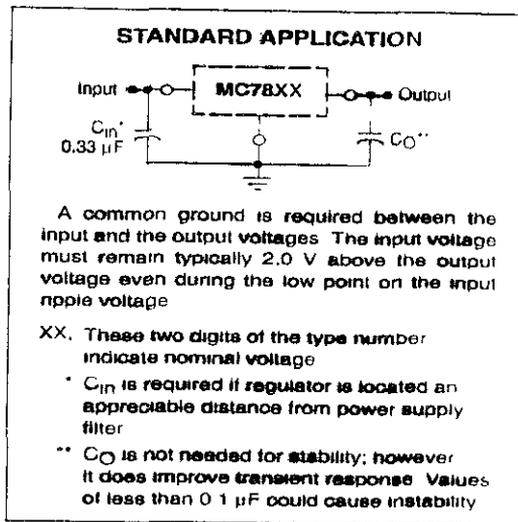


Figura C. 4 Aplicación estándar del LM 7800

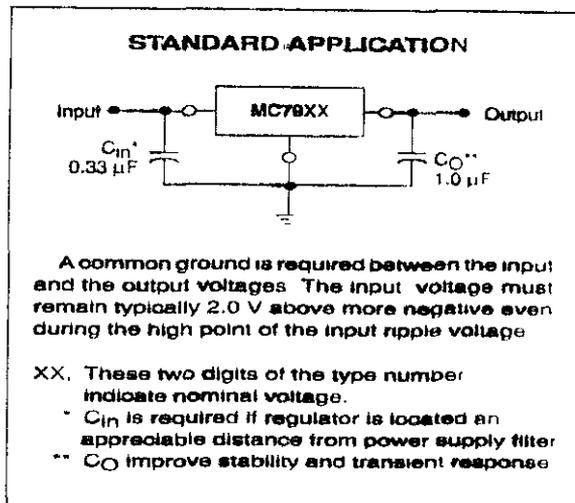


Figura C. 5 Aplicación estándar del 790.

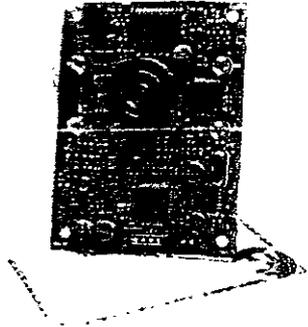
**APÉNDICE D**

- Esquema de la Tarjeta de Admisión de Cat. 77
- Imagen del circuito impreso de la tarjeta TA
- Especificaciones generales de la tarjeta de Adm. 77

# TEXAS INSTRUMENTS

## LC-324 Single Board Monochrome Low-Cost Camera Module

- Compact Single Board: 2.36 in x 3.16 in
- 12 V dc Operation
- EIA RS-330 Monochrome Video Output
- 324(H) x 242(V) Active Pixels Per Frame
- 1/4-in Frame Transfer CCD
- Electronic Auto-Iris Equivalent to 8 F Stops
- 60 (H) x 45 (V) FOV Custom Lens
- High IR Sensitivity and 4 On-Board IR Illuminating Diodes Allow For Night Vision Applications



### Applications

Camera applications requiring low cost, medium-resolution, performance, electronic auto-iris, low smear, low power, and small size are typical applications for the LC-324. The board has high near-IR sensitivity and is especially well suited for low-light applications.

#### Applications include:

- security surveillance
- door vision
- PC and workstation input
- automotive trucking and P.O. rear vision
- telecommunications/picture phone
- machine vision
- night vision

The effective resolution is 242 horizontal by 242 vertical lines, comparable to most standard consumer TV monitors and video recorders.

### Overview

The LC-324 is a small single-board black and white camera using the new 1/4-in TC255 Charge Coupled Device (CCD) sensor from Texas Instruments. The LC-324 module offers the superior advantages of the TC255 sensor, such as on-chip electronic auto-iris, antiblooming, capability of 20X, and very low smear. The board allows for 12 V dc input power operation and outputs standard EIA RS-330 monochrome video output. The electronic auto-iris yields 8 f stops of exposure control allowing operation from day to night. The module includes a custom-mount plastic lens with a 60° x 45° FOV. The camera electronics include the sensor, a clock generator and driver, video processor, IR illuminating diodes, and power supply - the only interface signals required are 12 V dc power and video out.

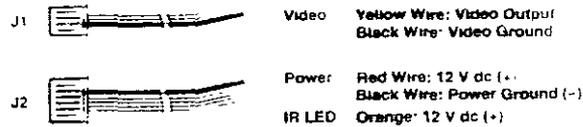
### Sensor Characteristics

The sensor is based on TI's patented advanced virtual-phase technology which provides high blue response, low dark signal, good uniformity, and single phase clocking. The sensor features a 4-mm-diagonal image sensing area configured into 243 lines with 324 pixels per line. The module operates the TC255 sensor in a standard field mode. The auto-iris function is implemented by controlling the integration time (i.e., the time between the clear of the image area and the transfer of the image into the storage area). The length of the exposure time depends upon the relative strength of the previous field. A relatively strong field causes a decrease in exposure time while a relatively weak field causes an increase in exposure time.

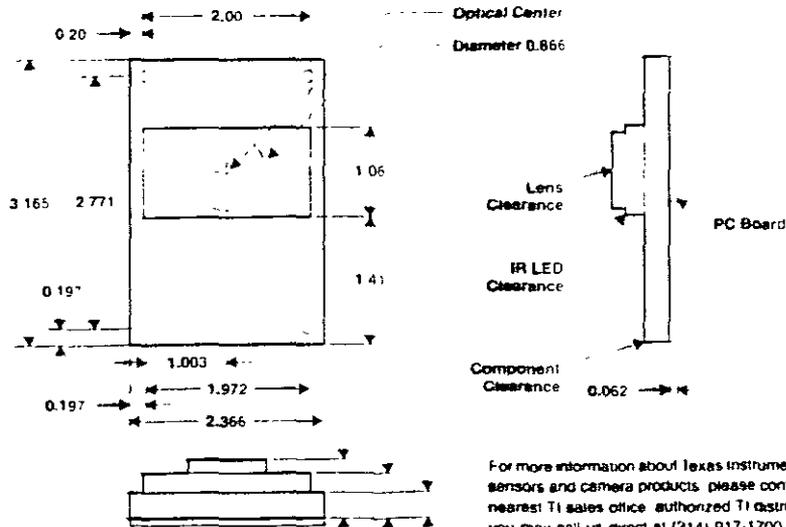
## Specifications

Imager:	Texas Instruments TC255 1/4-in Frame Transfer CCD
Total active pixels:	324(H) x 242(V) Pixels
Pixel size:	10 $\mu$ m X 10 $\mu$ m
Spectral response:	400 nm to 1100 nm
Effective resolution:	242(H) x 242(V) TV lines
Dynamic range:	Greater than 60 dB
Blooming overload ratio:	200 X saturation, typical
Smear:	0.011% (t <sub>int</sub> = 1.6 $\mu$ s, frame transfer rate = 12.5 MHz)
Gamma:	0.4
Video output:	1 V <sub>p-p</sub> @ 75- $\Omega$ (EIA RS-330)
Signal to noise ratio:	43 dB (t <sub>int</sub> = 1/60 s)
Sensitivity:	0.045 lx results in 50 IRE output (Gamma = 0.45)
Exposure control:	Electronic auto-iris (8 effective f-stops)
Shutter speed:	1/60 s - 1/8000 s
Pixel rate:	6.25 MHz
Scan format:	EIA RS-330
Synchronization:	Internal
Power supply: Camera:	12 V dc @ 100 mA maximum
Power supply: IR LED:	12 V dc @ 100 mA maximum
Power consumption:	0.89 W (without IR LED) / 2.27 W (with IR LED illumination)
Lens:	Custom 60(H) x 45(V)
Input/Output signals:	Power/Vide
Operating temperature:	-10 C to 40 C
Size: L x W:	3.16 x 2.37

## WIRING CONFIGURATION



## LC-324 BOARD CLEARANCE AREA



For more information about Texas Instruments CCD sensors and camera products, please contact your nearest TI sales office, authorized TI distributor, or you may call us direct at (214) 917-1700 or fax at (214) 917-1702.

All Dimensions are in inches

0.60 0.45 0.25





HOLE LEGEND			
INCH	TC	DIA	NO.
0.062		Ø	8
0.075		Ø	4
0.032		24x	
0.075		Ø	7
0.032		Ø	7
0.04		Ø	4
0.05		Ø	9
0.054		Ø	4
TOTAL			403



## APENDICE E.

Listado de los programas, que utiliza el colorimetro

del VETA

del estado :

del estado :

del estado : del estado :

del estado : del estado :

del estado : del estado : del estado : del estado : del estado :

del estado : del estado :

del estado : del estado :

del estado :

del estado : del estado : del estado : del estado :

del estado : del estado :

del estado : del estado :

del estado :

del estado :

del estado :

del estado :

del estado :

del estado : del estado :

del estado :

del estado : del estado :

del estado : del estado :

del estado :

del estado : del estado : del estado : del estado : del estado :

del estado : del estado : del estado : del estado :

del estado : del estado :

del estado : del estado :

del estado : del estado : del estado : del estado : del estado :

del estado : del estado :

del estado : del estado :

del estado :

del estado : del estado :

del estado : del estado :

del estado : del estado : del estado : del estado : del estado :

del estado : del estado :

## APENDICE E.

Listado de los programas, que utiliza el colorimetro.

```
ExitError: "Error al intentar conectar la base de datos".
```

```
exit
```

```
1 CURVA = database.cursor("SELECT color, indice, valor FROM CURVA", tipo  
CURVA SELECT")
```

```
2 CURVA = database.cursor("SELECT color, indice, valor FROM CURVA", tipo  
CURVA SELECT")  
3 CURVA = database.cursor("SELECT color, indice, valor FROM CURVA", tipo  
CURVA INSERTION "CURVA")
```

```
CURVA SELECT")
```

```
database.disconnect()
```

```
archivo = new File("salida.txt")
```

```
if (archivo.exists())
```

```
1 archivo.delete()
```

```
2 archivo.createNewFile()
```

```
3 archivo.close()
```

```
delete archivo
```

```
1 archivo = new File("salida.txt")
```

```
archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
1 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
2 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
3 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
4 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
5 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
6 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
7 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
8 archivo = new File("c:/tesis/tesis/entrada.txt")
```

```
9 archivo = new File("c:/tesis/tesis/entrada.txt")
```

## APÉNDICE E.

Listado de los programas, que utiliza el colorímetro.

```
.....  
*FUNCIONES A LAS QUE HACEMOS LLAMADA  
*LENGUAJE: JAVASCRIPT  
.....
```

```
function IntToHex Number:
```

```
Var i;  
var hex = ""
```

```
Remove when the I/O handles characters above 255  
Number = Number < 256 ? Number : Number;
```

```
if (Number ==  
hex = ""  
else  
for ( i = Number; i > 0; i = IntDiv(i,16)  
hex += "0123456789ABCDEF".substring(0,16 - i % 16).charAt(i % 16)  
hex = RevString(hex) + (hex.length > 1 ? "" : "0")
```

```
return hex;
```

```
function IntDiv Number, Divisor:
```

```
return Number / Divisor - (Number / Divisor < 0 ? 1 : 0);
```

```
function RevString(Source)
```

```
Var i;  
var RetStr = "";
```

```
for ( i = Source.length; i > 0; i--  
RetStr += Source.substring(i-1, i);
```

```
return RetStr;
```

```
function LPad String, Length, Char:
```

```
Var i;  
var PadStr = "";  
var RetStr = "";
```

```
if (Char == null  
Char = " ";  
else  
Char = "" + Char;
```

```
if (String.length < Length  
for ( i = 0; i < (Length - String.length); i++  
PadStr += Char;
```

```
RetStr = PadStr + String;
```



# APENDICE E.

Listado de los programas, que utiliza el colorimetro.

POSTER.HTML PUBLIC: (HTTP/STD HTML 1.2//EN)

<HTML

<HEAD

TITLE color wheel (TITLE

CONTENT LANGUAGE=avaes:en

...>

This script gets the x and y positions of a client-side image and sets the image to a server-side image map.

The client is set to reload the page when returned the x and y values of the document location as values following.

The x and y values for the point clicked are calculated and the background of the page is changed to that color.

```
hexadecimal_Convert()
function hexadecimal_base
    var hexList = "ABCDEF"
    var hexVal = Math.floor(Math.random() * 16)
    dig = hexList.charAt(hexVal)
    dig = hexVal + dig
    dig = hexList.charAt(dig)
    dig = hexList.charAt(dig)
    dig = hexList.charAt(dig)
    hexVal = dig + dig + dig + dig
    return hexVal

```

```
function hexToint(hex)
    // se asume que el valor es de dos digitos
    dig = hex.charAt(0)
    dig = hex.charAt(1)
    dig = dig * 16 + dig
    dig = dig * 16 + dig
    numero = dig * 16 + dig
    return numero

```

function IntToHex(Numero)

var

var Hex = ""

while (Numero > 0) // handles characters and numbers

var Numero =

Hex = "00"

while

Hex = "0123456789ABCDEF".substr(0, Hex.length + 1)

Hex = Hex + "0123456789ABCDEF".substr(0, Hex.length + 1)

Hex = Hex + "0123456789ABCDEF".substr(0, Hex.length + 1)

return Hex

Calculate the color value for Red, Green, and Blue

Value is between 0 and 255

For RED

In the area bounded by an angle of 60 degrees and 120 degrees Value

is:

(This is a right-hand-side quadrant)

## APENDICE E.

```
Listado de los programas, que utiliza el colorimetro.
Outside this area the value decreases linearly from the boundary of the
area to the edge of the circle of a line parallel to the x-axis
for GREEN.
The coordinates must be rotated 14 degrees clockwise and x and y be
calculated.
This transforms the circle so that the same calculation as for PER
value.
* * * * *
* * * * *
The coordinates are rotated 14 degrees.
*
function colorVal, color, ang, vect, xPos, yPos
with {Ma}
    var angRot = 0;          // rotation angle = 0 if Re
    if color = "green" angRot = 2*PI/3;
    if color = "blue" angRot = 4*PI/3;
    var angVal = ang + angRot; // Apply rotation
    if (angVal < 0) angVal += 2*Math.PI; // If angle
negative add 360 degrees
    var xVal = xPos; // If red, set x-value
    var yVal = yPos; // If red, set y-value
    if color = "red"
        yVal = abs(yVal) * sin(angVal); // Calculate new y
    else
        xVal = abs(xVal) * cos(angVal); // Calculate new x
    if (angVal > PI) && (angVal < 2*PI) // Get the
        xVal = -1 * xVal;

    var colVal = 0; // initialize the color value
    if (angVal >= 2*PI/6) && (angVal <= 10*PI/6) // If inside
the quadrant
        colVal = 65535;
    else // If outside the quadrant,
        var x = sqrt(10000 - yVal*yVal) + xVal;
        var xi = abs(yVal) / tan(PI/6) - xVal;
        colVal = 65535 * xi * xi + x;

    colVal = round colVal;
    return colVal;
}

* * * * *
1. Get the x and y coordinates using the Netscape example code
The gif is 216 pixels x 216 pixels. The circle has a radius of 10
pixels.
2. Set the x0 and y0 values to the center of the circle by subtracting
10 from x and y.
3. Calculate r, the vector from the center to the x-y coordinate.
4. Calculate theta, the angle from the x-axis to the x-y coordinate
(counterclockwise).
5. Calculate the RGB values (0 to 655).
6. Convert to Hexadecimal values.
7. Put the values in the form
8. Change the background color to match the chosen color.
*
var red = 0;
var green = 0;
var blue = 0;
```

Listado de ACS programas, que utiliza el colorimetro  
rrior\_colo=.

function show1.

```
var str = location.search;
if (str != "")
  comma = str.indexOf(",");
  va. = str.substring(0, comma);
  var i = str.substring(comma+1, str.length);
  var i = Math.sqrt(i*i + 1);
  if (i <= 1)
    with (Math)
      var theta = asin(va. / i);
      i = i * i;
      theta = theta * 180 / Math.PI;
      i = i * i;
      theta = theta * 180 / Math.PI;
      var theta_0 = theta * i * i;

  red = colorVal("red", theta, i, va.);
  green = colorVal("green", theta, i, va.);
  blue = colorVal("blue", theta, i, va.);
  red = hex(red, 255);
  green = hex(green, 255);
  blue = hex(blue, 255);
  document.getElementById("val").value = i;
  document.getElementById("tree").value = green;
  document.getElementById("blue").value = blue;
  parent.frames[2].document.bgColor = "#" + red + green +
```

blue

else

alert("Selecciona dentro del listado.")

function bright

```
var bval = 1;
fval = document.colorVals.bright.value;
if (fval <= 1) && fval >= 0)
  bval = bval / bright;
  with (Math)
    newRed = round(eva. " " + document.colorVals.red.value *
fval);
    newGreen = round(eva. " " + document.colorVals.green.value *
fval);
    newBlue = round(eva. " " + document.colorVals.blue.value *
fval);
```

```
newRed = newRed * 255 / 255;
newGreen = newGreen * 255 / 255;
newBlue = newBlue * 255 / 255;
newRed = hex(newRed, 255);
newGreen = hex(newGreen, 255);
newBlue = hex(newBlue, 255);
document.colorVals.red.value = newRed;
document.colorVals.green.value = newGreen;
document.colorVals.blue.value = newBlue;
parent.frames[2].document.bgColor = "#" + newRed + newGreen + newBlue;
bright_old=document.colorVals.bright.value;
```

else



