



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DISEÑO Y CONSTRUCCIÓN DE UN THEREMIN BASADO EN UN
PROCESADOR DIGITAL DE SEÑALES**

TESIS QUE PARA OBTENER EL TITULO DE INGENIERO EN COMPUTACIÓN
PRESENTA

ROSA ISELA HERNÁNDEZ DÍAZ

Director

ING. JOSÉ ANTONIO DE JESÚS ARREDONDO GARZA

MÉXICO, D. F.,

2007.





Universidad Nacional
Autónoma de México



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.

“... Me gustaría decirte lo que siento con un silencio
que expresarlo en un glissando...”

Rosa Isela Hernández Díaz

“...Mira:
Una carrera profesional lleva años de estudio,
años en que la dedicación a aprender
supone la renuncia a muchas diversiones y cosas gratas,
pero al final el éxito de la graduación compensa los esfuerzos...”

Ana María Rabatte

DEDICADO A...

Papá Dios:

Que sin él y sin su apoyo hoy no estaría aquí. Te Amo y te doy las gracias por la familia que elegiste para mí y por tus infinitas bendiciones para los míos y para conmigo. Gracias por no desampararme y por darme el privilegio de ser tu hija.

Abuelita Seve⁺:

Lo prometido es deuda y hoy la dicha que siento es infinita y tu sabes muy bien que todos los días me acuerdo de ti y de esas palabras que tuviste para mí la última vez que nos vimos. Donde quiera que te encuentres espero que te sientas orgullosa de mí y que sepas que siempre cuidaré de mi mamá y que nada le faltará. Te Amo y Te extraño y nunca olvidaré la bella infancia que me diste y tu protección. Nunca me olvidaré de ti y tú tampoco te olvides de mí, cuídanos mucho.

Rubén Díaz Sánchez⁺:

Te extraño, no pude despedirme de ti, pero tu sabes lo importante que fuiste para mí y que le doy gracias a Dios por haberme bendecido con un amigo como tu. Mis quince años fueron inolvidables gracias a ti. Gracias mi querido tenor. Cuídame.

Mis papás, Nicolás y Victoria:

Los amo, gracias por haberme dado la vida, todo lo que soy se los debo a ustedes. Nunca podré pagarles ni aún con todo el dinero del mundo lo que han hecho por mí y todos sus sacrificios. Gracias porque en vida me han dado la mejor herencia... mi carrera.

Papi, gracias por tu leche caliente de todas las noches, por darnos el lugar que se merece tu familia, por trabajar por nosotros, pero sobre todo por ser mi padre. Te Amo.

Mami, gracias por ser además de mi madre una de mis mejores amigas, por predicar con el ejemplo y no sólo con palabras, por defenderme, por defender a mis hermanos, por trabajar tanto para tu familia, por darme la vida, pero sobre todo por lo de hace cinco años. Te Amo.

Mis abuelitos, Concepción e Hilario:

Es un privilegio ser su nieta, sin ustedes y sin sus cuidados hoy no estaría aquí, gracias por enseñarme que querer es poder y que cualquier persona siendo humilde y sencilla es aceptada en todas partes. Lo mejor que ha podido sucederme es tenerlos a ustedes, porque sé que me quieren y me aman sin condiciones y me aceptan como soy al igual que a mis hermanos. Así que hoy junto conmigo ustedes también se reciben y sientan este triunfo de hoy como suyo. Los Amo.

Mis hermanos, Diana y Carlos:

Por aguantarme tanto, por hacerme reír, por hacerme soñar, no sé que haría sin ustedes que son mi complemento.

Diana, gracias por ser mi hermana, tu sabes que eres fundamental en mi vida, que Te Amo y que espero que siempre estemos unidas.

Carlos, gracias por ser el ángel que salvó a mi madre, por ser mi hermano, pero sobre todo mi latosito, eres esencial, porque le das vida a mi vida. Te Amo.

Mis tíos y maestros, Jesús Díaz, José Alfredo Govea y Tomás Macías:

Sin ustedes no hubiera entendido esas cosas que se me complicaban cuando era niña y hoy les agradezco haber sido parte este sueño, pero sobre todo les doy las gracias por ser parte de mi familia, por ser mis tíos, pero más por sus enseñanzas. Los quiero.

Mis mejores amigos, Coral Montes y Tulio González:

Mi vida no sería igual sin ustedes, gracias por su apoyo, por su comprensión, por todo este tiempo de amistad, por estar tanto en las buenas como en las malas conmigo, por todo lo que juntos hemos pasado, por nuestras aventuras, pero sobre todo por ser como mis hermanos. Tengo la certeza de que siempre estaremos juntos. Los quiero.

Mis amigos, Patricia Granados, Osvaldo González, Aureliano Saavedra, Alfredo Sánchez, Ruth Hernández, Gaspar Cárdenas, Federica Castañeda y Roberto Tejero:

Ustedes son como mis hermanos mayores, les doy las gracias por enseñarme a vivir con optimismo, por los momentos que hemos compartido, por sus palabras, porque nunca olvidan un cumpleaños mío y por ser incondicionales. Los quiero.

Mis tíos, Jesús, José Luís, Raquel, Rosario, María, Yolanda, Leticia, Rosangeles, Agustín, Héctor, Tomás, José Alfredo, Laura y Benjamín:

Por sus consejos, por no dejarme sola en este camino, porque han ayudado a mis padres y a mis hermanos, por su ejemplo, pero sobre todo porque siempre hemos estado unidos y porque sé que me quieren tanto como yo a ustedes.

Mis primos, Guadalupe, Alejandra, Marisol, Jesús, Rosario, Phany, Alfredo, Alondra, Toño, Lalo, Jenny, Fernando, Adriana, Juan Manuel, Pablo y Edgar:

Gracias, por esa infancia tan maravillosa que tuvimos, por nuestros juegos, por nuestras risas y por nuestros sueños de niños y por compartirlos conmigo. Somos unos primos especiales, al tener a nuestro lado a un angelito como Fernando. Los Amo.

Mis profesores, Pablo Rosell, Alejandro, Margarita Puebla, Manuel Vacío, Rafael Iriarte, Francisco Barrera, Bertha Franco⁺, Jaquelina López, Lucila Arellano, Carmelita, Patricia Pacheco, Eduardo Carranza, Arturo Zapata, Abel Herrera, Bernardo Navarro, Rosalinda, Socorro Alcántara, Rosario y Chela⁺:

Por ser mis guías, por sus enseñanzas, pero sobre todo por formar una parte fundamental en el logro de este sueño que hoy comparto con ustedes y espero no defraudarlos y que se sientan orgullosos de mí. Ha sido un privilegio ser su alumna.

Mi director de tesis, José Antonio de Jesús Arredondo Garza:

Por su paciencia, por su trabajo, por su enseñanza, pero sobre todo por su fortaleza y por ser mi guía en este camino de la música y de la aplicación de la Ingeniería en esta y conjugar dos de mis tres grandes pasiones; la Ingeniería y la Música.

Mi Universidad, La Universidad Nacional Autónoma de México y a la Facultad de Ingeniería:

Por ser mi ejemplo de sabiduría, por darme asilo a través de estos años y por ser como mi segundo hogar, gracias.

Al hijo(s) que algún día tendré:

A ustedes y que aún sin conocerlos, en gran medida, este triunfo y esta dicha van por nuestra unión, por ofrecerles un futuro mejor y porque me gustaría que se sintieran orgullosos de su madre así como yo lo estaré de ustedes.

ÍNDICE

TEMA:

INTRODUCCIÓN	7
---------------------------	---

CAPÍTULO I. ANTECEDENTES

1.1. Corrientes Musicales del Siglo.....	9
1.2. Escalas Musicales.....	12
1.3. Historia de los Instrumentos Musicales Electrónicos.....	14
1.3.1. Los Sintetizadores y los Instrumentos de Música Electrónica.....	19
1.3.2. Moog.....	20
1.3.3. Minimoog.....	21
1.3.4. Buchla	22
1.4. Diagrama de Bloques de un Sintetizador por Hardware Elemental... ..	27
1.5. El Theremin	29
1.6. Planteamiento del Problema de Diseño de un Theremin Digital.....	31

CAPÍTULO II. EL PROCESADOR DIGITAL DE SEÑALES dsPIC30

2.1. Arquitectura	33
2.1.1. RISC (Reduced Instruction Set Computer).....	33
2.1.2. Arquitectura Harvard.....	37
2.1.3. Características Principales de un DSP.....	39
2.1.4. Características Principales del dsPIC30F4013.....	41
2.1.5. Registros de la CPU.....	43
2.1.6. Mapas de las Memorias Internas.....	49

CAPÍTULO III. EL EMISOR Y SENSOR ULTRASÓNICOS

3.1. Principio Fundamental de Funcionamiento.....	51
3.2. Descripción del Sensor Ultrasónico.....	51
3.3. El Emisor y Receptor Ultrasónicos.....	53
3.3.1. El Emisor.....	53
3.3.2. El Receptor.....	54
3.4. La Interfase Digital Analógica.....	56

CAPÍTULO IV. DISEÑO DEL SOFTWARE

4.1. El Compilador de Lenguaje C.....	59
4.2. Rutina Principal del Sistema.....	61
4.2.1. Subrutinas llamadas por la Rutina Principal	63
4.3. Rutina de Interrupción del Módulo de Captura de Entrada IC1.....	66

4.4. Rutina de Interrupción del Módulo de Captura de Entrada IC2	68
4.5. Rutina de Interrupción del Timer 4.....	69

CAPÍTULO V. RESULTADOS

5.1. Pruebas y Análisis.....	71
5.2. Fotografías de nuestro proyecto del Theremin	72

CONCLUSIONES	73
---------------------------	----

APÉNDICE 1	77
-------------------------	----

APÉNDICE 2	109
-------------------------	-----

REFERENCIAS	115
--------------------------	-----

INTRODUCCIÓN

Con la realización de este proyecto lo que se pretende es mostrar la relación que existe entre las Matemáticas y la Ingeniería aplicada para con la música.

Asimismo, la propuesta que se hace es la de simplificar un instrumento musical que para ser más precisos fue de los primeros instrumentos electrónicos de su época y que como tal llegó a revolucionar al mundo de la música.

En este proyecto de tesis para poder llegar a la optimación de este instrumento musical, recurrimos al uso del dsPIC30F4013, el cual nos fue de gran ayuda puesto que es sencillo de manejar y de programar al no manejar tantas instrucciones.

También nos fue de gran ayuda la utilización de los sensores ultrasónicos, con lo cual ya no se volvió un aparato tan ostentoso como lo es el Theremin, que cuenta con las dos antenas, con este tipo de sensores y como se explicará más adelante única y sencillamente basta con no ir más allá de los 857 cm de distancia entre las manos y el dispositivo ya que el sensor capta el calor humano.

Por otro lado, lo que nos motivó a desarrollar este proyecto fue la necesidad que existe entre los músicos de seguir creando obras para ser tocadas con este instrumento tanto a nivel de orquesta como a nivel individual, además de que se sigan añadiendo a la lista más músicos que quieran tocar este instrumento, ya que existen muy pocos en el mundo y de gran tamaño.

Creemos firmemente en que al menos nuestro Theremin, será un impulso para que este tipo de instrumentos salgan del olvido y con él se nos puedan mostrar nuevas corrientes de la música.

Para el primer capítulo nos hemos enfocado en los antecedentes, en todo lo que conlleva una parte de la historia de la música, de los instrumentos electrónicos musicales, los sintetizadores y del manejo de los Theremins clásicos, para de esta forma dejar un registro de cómo surgió este instrumento musical.

En el capítulo II, explicamos como es el diseño RISC, la arquitectura Harvard y la Arquitectura según el modelo de Von Neumann y los registros de la CPU del dsPIC30F.

En lo que respecta al capítulo III, nos centramos en la descripción fundamental del funcionamiento del emisor y sensor ultrasónico, en el receptor y en la interfase digital analógica, así como a la descripción del circuito del emisor y sensor ultrasónico, además del circuito del DSP y la interfase analógica.

En relación al capítulo IV, explicamos el compilador de lenguaje C que se usó para programar el DSP y las rutinas y subrutinas que conforman el funcionamiento del Theremin.

Posteriormente, se ha añadido una parte de pruebas y resultados, en la que usamos un osciloscopio y lo conectamos a la salida de audio del Theremin, dicha prueba se explicará más adelante. Asimismo se dan las conclusiones a esta tesis.

CAPÍTULO I

ANTECEDENTES

1.1. CORRIENTES MUSICALES DEL SIGLO XX

La música de la misma manera que otras artes y ciencias han dependido de las costumbres que adoptan cada pueblo y cada época [1].

Las diferentes corrientes musicales del siglo XX tienen su origen en todos los movimientos musicales que se presentaron en los siglos anteriores que finalmente desembocaron en el Modernismo y el Vanguardismo.

Desde fines del siglo XIX y principios del XX se dieron corrientes musicales que comienzan a separarse de la tradición de la tonalidad, en la que se basó toda la música desde el Barroco. La variada música del Siglo XX no sólo ha destacado por la cantidad sino básicamente por haber planteado nuevos rumbos en cuanto a la realidad entre el compositor y las personas que disfrutaban de sus obras.

Para comprender la música del siglo XX, es importante acercarse primero a la del siglo XIX, y en especial a una corriente llamada Impresionismo. El Impresionismo nace como una nueva etapa en la cual los artistas van a expresar el mundo según lo ven y no guiados por las convenciones y costumbres del momento. Se va a luchar por una música más individual y personal, saliéndose para ello de las normas establecidas.

En el Impresionismo [1], lo que se busca es el color que puede dar el sonido al ser ejecutado con diferentes niveles de intensidad, lo cual al combinarse con diferentes acordes va a crear atmósferas e ilusiones que evocan imágenes en donde existe un fondo y forma. Dicho de otra manera, se presenta al sujeto y su entorno a través de temas y efectos sonoros mediante los cuales dicho sujeto forma parte de un cuadro sonoro dentro del cual los sonidos individuales forman parte de las pinceladas de dicho cuadro.

Dentro del Impresionismo, se comienzan a utilizar escalas no convencionales como la escala hexafónica formada por tonos enteros y el uso de acordes aumentados y disminuidos que rompen con toda la armonía clásica que se venía utilizando desde siglos pasados e incluso el uso de la politonalidad. Aquí lo que se pretende es buscar efectos y colores nuevos del sonido que difícilmente se apegarán a las normas que estaban preestablecidas, dándose pie de esta manera a las nuevas armonías que posteriormente se usarán en el siglo XX [1].

Dos de los músicos más importantes de este movimiento son Claude Debussy y Maurice Ravel, aunque hubo otros.

Otra corriente fue el Neoclasicismo, surge hacia 1920 como reacción frente al Romanticismo, incluyendo el Impresionismo y el Expresionismo. La actitud antirromántica rompe con el pasado inmediato y se siente familiarizada con los conceptos anteriores al Romanticismo, especialmente los del siglo XVIII. De nuevo se adoptan modos de interpretación, formas y géneros del Barroco y el Preclasicismo, como suite, concierto, sinfonía y sonata. Los músicos de este movimiento utilizan para sus composiciones toda la historia de la música, con sus diferentes estilos (incluido todo el siglo XIX, con criterio distanciador, como en el resto de los casos), así como la música de culturas no europeas y el jazz. Al igual que en la literatura, se pretende destruir la rutina tanto en la creación como en la audición, a través de efectos de distanciamiento y parodia. El Neoclasicismo es tonal y, a partir de los años treinta, se acentúa la tendencia hacia un ordenamiento racional de las formas y los géneros. El movimiento finaliza entre 1950 y 1960 [2].

Prokófiev [2], como compositor se adhirió al Neoclasicismo y muchas de sus composiciones tienen claras referencias al estilo del siglo XVIII y a Haydn. Este estilo aparece claramente definido en su sinfonía "Clásica" que fue compuesta en 1917. Su obra estuvo muy limitada por considerársele indeseable, para los tiempos que Rusia vivía en 1936 y este es un claro ejemplo de que había influencia política en la música.

Dentro del siglo XX aparece también la franca ruptura de la tonalidad y el atonalismo, y como ejemplo de ello podemos mencionar a Arnold Schoenberg [2], quien incluso propuso un modelo como tema basado en los 12 sonidos de la escala cromática. Su modelo consiste en presentar 12 sonidos de una manera predestinada para después desarrollar ese tema jugando con esos sonidos, lo cual trajo una controversia dentro de la música porque rompe con cualquier concepto melódico clásico. A la corriente propuesta por Schoenberg, se le llamó el Dodecafonismo.

Bela Bartok [2], en cambio propone en su música el rescate del Modalismo basado en las antiguas escalas griegas, aunque también fue un nacionalista que se interesó por la música folklórica de su país.

Bartok, dentro de su música propone un nuevo lenguaje que se sigue utilizando hasta nuestros días porque en sus armonías se utilizan nuevos elementos armónicos, melódicos y modales.

De esta misma forma va surgiendo el Serialismo Integral, que es parecido al Serialismo Dodecafónico, pero ampliando aún más el concepto de serie a parámetros como lo son la intensidad, el timbre y la duración. Dos de sus mejores exponentes son Oliver Messiaen y Stockhausen [2].

Otra corriente de la cual podemos hacer mención es de la Música Concreta o Aleatoria que se dio a finales de los 50's y principios de los 60's y que hasta la fecha sigue.

A grandes rasgos la música concreta, es un collage musical, que trabajaba con una cinta magnetofónica. Su materia prima eran los sonidos grabados con un micrófono y luego

manipulados electrónicamente (con reverberación, cambio de velocidad) y mezclados (recortando y haciendo un collage con los trozos de cinta).

En la actualidad la Música Concreta ya no se hace manipulando cintas magnéticas, sino con el uso de computadoras que permiten manejar la grabación pegando e insertando segmentos de la misma, invirtiendo la grabación o acelerando o retardando la misma. También es posible superponer grabaciones creándose así la experimentación con infinitud de nuevos sonidos.

En la práctica esta música es usada en la ambientación de películas, pero también hay gente que se dedica a presentar recitales que en realidad vienen siendo solo experimentos sonoros [1], [2].

También se está usando a la Música Concreta en combinación con la Música Tradicional haciéndose presentaciones de conciertos en vivo de músicos acompañados por cintas pregrabadas.

El Microtonalismo, es otra corriente que surgió en el siglo XX, en esta destacan, Julián Carrillo, quien es el primero en generar la idea de Microtonalismo, decía que cualquier escala musical puede ser subdividida en intervalos más pequeños que un semitono, al cual corresponde la relación matemática $\sqrt[12]{2}$ (división de un semitono).

Un intervalo más pequeño que un semitono podría ser por ejemplo $\sqrt[24]{2}$, que correspondería a $\frac{1}{4}$ de tono, o bien, $\sqrt[48]{2}$, que correspondería a $\frac{1}{8}$ de tono.

Otro representante del Microtonalismo fue también el mexicano, Augusto Novaro, que en 1927, creó escalas microtonales basadas en la sucesión de armónicos característicos a una escala musical básica, la cual planteó nuevas posibilidades a la música Microtonal, en donde las relaciones matemáticas no necesariamente tenían que ser iguales a lo largo de la escala.

Una de las nuevas tendencias que surgen en los años 60 y que resultan más interesantes y aceptadas es el Minimalismo, que es una reacción absoluta a los excesos de las creaciones de la posguerra, y un intento de volver a la claridad y la sencillez de la obra. Esta corriente trabaja con un estilo de música más sencillo y claro, utilizando un material más reducido con una estructura tonal estática y que se limita a uno o muy pocos acordes distintos y repetitivos, la insistencia de ritmos aditivos, la consistencia textural o la constante repetición temática.

El desarrollo de los instrumentos musicales electrónicos a lo largo del siglo XX y siglo XXI ha permitido afinarlos bajo cualquier sistema de afinación microtonal, lo cual ha permitido a mucha gente seguidora de las corrientes microtonales componer música vanguardista con relativa facilidad.

1.2. ESCALAS MUSICALES

Las bases fundamentales de las escalas musicales que comúnmente se usan en occidente tienen sus orígenes en la antigua Grecia.

Se puede mencionar que Pitágoras, fue uno de los primeros que estudió la estructura matemática de las escalas musicales y además creó algunas escalas nuevas conocidas como Pitagóricas que hasta la fecha se siguen utilizando por personas dedicadas a la investigación musical.

Antes de que apareciera el concepto de tonalidad aparecieron en Grecia las escalas naturales basadas en el desarrollo natural de las mismas, es decir, que todas carecían de las alteraciones causadas por los sostenidos y bemoles.

Las escalas naturales modales eran las siguientes: la Iónica, la Dorica, la Frigia, la Lidia, la Mixolidia, la Eolica y la Locrian [3].

Por ejemplo, la escala en Modo Dorico, está formada por las notas re, mi, fa, sol, la, si, do y re, pero sin ninguna alteración. El Modo Mixolidio, está integrado por las notas sol, la, si, do, re, mi, fa y sol, sin tener la alteración del fa# que se esperarían en la escala del sol mayor.

En la composición de obras modales, al igual que en las escalas modales, también se conservaba la forma natural que dictaba la escala, es decir, la ausencia de alteraciones, teniendo como consecuencia esto el hecho de que esta música tenga un carácter antiguo en sí misma.

Bela Bartok, en su Modernismo utilizó elementos modales dentro de su composición musical y dentro de su obra modal podemos mencionar sus cinco libros para piano conocidos como Mikrokosmos.

Una de las culturas que desarrolló un sistema musical microtonal fue la cultura hindú, de la cual poco se sabe en la actualidad.

En la cultura occidental las escalas musicales están basadas en 8 sonidos, tomando en cuenta que existen escalas cromáticas basadas en 12 sonidos.

Cabe mencionar que la cultura china y curiosamente algunas de las culturas mesoamericanas tenían escalas sólo basadas en 5 sonidos (pentáfonas), mientras que la cultura hindú, en cambio tuvo escalas basadas en más de 12 sonidos.

Los sonidos musicales son infinitos, clasificándolos en siete grados y que denominaremos: do, re, mi, fa, sol, la, si, al ascender del si se llega al duplo del do, es decir, el do² y si seguimos ascendiendo de esta forma llegaremos al do³, al do⁴, etc.

Se definen en siete grados a la escala musical, no porque invariablemente sean siete, pueden ser cinco o nueve, que tal vez sean los indicados, o bien, los que fuera más práctico fijar, tanto más que hay escalas de gran eufonía con menos sonidos, pero como homenaje al pasado, se han determinado en siete los grados, ya que en dicho número y en escalas clasificadas como irregulares han impreso pensamientos inmortales que son hermosos y majestuosos al escucharlos en el Sistema Natural; en cualquier orden de escala que se fijara no alteraría en nada a las restantes, pues la fuerza expresiva de la música radica en gran parte en la unión de todas ellas.

Igualmente, lo más cómodo sería prescindir de las sílabas do, re, mi, fa, sol, la, si, ya que se precisan mejor los sonidos usando únicamente las vocales, pero al suprimirlas se destruiría un bello concepto de la música.

Dice la historia musical que fueron tomadas de las primeras sílabas de cada emistiquio de un himno a San Agustín, que decía:

Ut queant laxis
Resonare fibras
Mira gestorum
Famuli tuorum
Solve polluti
Labbii Restum

El nombre de la nota do de acuerdo a los emistiquios anteriores no se conservó en la lengua española, pero en el italiano y el francés actualmente existe [4].

1.3. HISTORIA DE LOS INSTRUMENTOS MUSICALES ELECTRÓNICOS

Antes de mencionar a los instrumentos musicales electrónicos es necesario mencionar los orígenes de la música producida por los sintetizadores electrónicos.

El instrumento de síntesis musical más antiguo que se conoce es el órgano accionado por aire, dado que su principio de operación está basado en la suma de armónicos para obtener un determinado sonido.

Como anécdota sobre la Historia del órgano podemos mencionar que a Nerón le gustaba tocar el Órgano, aunque el órgano del siglo I d.C. no era como el que conocemos en la actualidad.

El órgano con el teclado que conocemos data de la Edad Media y dio pie al desarrollo de instrumentos musicales basados en el mismo principio de ese teclado (tablatura), como son el clavecín y el piano que se desarrollaron posteriormente.

En el órgano se desarrolló el concepto de “registro”, que consiste en la adición de sonidos mediante el uso de botones que controlan la emisión de aire hacia los diferentes tubos mejor conocidos como flautas.

La contaminación de un sonido fundamental con armónicos predestinados nos da la característica timbrica del órgano y ello se logra a través de la llamada registración. A través de ella se buscaba imitar el sonido de instrumentos musicales e incluso de la voz humana.

Es muy usual ver registraciones con los nombres Voz Humana, Trompetas, Cuerdas, Lengüetas (clarinetes), etc., y como su nombre lo sugiere, se refiere a la síntesis de dichos instrumentos.

Dentro de los órganos podemos hablar de dos clasificaciones: el órgano positivo y el órgano negativo.

El órgano positivo es aquel que funciona mediante la adición de armónicos para formar un determinado timbre sonoro, mientras que el órgano negativo logra este mismo fin utilizando la resta de armónicos.

Cabe hacer mención con base en lo anterior que según la serie de armónicos dada por Fourier, cuando dos o más sonidos se mezclan se generan armónicos de alta frecuencia y simultáneamente también se generan armónicos de baja frecuencia, siendo este último principio utilizado por los órganos negativos para generar sonidos que físicamente serían imposibles de generar en forma normal por un órgano carente de recursos.

No es raro que en el órgano se haya optado por imitar los sonidos de los instrumentos musicales mediante el uso de armónicos, dado que la mayoría de los compositores hasta la fecha utilizan empíricamente los sonidos de la orquesta mezclados de tal manera que se logren los colores de sonido deseados.

Como última mención con respecto a los órganos podemos decir que evolucionaron según el país, dando origen a registraciones características del órgano español, el órgano francés y el órgano alemán.

En México el desarrollo de los órganos tuvo su desarrollo característico durante la época de la colonia, pero desgraciadamente se perdió toda esa tradición en la Revolución Mexicana.

Con el advenimiento de la electrónica se comenzaron a desarrollar órganos electrónicos que en principio distaron mucho de parecerse al órgano tradicional, como fue el órgano Hammond y muchas de las versiones desarrolladas en los 60's y que incursionaron dentro de diferentes géneros de la música (rock, iglesias, etc.).

Un instrumento electrónico o eléctrico es un instrumento musical que produce sus sonidos usando la electrónica. Es decir, son aquellos instrumentos en los cuáles el sonido se genera o se modifica mediante corrientes o circuitos eléctricos.

Los instrumentos electrófonos, son la última familia en integrarse a la clasificación Sachs - Hornbostel, aproximadamente en 1940 [5].

Los electrófonos se dividen en 2 grupos:

- Instrumentos tradicionales eléctricamente amplificados, el instrumento es como el acústico, pero la amplificación se realiza por medios electrónicos. Por ejemplo, el Clavecín eléctrico o la guitarra eléctrica.
- Instrumentos con generación eléctrica completa. Tanto la vibración inicial como la amplificación se realizan electrónicamente. El generador sonoro acústico (cuerda, etc) es sustituido por otro de tipo electrónico que produce una señal oscilatoria armónica.

Los electrófonos de generación eléctrica completa se dividen en 2 subgrupos [5]:

1. Instrumentos radio-eléctricos. Son los instrumentos en donde el sonido es producido totalmente a partir de oscilaciones eléctricas. Por ejemplo, Ondas Martenot, sintetizadores o samplers.
2. Instrumentos mecánico-eléctricos. Son los instrumentos musicales que mezclan elementos mecánicos y eléctricos para generar el sonido. Por ejemplo, Órgano Hammond o Melotrón.

Ahora bien todas las clases de instrumentos, salvo la electrónica, se mencionan en fuentes antiguas, como las inscripciones egipcias y la Biblia. El cuerpo humano, generando sonidos vocales y percusivos, fue, probablemente, el primer instrumento o quizá usó

instrumentos de percusión, tales como piedras, troncos huecos, y brazaletes, conchas y dientes de animales que pueden ser considerados como sacudidores. Se han encontrado flautas de hueso de treinta mil años de antigüedad en excavaciones arqueológicas.

El desarrollo de la música siempre ha estado vinculado al desarrollo tecnológico de cada época. Por ejemplo, el cambio de la música vocal a la instrumental ocurrido durante el siglo XVII, y su posterior desarrollo, tuvo un gran estímulo con el desarrollo y perfección logrados en la construcción de instrumentos de cuerdas por las familias italianas Stradivarius y Guarneri. Otro ejemplo que podemos mencionar es la invención en 1849, por el belga Adolfo Sax, de un instrumento de viento que desde ese momento aportó un nuevo sonido a la música: el saxofón.

De tal forma que la música electrónica es la música creada por medios electrónicos. Este concepto incluye la música compuesta con cintas magnetofónicas (que sólo existe sobre la cinta y se interpreta por medio de altavoces), la música electrónica en vivo (creada en tiempo real con sintetizadores y otros equipos electrónicos), la música concreta (creada a partir de sonidos grabados y luego modificados) y la música que combina el sonido de intérpretes en vivo con música electrónica grabada. Si bien estos tipos de música se refieren en principio a la naturaleza de la tecnología y las técnicas empleadas, estas divisiones son cada día menos claras. Hoy están en uso otras terminologías, como música por computadora, música electroacústica o música radiofónica, definiciones que suelen referirse más a la estética que a las tecnologías utilizadas [6].

El desarrollo tecnológico actual tiene su fundamento en la electricidad. Se considera que el primer instrumento musical que se basaba en la electricidad fue el Clavecín Eléctrico, que como parte esencial de su funcionamiento utilizaba electricidad estática. Este instrumento fue construido en 1759 por el jesuita, físico y matemático francés Jean-Baptiste de la Borde. El desarrollo de instrumentos musicales eléctricos ya se anticipaba desde el siglo pasado.

En la década de 1910, los futuristas italianos, liderados por el compositor Luigi Russolo, concibieron una música creada con ruidos y cajas de música electrónicas. Fue entonces cuando aparecieron los primeros instrumentos musicales electrónicos comerciales. No obstante, ciertos compositores visionarios, como Alexandre Skriabin o Henry Cowell, habían soñado una música creada solamente por medios electrónicos, aunque este tipo de música no fue un hecho real hasta el desarrollo de la tecnología de la grabación de sonidos durante la II Guerra Mundial [6].

El compositor e ingeniero francés Maurice Martenot, presentó en 1928 un instrumento electrónico llamado Ondas Martenot, consistente en un generador de bajas frecuencias manejado con un teclado de piano. En las décadas de 1940 y 1950, nacieron varios estudios de grabación asociados a los nombres de figuras destacadas, cuyos objetivos eran específicamente artísticos. En Francia, el ingeniero de sonido y compositor Pierre Schaeffer fundó el estudio de Radio Francia (RTF) en París, compuesto por varias grabadoras, micrófonos y equipos para la edición de cintas. Las principales técnicas utilizadas para crear música consistían en cortar, pegar, empalmar o recorrer hacia atrás distintos fragmentos de la cinta grabada. Estas técnicas de manipulación de las cintas dieron

lugar a un nuevo tipo de montaje de sonido, elaborado con mucho esfuerzo a partir de las grabaciones provenientes del mundo real. Schaeffer calificó los resultados como música concreta, un término que aún se sigue utilizando ampliamente, sobre todo en Francia. Su primer experimento en este nuevo género utilizaba sonidos de grabaciones de trenes. Todas sus obras de esa época eran breves estudios sobre sonidos y tienen títulos evocadores como la *Symphonie pour un homme seul* (1950), compuesta en colaboración con su colega (más joven que él) Pierre Henry. Los experimentos prácticos de Schaeffer en la composición de música electrónica se basaban en algunos escritos teóricos sobre el tema que influyeron en él. El estudio de Henry y Schaeffer atrajo a diversos compositores, entre los cuales destacaba Pierre Boulez.

Oliver Messiaen, utilizó en varias de sus obras el Ondas Martenott, combinando sus posibilidades sonoras con instrumentos tradicionales sinfónicos y hasta la fecha sigue componiendo obra basada en este instrumento [6], [7].

A finales de la década de 1940, Werner Meyer-Eppeler, físico y director del Instituto de Fonética de la Universidad de Bonn, presentó por vez primera el Vocoder, un dispositivo que conseguía sintetizar la voz humana. Su trabajo teórico influyó en los compositores relacionados con el estudio de la Radio de Alemania Occidental en Colonia (fundada en 1953), y cuyo interés giraba en torno a la síntesis electrónica de los sonidos mediante generadores de sonido y otros aparatos. El primer director del estudio de Colonia, Herbert Eimert, ejerció una gran influencia por su forma de utilizar el Serialismo total como base para la construcción de obras electrónicas. Según este método, todos los aspectos de la música, incluidos el tono, el ritmo y el volumen relativo, quedaban bajo el control de unos principios definidos numéricamente. Los sonidos y aparatos electrónicos brindaban la precisión y el control necesarios para la realización de este concepto. Compositores como Bruno Maderna y Karlheinz Stockhausen elaboraron piezas electrónicas cortas, llamadas síntesis aditiva. Estas obras se componían enteramente a partir de sonidos electrónicos.

A finales de la década de 1950, ya se habían establecido gran parte de los estudios de música electrónica de Europa. Los más importantes eran el de la RAI en Milán, fundado por Luciano Berio y Maderna, el Instituto de Sonología de Utrecht y el estudio EMS de Estocolmo. La división entre la música concreta y la música electrónica pura fue un fenómeno básicamente europeo. Si bien en EEUU surgieron varios estudios en aquella época, las distinciones estéticas eran allí menos importantes. En la década de 1950 en Nueva York, los compositores Otto Luening y Vladimir Ussachevski crearon músicas para cinta magnetofónica con un equipo de estudio muy rudimentario que transformaba los sonidos grabados de instrumentos y voces mediante técnicas de manipulación de cintas y simples unidades de reverberación. A finales de la década se asociaron con el Centro de Música Electrónica de Columbia - Princeton, donde el compositor Milton Babbitt empleaba una enorme computadora RCA para crear una música compuesta con los mismos principios seriales que Eimert y Stockhausen en Colonia. Su obra *Philomel* (1964) fue una de las primeras composiciones para soprano solista y cinta magnetofónica. El desarrollo de la tecnología informática en las décadas de 1950 y 1960 llevó al establecimiento de una serie de estudios dedicados en exclusiva a la música por ordenador en universidades estadounidenses y, en menor medida, en Europa.

Durante las décadas de 1960 y 1970, los estadounidenses Lansky y Barry Vercoe, entre otros, desarrollaron paquetes de programas para música (programas informáticos diseñados para la manipulación y creación de sonido), que se entregaban gratuitamente a los compositores interesados. Esta tradición de software procedente de las universidades de Estados Unidos ha contribuido en gran medida al crecimiento de la música por computadora en todo el mundo. Tanto el Centro para la Investigación Informática en Música y Acústica (CCRMA) de la Universidad de Stanford, en el estado de California, como el Institut Recherche et de Coordination Acoustique/Musique (IRCAM) de París (fundado por Pierre Boulez en 1976), utilizaron los ordenadores de forma significativa y continúan siendo centros influyentes en la composición de música electrónica en la actualidad.

El rápido desarrollo de la tecnología informática durante los últimos 15 años ha conducido a una revolución en la música por computadora y en la música electrónica en general. Los ordenadores son ahora más asequibles, y los programas informáticos, que antes eran muy lentos, funcionan ahora en cuestión de segundos o incluso de forma simultánea a los instrumentos. Son muchas las universidades y conservatorios actuales equipados con un estudio de música por ordenador y varios países tienen estudios nacionales, dedicados a la composición de música electrónica. Además, los músicos trabajan cada vez más de forma independiente, en estudios particulares [6].

El advenimiento de los algoritmos genéticos y su aplicación a la música ha permitido a algunos investigadores y compositores vanguardistas incursionar en la llamada programación evolutiva, en donde un tema musical propuesto puede evolucionar y formar nuevos elementos temáticos, lo cual ha creado una gran controversia entre los músicos tradicionalistas y los vanguardistas. Dicha controversia radica en que solo el ser humano es capaz de crear y no así una máquina.

1.3.1. LOS SINTETIZADORES Y LOS INSTRUMENTOS DE MÚSICA ELECTRÓNICA

A principios del siglo XX se inventaron instrumentos electrónicos como el Telharmonium cuyo principio de funcionamiento sería la base de lo que sería años más tarde el sintetizador electrónico o el Theremin, pero eran muy rudimentarios en su construcción y complicados en su funcionamiento en un principio y como de esto se trata esta tesis, no me adelantaré en seguirlo describiendo, conforme al avance de estas líneas se podrán dar cuenta de la aportación que se hace a la mejora del instrumento [6].

También en la década de los 30's se diseñaron y construyeron infinidad de instrumentos musicales electrónicos, destacando: el Órgano Hammond (1933), que fue el primer instrumento electrónico producido a gran escala, a partir de 1935, el Heliophon (1936) y el Clavioline (1947).

El sintetizador, es un aparato que genera y manipula sonidos por medios electrónicos. Con este artefacto se pueden crear nuevos sonidos así como reproducir los de los instrumentos musicales conocidos. La forma de la onda generada es alterada en su duración, altura y timbre mediante el uso de dispositivos tales como amplificadores, mezcladores, filtros, reverberadores, secuenciadores y moduladores de frecuencia. El primer sintetizador apareció en 1955. Fue desarrollado por Olsen y Belar en la Radio Corporation of America (RCA) en Princeton. Estaba destinado a investigar las propiedades del sonido, no a ser un instrumento musical. Varios compositores se interesaron en ampliar las posibilidades de producción de sonidos y en controlar todos los aspectos de la música gracias a esta herramienta. Entre ellos destacan Milton Babbitt, Marion Davido, Morton Subotnik y Charles Wuorinen. El sintetizador más conocido es el Moog, la primera máquina sencilla [6].

Entre 1960 y 1980 se fabricaron aparatos más reducidos que usaban microprocesadores, con lo que se hizo posible la grabación digital del sonido.

Los primeros sintetizadores, construidos a principios de los años 60, estaban hechos de componentes sueltos como osciladores (para proporcionar la materia prima para construir sonidos), filtros (para modificarlos suprimiendo armónicos), generadores de envolvente (que controlan cómo empieza y termina el sonido sintetizado), y circuitos electrónicos llamados secuenciadores (que prenden y apagan los sonidos de la composición en el momento adecuado). Estos sistemas modulares de composición estaban dirigidos sobre todo a los centros académicos de música electrónica que por esa época estaban surgiendo sobre todo en Europa y Estados Unidos.

Los sintetizadores e instrumentos electrónicos con amplificación, susceptibles de ser utilizados en las actuaciones en vivo, se perfeccionaron en la década de 1960. La computadora central RCA fue en esencia uno de los primeros sintetizadores electrónicos. El aparato contenía todos los recursos necesarios para generar y modificar los sonidos por medios electrónicos. En la década de 1960, Robert Moog, Donald Buchla y los fabricantes británicos EMS, entre otros, comenzaron a producir sintetizadores que utilizaban el control

de voltajes. Se trataba de un medio electrónico para variar el tono o el volumen, y se empleaba para crear, filtrar y modificar, sonidos electrónicos. Pronto aparecieron sintetizadores que ofrecían una serie de sonidos y efectos pregrabados. Éstos serían utilizados cada vez más con propósitos comerciales, desde los anuncios de la televisión hasta la banda sonora de películas de ciencia ficción. Hoy existe una amplia variedad de sintetizadores, capaces de imitar casi cualquier sonido y de crear gamas tonales y timbres completamente nuevos [6].

Los sintetizadores digitales musicales según la evolución histórica que han tenido los podemos clasificar en dos generaciones.

La primera generación estuvo basada en la síntesis musical mediante el uso de algoritmos que aproximan matemáticamente al modelo de un instrumento musical, lográndose así la síntesis aproximada de los sonidos.

Esta primera generación tuvo como limitante el tamaño de la palabra digital que tecnológicamente estaba disponible y por lo tanto las posibilidades polifónicas eran pobres.

La segunda generación se basó en el muestreo y grabación directa de la onda producida físicamente por el instrumento musical (samplers), lo cual dio una aproximación más exacta del sonido original, y se siguió manteniendo sin embargo las posibilidades de manipular el sonido grabado con fines meramente experimentales.

En la actualidad se sigue utilizando este tipo de sintetizadores y el advenimiento de las nuevas tecnologías ha permitido el uso de librerías virtuales más fieles a los sonidos de los instrumentos musicales y resulta a veces difícil distinguir entre un instrumento real y uno sintetizado.

1.3.2. MOOG

Robert Moog, es un inventor estadounidense, nacido en Flushing, Nueva York. Inventó el sintetizador musical.

Cuando era aún estudiante en el Queens College de Nueva York formó su propia empresa para producir un instrumento de teclado que reprodujera electrónicamente el sonido de cualquier instrumento musical. En colaboración con el compositor Herbert Deutsch introdujo el prototipo Sintetizador Moog en un congreso en 1964. Este instrumento tuvo tal demanda en todo el mundo que empezó a ser conocido como 'Moog'. Después trabajó con el compositor y virtuoso Walter Carlos, que llevó a cabo su grabación Switched on Bach en 1969, una prodigiosa muestra de virtuosismo electrónico interpretando música de J. S. Bach. y popularizó el sintetizador. El Polymoog, que podía producir acordes, apareció en 1976. Hoy día este instrumento se utiliza con frecuencia en la música popular, aunque muchos compositores de música electrónica también han explotado sus posibilidades [6].

El Moog modular es un sintetizador (instrumento musical electrónico) diseñado por Bob Moog, a quien debe su nombre.

Con el Moog era posible crear un número casi infinito de sonidos. Estos sonidos podían imitar los sonidos de los instrumentos convencionales o generar sonidos completamente nuevos.

El Moog modular era un sintetizador que se formaba a partir de la combinación de distintos módulos, por lo que su estructura no era siempre la misma. Cada módulo controlaba unas funciones determinadas. Como los módulos se vendían por separado, cada Moog se podía configurar según las necesidades específicas que cada músico demandaba. Las posibilidades eran casi infinitas y cada cual buscaba la configuración que deseada.

Cuando el primer Moog fue ensamblado, los distintos circuitos que generaban y filtraban el sonido se conectaban mediante cables que se podían intercambiar, como si se tratase de una centralita telefónica). El núcleo básico del sintetizador era el módulo con el filtro paso bajo.

Desde su primera versión, el Moog, siempre fue más barato que los prototipos que la RCA fabricaba como los MARK I y II (sintetizadores) [8].

1.3.3. MINIMOOG

El Minimoog es un sintetizador, por tanto, un instrumento musical electrónico. Se trata de una versión compacta del Moog modular, aparecida en 1971. Superaba su principal inconveniente, era compacto, resistente, con lo que se podía llevar a cualquier lugar. Además, al ser más pequeño, resultaba más económico que el Moog modular. El sonido del Minimoog se caracterizaba por sonido grueso y su timbre cálido y vibrante.

El diseñador del Minimoog, como del Moog, fue Bob Moog, a quien se debe el nombre.

Como todos los sintetizadores de su era, era monofónico (sólo sonaba una nota a la vez) y no disponía de memoria para guardar sonidos, debiendo ser creados o recreados en el momento. Nuevas series con modernas incorporaciones (MIDI, control digital, memorias, etc.) se producen en la actualidad bajo el nombre de "Minimoog Voyager" [9].



Figura 1.1. El Minimoog. [9]

1.3.4. BUCHLA

En 1963, Buchla, empezó a desarrollar lo que sería su primer sintetizador modular.

Este sintetizador no fue más que el primero de una interminable lista de instrumentos, tanto electrónicos como acústicos, con los que este personaje ha contribuido, desde una posición mucho más discreta que otros pioneros de la época, al desarrollo de la tecnología musical tal y como la conocemos actualmente.

La postura de Don Buchla ha sido la misma durante estos 30 años; sus diseños nunca han despertado el interés del mercado, pero los compositores más cercanos a los postulados vanguardistas lo adoran.

En cualquier caso, Buchla se ha ganado con todos los méritos un puesto en la historia de la música electrónica, y es necesario reconocer que la mayoría de sus diseños, si bien en el momento y forma en que fueron concebidos no resultaban demasiado rentables desde el punto de vista económico, introdujeron conceptos e ideas que se adelantaban espectacularmente en el tiempo.



Figura 1.2. Buchla y su sintetizador modular. [10]

Sintetizador modular modelo 100

Como el resto de los primeros modelos diseñados por Buchla, esta unidad no incorporaba teclado, o al menos no lo que identificamos habitualmente como "teclado". En su lugar, disponía de una serie de plaquitas sensibles al tacto, en ocasiones sensibles a la capacitancia de los dedos y en otras a la resistencia de los mismos, organizadas en diferentes configuraciones. Resulta interesante observar que, a pesar de lo insólito del sistema, estos

dispositivos de entrada ya eran sensibles a la presión con que eran pulsados, lo cual representa un interesante antecedente a los teclados sensitivos.

La unidad incorporaba dos "teclados"; el primero disponía de doce "teclas" que podía ser afinado a voluntad, y que ofrecía tres tensiones de control por tecla. El segundo "teclado" disponía de diez teclas, y sólo entregaba una tensión de control por cada tecla. Este segundo teclado se solía utilizar para generar tensiones de control, como podía ser la amplitud de las notas, o controlar loops generados por los secuenciadores. . Efectivamente, el sistema 100 estaba equipado con tres secuenciadores, dos de ellos de ocho pasos, y un tercero de 16 pasos. Cada paso de los secuenciadores entregaba hasta tres tensiones de control diferentes; era posible conectar en cascada dos secuenciadores, de forma que corrieran simultáneamente, entregando hasta seis tensiones de control por cada paso. De esta forma, por ejemplo, una tensión podía controlar el tono, otra la posición espacial de la nota, etc., mientras que otra podía utilizarse para controlar el generador de reloj del secuenciador, lo cual permitía ejercer un control total sobre el ritmo absoluto de la melodía. Esta curiosa prestación permitía la programación de estructuras rítmicas cíclicas de una complejidad pasmosa.

A pesar de lo que pueda pensar mucha gente, Buchla fue el verdadero inventor del secuenciador analógico y, como puede deducirse de la explicación anterior, sus primeros modelos estaban muy por encima de lo que posteriormente ofrecieron marcas con un éxito e incidencia comercial mucho más amplia.



Figura 1.3. Sintetizador modular modelo 100. [10]

Uno de los aspectos de los primeros sintetizadores analógicos que se criticaron con más frecuencia fue la manifiesta inestabilidad de algunos de sus componentes, especialmente en aquellos aspectos que afectaban a la estabilidad de afinación.

En el caso de Buchla, su creador consideró más importante dotar a sus prototipos de muchas más posibilidades que otros, aún a costa de sacrificar cierto grado de precisión. En cuanto a la estabilidad en la afinación, se siguió el criterio impuesto por Subotnick, que reclamó que la estabilidad en la afinación fuera la equivalente al tiempo que puede permanecer afinado un violín tradicional durante un concierto; según el criterio de este compositor, si era normal y estaba permitido reafinar un violín durante un concierto, por qué no un oscilador electrónico. Este sintetizador se mantuvo en producción entre 1963 y 1969.

Serie 200

Fabricado entre 1970 y 1978, este sistema modular se ofreció en multitud de configuraciones diferentes, incluida una que se consideraba portátil (incluía su propio flight-case). También en esta ocasión se ofreció la opción de incorporarle un teclado convencional, de tipo piano. En general, seguía la línea marcada por su predecesor.

Modelo touché

La década de los 80 empezó para Buchla lanzando un modelo que, una vez más, introducía conceptos que se adelantaban al estado del mercado de aquellos momentos. Efectivamente, sacando el máximo provecho posible a la emergente tecnología informática de aquellos momentos, Buchla lanzó el modelo Touché. Se trataba de un sintetizador que incorporaba 24 osciladores, ocho voces y un teclado capaz de hacer splits y layers; lástima que sólo se llegaron a fabricar cuatro o cinco unidades de este modelo.

El objetivo de su creador es que el Touché fuera la herramienta definitiva para la interpretación en directo, y para ello introdujo prestaciones incomparables, como la posibilidad de grabar patrones, riffs o elementos rítmicos en tiempo real, que posteriormente podían ser reproducidos durante la interpretación.

La posibilidad de split (partición de teclado) era capaz de seguir el movimiento de la mano, de forma que era posible cambiar de forma suave y dinámica de un instrumento a otro; para ello, se contaba con controles adecuados para definir los tiempos y modos de cambio de un sonido a otro (fade), lo cual permitía seleccionar un nuevo instrumento y continuar tocando, mientras el instrumento se transformaba de forma gradual en el nuevo instrumento.

Serie 400

En la serie 400 Buchla integró toda una serie de elementos dispares, pero ofreciendo por primera vez la posibilidad de que un sistema de estas características no se viera limitado únicamente por su hardware, ya que había la posibilidad de que el sistema fuera gobernado por un ordenador. En realidad, el sistema 400 sacaba partido a mucha de la tecnología implicada en el anterior modelo Touché. El sistema era totalmente operativo en 1983.

Buchla, hoy

En la actualidad, esta firma sigue en activo, y ofrece un único producto, se trata del curioso controlador denominado LIGHTNING II.

LIGHTNING II es un controlador MIDI (Interfase Digital para Instrumentos Musicales) especializado que detecta la posición y movimiento de unos guantes que se coloca el intérprete, y transforma esa información en mensajes MIDI capaces de controlar cualquier módulo de sonido o dispositivo MIDI. Además de esa función como controlador MIDI especializado, Lightning II incorpora su propio sintetizador de 32 voces, lo que lo convierte en un instrumento autosuficiente.

Básicamente, LIGHTNING II detecta la posición vertical y horizontal de cada mano, dentro de cuatro coordenadas completamente independientes. A partir de esta información, el procesador digital de señal que incorpora el equipo interpreta la velocidad instantánea y la aceleración y realiza un análisis completo del gesto.

La información interpretada por el LIGHTNING II puede ser asignada a diferentes controladores MIDI sobre varios canales simultáneamente, de forma que es posible convertir esa información en notas y todo tipo de parámetros controladores del sonido.

El usuario también tiene la posibilidad de definir diferentes escalas musicales y tablas de afinación; la sucesión de tonos puede configurarse a voluntad, así como la amplitud de las zonas que corresponden a cada nota. También puede transmitir reloj MIDI, de forma que sea posible controlar la velocidad de secuenciadores y cajas de ritmo en función de los movimientos que realicemos [10].



Figura 1.4. LIGHTNING II. [10]

Algunas diferencias entre el Moog, Minimoog y Buchla

El Moog y el Minimoog, son sintetizadores analógicos, además al Moog se le adicionaba un teclado, con lo cual se le podía manipular.

Por otro lado el Buchla, se componía de 6 osciladores senoidal, 6 osciladores diente de sierra, 6 osciladores de onda cuadrada, mezcladores de onda analógicos y modificadores de envolvente. Para el Buchla no había concepto de melodía en si mismo, dado que su tecnología estaba pensada para crear ambientes sonoros diferentes a los que estamos acostumbrados. Siendo estos aprovechados dentro de películas y por los experimentadores vanguardistas dentro de la música.

En otro orden de ideas los nuevos instrumentos musicales electrónicos de la década de 1960 también contribuyeron al crecimiento de la música electrónica en vivo, en la cual la creación y la manipulación electrónica del sonido ocurren en tiempo real (es decir, que se tocan y se escuchan en el momento). Al mismo tiempo, ciertos compositores creaban música electrónica en vivo con equipos fabricados por ellos mismos. La obra Solo de Stockhausen (1966), por ejemplo, utilizaba un complejo sistema de retardo de la grabación que requería cuatro ayudantes para su interpretación. Durante la década de 1960, surgieron numerosos grupos de música electrónica en vivo, muchos de los cuales interpretaban música improvisada de carácter teatral. En años recientes, la revolución tecnológica también se tradujo en la aparición de numerosos instrumentos electrónicos y programas de ordenador diseñados para las actuaciones en directo. Destacan, entre otros, el músico y diseñador de instrumentos Michel Waisvisz, que trabaja en los estudios STEIM de Holanda, inventor de diversos instrumentos como el llamado The Hands (las manos), en el cual el movimiento de las manos del intérprete en el espacio activa sonidos complejos y permite manipulaciones sonoras. El compositor-intérprete estadounidense George Lewis combina la improvisación al trombón con programas personalizados de composición por ordenador que trabajan de forma simultánea. La estación de trabajo de procesamiento de señales desarrollada en el IRCAM permite una manipulación muy compleja en tiempo real de los sonidos instrumentales y sintéticos. Además de estos interfaces (superficie de contacto) altamente especializados, muchos compositores e intérpretes utilizan algunas cajas de efectos disponibles en el mercado, diseñadas pensando en el mercado de la música popular.

La evolución del sintetizador y la guitarra eléctrica a partir de la década de 1950 también ha influido en gran medida en el mundo del pop y la música rock.

En la década de 1980 se inventó una nueva forma de trabajar con instrumentos de música electrónica dirigida al mercado popular: el llamado MIDI (interfaz digital de instrumentos musicales). El MIDI no es un instrumento musical, sino un medio que permite a los sintetizadores y otros equipos comunicarse unos con otros mediante el envío de instrucciones digitales. Por ejemplo, mediante el MIDI un intérprete puede hacer que varios sintetizadores respondan cuando se toque uno de ellos. El uso de la tecnología musical en la música pop y rock es tan importante que la naturaleza de los instrumentos y las otras tecnologías musicales relacionadas con la música electrónica están dictadas, en gran medida,

por las necesidades de la industria del rock. Además cabe destacar que esto no sólo ha beneficiado al rock, sino que también a otro tipo de géneros musicales, tales como jazz, swing por mencionar algunos e incluso a para arreglos de algunas piezas de conciertos de música clásica [6].

1.4. DIAGRAMA DE BLOQUES DE UN SINTETIZADOR POR HARDWARE ELEMENTAL

El teclado controlador

El teclado es el controlador más popular para todo tipo de sintetizadores. Se trata de una fuente que suministra tensiones proporcionales a la tecla que se oprime. En los sintetizadores polifónicos, más modernos, se obtienen varias tensiones de control, una para cada tecla pulsada. El teclado también puede proporcionar otras tensiones de control, como puede ser la velocidad de pulsación de la tecla, o la presión sobre la misma, lo que permite aumentar las posibilidades expresivas del sintetizador.

Para el funcionamiento del conjunto de módulos, el teclado suministra varias señales de control:

- Un impulso de disparo cada vez que se oprime una tecla, que se utiliza para iniciar los generadores de envolvente.
- Un escalón de tensión de duración igual al tiempo que la tecla está pulsada, que permite generar las señales de mantenimiento y relajación del generador ADSR.
- Un nivel de tensión discreto correspondiente a la tecla pulsada, de tal modo que 1V corresponde a una diferencia de una octava.
- Una señal de control de portamento, que permite generar un deslizamiento entre dos notas.
- Una memoria analógica que mantiene la tensión de la última tecla oprimida.

El oscilador controlado por tensión, VCO

Se trata de la fuente de sonido básica del sintetizador. La frecuencia de la señal que genera se obtiene como suma algebraica de las tensiones de control aplicadas a sus entradas. En los sintetizadores se usan VCO con respuesta exponencial normalizada a 1V/octava. Dispone de varias formas de onda, senoidal, diente de sierra, triangular y cuadrada, y cada una de ellas dará un tipo de sonido peculiar y diferente, dado el distinto contenido armónico que tienen.

El mezclador de audio

Es un dispositivo que permite la combinación de una serie de señales de entrada en las proporciones deseadas de modo que se obtiene otra señal que es la combinación de aquellas.

El generador de envolventes, ADSR

ADSR es el acrónimo de attack, decay, sustain release, y se refiere a cuatro zonas características de la amplitud de una señal de control. Las zonas ADR se refieren a tiempos, mientras que S se refiere a un nivel. Estos generadores de envolvente controlarán tanto la amplitud de la señal sonora como la respuesta del filtro.

El amplificador controlado por tensión, VCA

La combinación de un generador ADSR y un VCA imparte a los sonidos que modifica unas determinadas características de ataque y decaimiento en su intensidad, de modo que permite la simulación de la dinámica de diversos instrumentos. También es posible modular en amplitud la señal que se procesa para añadir trémolos y otros efectos que dependerán mucho de la frecuencia de la señal moduladora.

El filtro controlado por tensión, VCF

El filtro es un elemento que permite el paso de unas determinadas frecuencias de la señal de entrada, y es el equivalente eléctrico de los resonadores acústicos de los instrumentos tradicionales, como la caja del violín, que da el sonido característico del instrumento. Este filtro se puede configurar en varios modos, paso bajo, paso alto, paso de banda y rechazo de banda, según sea el efecto que deseamos resaltar o atenuar de la onda básica generada por los VCO.

Los parámetros característicos de un filtro son la frecuencia de corte, a partir de la cual la respuesta es la mitad, y la pendiente del filtro, que determina la atenuación de los sucesivos armónicos de la señal de entrada.

El oscilador de baja frecuencia, LFO

Generalmente se compone de varios osciladores independientes, controlados por tensión, con varias formas de onda cada uno, y que se sitúan en la banda de bajas frecuencias, hasta unos 20Hz. Su salida puede usarse tanto para controlar la amplitud, trémolo, como la frecuencia, vibrato, de los VCO, obteniendo sonoridades muy variadas.

El generador de ruido

Suelen proporcionar ruido blanco y rosa, así como una salida de tensión de control aleatoria, pero ajustable en ciertos parámetros. Estas tres señales sirven para producir sonidos no convencionales similares a los naturales, desde el mar al trueno y el siseo del aire [11].

1.5. EL THEREMIN

El Theremin, llamado en su versión primitiva aetherophone (eterófono), es uno de los primeros instrumentos musicales electrónicos, y el primer sintetizador de la Historia, inventado en 1919 por el físico ruso Lev Serguéievich Termen (quien luego afrancesó su nombre a León Thérémin).

Cómo funciona el Theremin

El Theremin, consta de dos antenas que crean un campo electromagnético cada una, el intérprete actúa como conductor eléctrico sin llegar en ningún momento a entrar en contacto físico con el Theremin; al acercar la mano derecha a la antena vertical aumenta la frecuencia del sonido y al alejarla vamos consiguiendo notas cada vez más graves, la antena horizontal es la encargada de dar las distintas intensidades al sonido, la conjunción de intensidades y tonos bien medidos da como resultado la música tal y como la conocemos en cualquier otro instrumento musical.

Originalmente, su versión más primitiva fue llamada Aetherophone (se podría traducir como Eterófono), y constaba sólo de la antena de tono. Dicho diseño fue tempranamente mejorado por el inventor, añadiendo posteriormente una antena para controlar el volumen tal y como hemos descrito. Actualmente, algunos de los modelos caseros y comercializados de Theremin disponen tan sólo de la antena que controla el tono, lo cual siendo rigurosos les convierte en realidad en un "Eterófono", y su uso frecuentemente es el de un aparato para efectos especiales más que un instrumento musical, al no poder acentuar ni separar las notas producidas.

Su sonido

Originalmente, el timbre de los Theremins se asemejaba a algo entre un violonchelo y una voz humana: no en vano su inventor, además de físico, era violonchelista. Hoy en día existen incluso modelos que participan de la tecnología MIDI, lo cual les posibilita tener, virtualmente, cualquier timbre que se desee utilizando un sampler o muestreador. Actualmente existen multitud de firmas que comercializan versiones transistorizadas de theremines (los originales estaban fabricados con las, entonces novedosas, válvulas de vacío). Posiblemente la casa Big Briar (antes Moog) sea la que goza de mayor reconocimiento en la fabricación de theremines, y su modelo Etherwave Standard se puede calificar como el estándar de calidad de sonido en el mercado actual de theremines a transistores.

Intervalo armónico y posibilidades de ejecución

Le pertenece un gran intervalo armónico, el intérprete puede escoger tocar en una tesitura natural para las melodías a un registro grave para los bajos o todo lo musicalmente agudo que quiera. Este instrumento monófono es eminentemente melódico, con un sonido característicamente vibrado y glissado, realmente es el instrumento electrónico más orgánico que hay. El sonido del Theremin es fácilmente reconocible por el gran público y un tanto inconscientemente por su continua aparición en películas de terror y ciencia-ficción de los años 40's, 50's y 60's e incluso en algunas películas actuales. Aunque su vertiente clásica no es tan conocida y eso que realmente fue para lo que se inventó, aquí el Theremin alcanza su mayor variedad sonora, plenitud interpretativa y expresividad.



Figura 1.5. León Thérémin tocando el Theremin. [12]

Aprendizaje

Aprender a tocar este instrumento es difícil, empezando porque hay muy pocos profesores a nivel mundial dedicados a la ejecución del mismo y por ende a la impartición de clases, la mayoría de los Thereministas son autodidactas aunque actualmente podemos disfrutar de algunos videos pedagógicos que son vitales para conocer la técnica de ejecución; se requiere de mucha disciplina, dedicación y esfuerzo, el mismo o más que el que aplicaríamos para dominar un violín o un violonchelo. Hasta los años 30's, se tocaba de una

forma un tanto lógica, pero pronto Clara Rockmore (virtuosa del Theremin) revolucionó totalmente la técnica aplicando lo que ella denominó la digitación aérea, con lo que las interpretaciones llegaron a unos grandes niveles de fluidez y virtuosismo.

En gran medida el desarrollo de este proyecto se debe a la contribución que se haría para incrementar el número de profesores y personas deseosas de aprender la ejecución de este instrumento y no simplemente a base de videos pedagógicos, sino meramente con el instrumento [12], [13].

1.6. PLANTEAMIENTO DEL PROBLEMA DE DISEÑO DE UN THEREMIN DIGITAL

El diseño planteado por León Theremin originalmente en la actualidad no es posible realizarlo literalmente debido a que el campo radial se encuentra totalmente saturado, lo cual implica el buscar zonas radiales que todavía se encuentren vírgenes (Ver apéndice 1)*.

En el diseño inicial el intervalo de radiofrecuencias a las que trabajaba el Theremin estaba ubicado entre 200 KHz y 600 KHz, pero en la actualidad estas frecuencias se encuentran saturadas por los subarmónicos provenientes de las señales de la banda de AM, lo cual crea el problema de que se metan sonidos de las estaciones de radio y por lo tanto podemos decir que el diseño original en la actualidad es irrealizable, no obstante que la tecnología actual ha mejorado enormemente comparativamente con la época en que se diseñó el Theremin.

Los Theremins actuales han cambiado de intervalo de radiofrecuencias y sin embargo se sigue teniendo el problema de contaminación por ruido de las emisoras de radio.

Hay que recordar que un Theremin trabaja con base en dos osciladores de radiofrecuencia, donde uno de los osciladores es de frecuencia fija y el otro es de frecuencia variable, en donde la antena del segundo oscilador forma parte integral del mismo, sucediendo el fenómeno de que al acercar un objeto ferromagnético e incluso una mano puede provocar una variación en la frecuencia de oscilación de este oscilador.

Al sumarse las señales provenientes de los dos osciladores se generan armónicos de alta frecuencia y armónicos de baja frecuencia, siendo estos últimos los útiles para el Theremin, dado que la frecuencias bajas son las audibles. Cabe aclarar que un Theremin funciona en base al batimento de dos señales, de las cuales se obtiene la señal audible.

El Theremin completo está integrado por dos sistemas independientes de osciladores, dentro de los cuales uno regula la afinación y otro regula la intensidad, teniendo cada uno su oscilador fijo y su oscilador variable que es controlado específicamente por las manos.

El planteamiento de una nueva versión de Theremin que se pretende diseñar estará basado en un procesador digital de señales que se utilizará como generador de una onda senoidal cuya frecuencia será modulada para producir un sonido de frecuencia variable.

Para controlar la frecuencia se usará un transductor ultrasónico con una frecuencia de 40 KHz que será capaz de detectar la presencia de un objeto así como su distancia. La frecuencia del generador de onda senoidal será proporcional a la distancia detectada, siendo en este caso la presencia de la mano el objeto a detectar.

Como se requiere de un generador de onda senoidal variable y un control de volumen también variable, se hará uso de dos procesadores digitales de señales. Uno de los procesadores estará dedicado a la modulación de la frecuencia y el otro a la modulación de la amplitud dando de esta manera la emulación del funcionamiento que tenía el Theremin propuesto originalmente.

La innovación tecnológica aplicada a este nuevo Theremin radica en el uso de procesadores digitales de señales, el uso de dos sistemas de sonar y la electrónica de estado sólido adicional utilizada en los elementos periféricos, así como su costo extremadamente bajo y por supuesto la precisión de todos estos elementos. Sin embargo, el proceso de manejo e interpretación de obra musicales por medio de este instrumento serán las mismas que planteó León Theremin en su diseño original en 1919.

Necesariamente el diseño involucrará una parte de hardware y una parte de software de las cuales se hará una descripción más adelante.

CAPÍTULO II

EL PROCESADOR DIGITAL DE SEÑALES dsPIC30

En este capítulo se describirá la arquitectura de la CPU, de la memoria y de los periféricos internos que configuran a este procesador digital de señales.

En primer lugar se hará un análisis de la arquitectura en donde se hablará de su característica RISC, de su modelo Harvard, de los registros de la CPU y de la organización de la memoria interna.

Posteriormente se hará una descripción paso a paso de cada uno de los periféricos que integran a este procesador.

2.1. ARQUITECTURA

2.1.1. RISC (Reduced Instruction Set Computer)

La arquitectura RISC, debe cumplir con las siguientes características fundamentales:

1. Instrucciones de tamaño fijo y presentadas en un reducido número de formatos.
2. Sólo las instrucciones de carga y almacenamiento acceden a la memoria por datos.

Además estos procesadores suelen disponer de muchos registros de propósito general.

El objetivo de diseñar máquinas con esta arquitectura es posibilitar la segmentación y el paralelismo en la ejecución de instrucciones y reducir los accesos a memoria.

Las máquinas RISC protagonizan la tendencia actual de construcción de microprocesadores. Power PC, DEC Alpha, MIPS, ARM... son ejemplos de algunos de ellos.

RISC es una filosofía de diseño de CPU para computadora que ésta a favor de conjuntos de instrucciones pequeños y simples que toman menor tiempo para ejecutarse. El tipo de procesador más comúnmente utilizado en equipos de escritorio, el x86, está basado en CISC en lugar de RISC, aunque las versiones más nuevas traducen instrucciones basadas en CISC x86 a instrucciones más simples basadas en RISC para uso interno antes de su ejecución [14].

La idea fue inspirada por el hecho de que muchas de las características que eran incluidas en los diseños tradicionales de CPU para aumentar la velocidad estaban siendo ignoradas por los programas que eran ejecutados en ellas. Además, la velocidad del procesador en relación con la memoria de la computadora que accedía era cada vez más alta. Esto conllevó la aparición de numerosas técnicas para reducir el procesamiento dentro del CPU, así como de reducir el número total de accesos a memoria.

Terminología más moderna se refiere a esos diseños como arquitecturas de carga-almacenamiento.

Filosofía de diseño RISC

Los diseños RISC cuentan con una vasta cantidad de conjuntos de instrucciones para ellos. La verdadera diferencia es la filosofía para hacer todo en registros y llamar y guardar los datos hacia ellos y en ellos mismos. Ésta es la razón por la que la forma más correcta de denominar este diseño es cargar-almacenar. Con el paso del tiempo las técnicas de diseño antiguas se dieron a conocer como Computadora con Conjunto de Instrucciones Complejo, CISC por sus siglas en inglés, aunque esto fue solamente para darles un nombre diferente por razones de comparación.

Por esto la filosofía RISC fue crear instrucciones pequeñas, implicando que había pocas, de ahí el nombre conjunto de instrucciones reducido. El código fue implementado como series de esas instrucciones simples, en vez de una sola instrucción compleja que diera el mismo resultado. Esto hizo posible tener más espacio dentro de la instrucción para transportar datos, resultando esto en la necesidad de menos registros en la memoria. Al mismo tiempo la interfaz con la memoria era considerablemente simple, permitiendo ser optimizada.

Sin embargo RISC también tenía sus desventajas, debido a que una serie de instrucciones son necesarias para completar incluso las tareas más sencillas, el número total de instrucciones para la lectura de la memoria es más grande, y por lo tanto lleva más tiempo. Asimismo no estaba claro donde habría o no una ganancia neta en el desempeño debido a esta limitación, y hubo una batalla casi continua en el mundo del diseño sobre los conceptos de RISC.

Los primeros diseños de RISC ofrecían una mejora de rendimiento muy pequeña, pero fueron capaces de añadir nuevas características y para finales de los ochenta habían dejado totalmente atrás a sus contrapartes CISC. Con el tiempo esto pudo ser dirigido como una mejora de proceso al punto en el que todo esto pudo ser añadido a los diseños CISC y aún así caber en un solo chip, pero esto tomó prácticamente una década entre finales de los ochenta y principios de los noventa.

En pocas palabras esto significa que para cualquier nivel de desempeño dado, un chip RISC típicamente tendrá menos transistores dedicados a la lógica principal. Esto permite a los diseñadores una flexibilidad considerable; así pueden, por ejemplo:

- Incrementar el tamaño del conjunto de registros
- Implementar medidas para aumentar el paralelismo interno
- Añadir más cachés, los cuales son datos duplicados de otros originales, con la propiedad de que los datos originales son difíciles de acceder, normalmente en tiempo, respecto a la copia en el caché, por lo que al tener acceso por primera vez a un dato, se hace una copia en el caché; los accesos siguientes se realizan a dicha copia, haciendo que el tiempo en acceder al dato sea menor.
- Añadir otras funcionalidades, como E/S y relojes para minicontroladores
- Construir los chips en líneas de producción antiguas que de otra manera no serían utilizables

Las características que generalmente son encontradas en los diseños RISC son:

- codificación uniforme de instrucciones (ejemplo: el código de operación se encuentra siempre en la misma posición de bit en cada instrucción, la cual es siempre una palabra), lo que permite una decodificación más rápida;
- un conjunto de registros homogéneo, permitiendo que cualquier registro sea utilizado en cualquier contexto y así simplificar el diseño del compilador (aunque existen muchas formas de separar los ficheros de registro de entero y coma flotante);
- modos de direccionamiento simple con modos más complejos reemplazados por secuencias de instrucciones aritméticas simples;
- algunos tipos de datos soportados en el hardware (por ejemplo, algunas máquinas CISC tiene instrucciones para tratar con tipos byte, cadena; tales instrucciones no se encuentran en una máquina RISC).

Los diseños RISC también prefieren utilizar como característica un modelo de memoria Harvard, donde los conjuntos de instrucciones y los conjuntos de datos están conceptualmente separados; esto significa que el modificar las direcciones donde el código se encuentra pudiera no tener efecto alguno en las instrucciones ejecutadas por el procesador (porque la CPU tiene separada la instrucción y el caché de datos, al menos mientras una instrucción especial de sincronización es utilizada). Por otra parte, esto permite que ambos caches sean accedidos separadamente, lo que puede en algunas ocasiones mejorar el rendimiento.

Muchos de esos diseños RISC anteriores también compartían una característica no muy amable, el slot de salto retardado. Un slot de salto retardado es un espacio de instrucción siguiendo inmediatamente un salto. La instrucción en este espacio es ejecutada aun si el salto se produce o no (en otras palabra el salto es retardado). Esta instrucción mantiene la ALU de la CPU ocupada por el tiempo extra normalmente necesario para ejecutar una brecha. En nuestros días el slot de salto retardado se considera un desafortunado efecto colateral de la estrategia particular por implementar algunos diseños

RISC. Es por esto, que los diseños modernos de RISC, tales como PowerPC, y versiones más recientes de SPARC y de MIPS, generalmente eliminan esta característica.

El RISC moderno

La técnica de diseño RISC ofrece poder incluso en medidas pequeñas, y esto ha venido a dominar completamente el mercado de CPU integrados de bajo consumo de energía. Los CPU integrados son por mucho los procesadores más comunes en el mercado: considera que una familia completa con una o dos computadoras personales puede poseer varias docenas de dispositivos con procesadores integrados. RISC se ha apoderado completamente del mercado de estaciones de trabajo. Después del lanzamiento de la SUN SPARCstation los otros proveedores se apuraron a competir con sus propias soluciones basadas en RISC. Incluso el mundo de las computadoras centrales está ahora basado completamente en RISC.

Esto es sorprendente en vista del dominio del Intel x86 en el mercado de las computadoras personales de escritorio y el mercado de servidores de la gama baja. Aunque RISC fue capaz de avanzar en velocidad muy rápida y económicamente.

Los diseños RISC han llevado a un gran número de plataformas y arquitecturas al éxito, algunas de las más grandes son:

1. La línea MIPS Technologies Inc., que se encuentra en la mayoría de las computadoras de SGI, en la Nintendo 64 y PlayStation.
2. La serie IBM POWER, utilizado principalmente en Servidores de IBM.
3. La versión PowerPC de Motorola e IBM (una versión de la serie IBM POWER) utilizada en los ordenadores Apple Macintosh como el iMac, eMac, Power Mac y posteriores.
4. El procesador SPARC de SUN Microsystems y el UltraSPARC, que se encuentra en todos sus últimos modelos de equipos.
5. El PA-RISC y el HP/PA de Hewlett-Packard.
6. El DEC Alphaservidores HP AlphaServer.
7. El ARM – Se encuentra en dispositivos PALM, Nintendo DS, Game Boy Advance y en múltiples PDAs [14].

2.1.2. ARQUITECTURA HARVARD

El término Arquitectura Harvard originalmente se refería a las arquitecturas de computadoras que utilizaban dispositivos de almacenamiento físicamente separados para las instrucciones y para los datos (en oposición a la Arquitectura Von Neumann). El término proviene de la computadora Harvard Mark I, que almacenaba las instrucciones en cintas perforadas y los datos en interruptores.

Todas las computadoras constan principalmente de dos partes, la CPU que procesa los datos, y la memoria que guarda los datos. Cuando hablamos de memoria manejamos dos parámetros, los datos en sí, y el lugar donde se encuentran almacenados (o dirección). Los dos son importantes para la CPU, pues muchas instrucciones frecuentes se traducen a algo así como "coge los datos de ésta dirección y añádelos a los datos de ésta otra dirección", sin saber en realidad qué es lo que contienen los datos.

En los últimos años la velocidad de las CPUs ha aumentado mucho en comparación a la de las memorias con las que trabaja, así que se debe poner mucha atención en reducir el número de veces que se accede a ella para mantener el rendimiento. Si, por ejemplo, cada instrucción ejecutada en la CPU requiere un acceso a la memoria, no se gana nada incrementando la velocidad de la CPU - este problema es conocido como 'limitación de memoria'.

Se puede fabricar memoria mucho más rápida, pero a costa de un precio muy alto. La solución, por tanto, es proporcionar una pequeña cantidad de memoria muy rápida conocida con el nombre de caché. Mientras los datos que necesita el procesador estén en la caché, el rendimiento será mucho mayor que si la caché tiene que obtener primero los datos de la memoria principal. La optimización de la caché es un tema muy importante de cara al diseño de computadoras.

La arquitectura Harvard ofrece una solución particular a este problema. Las instrucciones y los datos se almacenan en cachés separadas para mejorar el rendimiento. Por otro lado, tiene el inconveniente de tener que dividir la cantidad de caché entre los dos, por lo que funciona mejor sólo cuando la frecuencia de lectura de instrucciones y de datos es aproximadamente la misma. Esta arquitectura suele utilizarse en DSPs, o procesador de señal digital, usados habitualmente en productos para procesamiento de audio y video.

Cabe mencionar que tradicionalmente los microprocesadores se basan en la estructura de Von Neumann, como la de la figura 2.1, que se caracteriza por disponer de una única memoria principal en la que se almacenan los datos y las instrucciones. A esta memoria se accede a través de un sistema de buses único:

- Bus de datos
- Bus de direcciones
- Bus de control

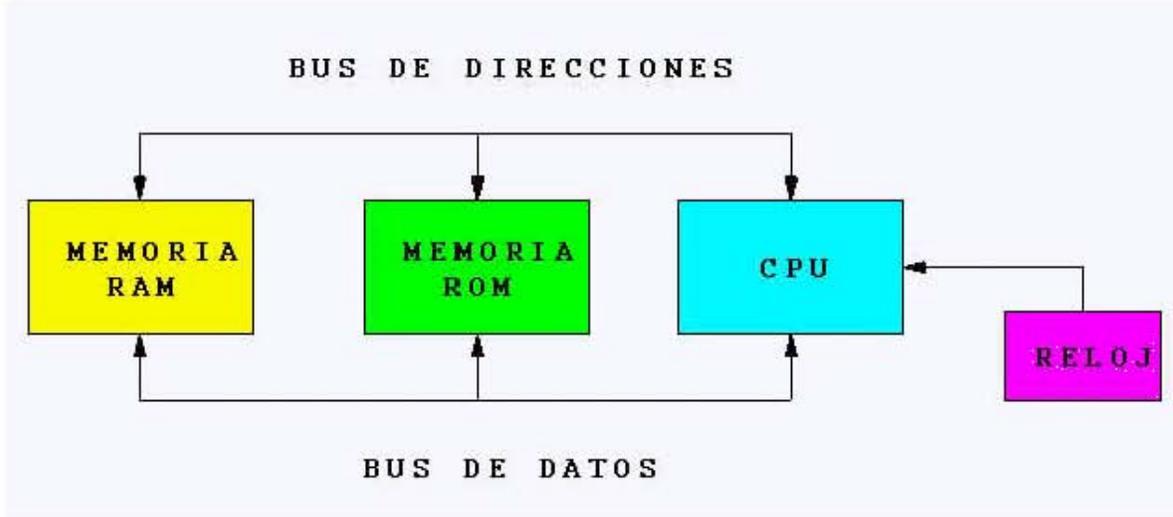


Figura 2.1. Arquitectura según el modelo de Von Neumann. [16]

El modelo Harvard, representado en la figura siguiente, dispone de dos memorias:

- Memoria de datos
- Memoria de Programa

Además cada memoria dispone de su respectivo bus, lo que permite, que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos y a la de instrucciones. Como los buses son independientes éstos pueden tener distintos contenidos en la misma dirección [15], [16].

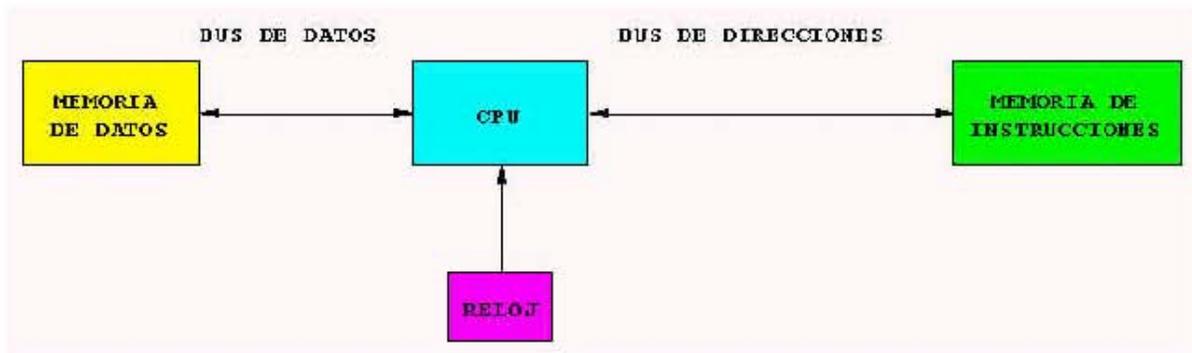


Figura 2.2. Arquitectura según el modelo Harvard. [16]

2.1.3. CARACTERÍSTICAS PRINCIPALES DE UN DSP

DSP, Digital Signal Processor, que significa Procesador Digital de Señal. Un DSP es un sistema basado en un procesador o microprocesador que posee un juego de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto es especialmente útil para el procesamiento y representación de señales analógicas en tiempo real: en un sistema que trabaje de esta forma (tiempo real) se reciben muestras (samples), normalmente provenientes de un conversor analógico/digital (ADC).

Puede trabajar con señales analógicas, pero es un sistema digital, por lo tanto necesitará un conversor analógico/digital a su entrada y digital/analógico en la salida.

Como todo sistema basado en un procesador programable necesita una memoria donde almacenar los datos con los que trabajará y el programa que ejecutará.

Si se tiene en cuenta que un DSP puede trabajar con varios datos en paralelo y un diseño e instrucciones específicas para el procesamiento digital, nos podemos dar una idea de su enorme potencial para este tipo de aplicaciones. Estas características constituyen la principal diferencia de un DSP y otros tipos de procesadores.

Un DSP, es un microprocesador que está optimizado internamente para realizar los cálculos necesarios para implementar algoritmos de procesamiento de señal. Ésta optimización se consigue mediante algunos aspectos principales:

- implementación de operaciones por hardware
- instrucciones poco comunes que ejecutan varias operaciones en un solo ciclo
- modos de direccionamiento especiales
- memoria de programa “ancha”, con más de 8 bits

Las aplicaciones más habituales en las que se emplean DSP son el procesamiento de audio y video; y cualquier otra aplicación que requiera el procesamiento en tiempo real. Con estas aplicaciones se puede eliminar el eco en las líneas de comunicaciones, lograr hacer más claras imágenes de órganos internos en los equipos de diagnóstico médico, cifrar conversaciones en teléfonos celulares para mantener privacidad, analizar datos sísmicos para encontrar nuevas reservas de petróleo, hacer posible las comunicaciones wireless LAN, el reconocimiento de voz, los reproductores digitales de audio, los modems inalámbricos, las cámaras digitales, etc.

Los modos de direccionamiento especiales se implementan para agilizar el tratamiento de estructuras de datos (como por ejemplo, buffers circulares) similares a las que conocemos de lenguajes de programación de nivel medio-alto. De este modo se han implementado algunos modos de direccionamiento tales como:

- Direccionamiento circular
- Direccionamiento con bits invertidos
- Pre y post-modificación de los punteros de direcciones (similar al $n++$, $++n$, $n--$ y $--n$ del lenguaje C)

El uso de juego de instrucciones con palabras "anchas" nos permite codificar en una sola palabra datos e instrucciones, disminuyendo de este modo el número de accesos a memoria por instrucción y aumentando así el rendimiento del sistema. Este método se usa también en muchos microcontroladores actuales, como la familia PIC.

Arquitectura

Un DSP está diseñado tomando en cuenta las tareas más habituales del procesado digital: sumas, multiplicaciones y retrasos (almacenar en memoria).

Tienen arquitectura Harvard y algunos pueden manejar números con punto flotante.

Los elementos básicos que componen un DSP son:

- Conversores en las entradas y salidas
- Memoria de datos, memoria de programa y DMA
- MACs: multiplicadores y acumuladores
- ALU: Unidad aritmético-lógica
- Registros

Normalmente en los DSP, se usa una arquitectura Harvard modificada con 3 buses: uno de programa y dos de datos, lo cual permite que la CPU lea una instrucción y dos operandos a la vez (pero no dos posiciones de memoria a la vez, para lo que hace falta una RAM de doble puerto). En el proceso de señales, las operaciones con 2 operandos son muy comunes, motivo por el cual se hace esta modificación.

Variaciones en la arquitectura de los DSP

Una característica bastante común de los DSP, es su adaptación a diversas aplicaciones por parte de los fabricantes, añadiendo diversas características adicionales y periféricos, de modo que un tipo determinado de DSP, sea la respuesta casi perfecta a una necesidad concreta. Así, consultando los catálogos, encontraremos unos DSP, que se parecen bastante a un microprocesador avanzado, y otros que nos recuerdan a algunos microcontroladores de gama alta. En los DSP, dedicados a tareas de control, podemos encontrar los periféricos típicos de los microcontroladores, y otros no tan comunes. En concreto, podemos hallar periféricos tales como:

- Puertos de entrada / salida
- Patillas de interrupción externa
- Unidades de comunicación serie (RS 232)
- Temporizadores
- Contadores
- Bucles enganchados en fase (PLL)
- Buses I2C y/o SPI
- Convertidores A/D y D/A
- Módulos de control de ancho de pulso

En DSP, más especializados podemos encontrar interfases serie con buffer de lectura/escritura de unos 2 Kbytes, con lo cual se puede obtener una disminución bastante notable de la carga media de la CPU [17], [18].

2.1.4. CARACTERÍSTICAS PRINCIPALES DEL dsPIC30F4013

CPU Modificada de Alto Funcionamiento RISC:

- Arquitectura Harvard modificada
- Arquitectura optimizada del sistema de instrucción del compilador de C
- Modos de direccionamiento flexibles
- 84 instrucciones base
- 24 bits de instrucciones amplias, trayectoria de datos extensos de 16 bits
- Espacio del programa Flash por arriba de 48 kbytes
- 2 kilobytes de datos en RAM
- 1 kilobyte de los datos no volátiles en EEPROM
- Arreglo del registro de trabajo de 16x16 bits
- Hasta 30 MIPS de operación:
 - Entrada externa del reloj de DC a 40 MHz
 - Entrada del oscilador de 4MHz-10MHz con PLL activo (4x, 8x, 16x)

- Hasta 33 fuentes de interrupción:
- 8 niveles seleccionables de prioridad para el usuario
- 3 fuentes externas de interrupción
- 4 trampas del procesador

Características del DSP:

- Trae datos dobles
 - Módulo y modos de bits invertidos
 - Dos acumuladores anchos de 40 bits con lógica de saturación opcional
 - Multiplicador de hardware decimal / entero de ciclo sencillo de 17x17 bits
 - Todas las instrucciones del DSP son de ciclo simple
- Multiplicar-Acumular la operación (MAC)
- Ciclo Sencillo alternado ± 16

Características Periféricas:

- Pines de corriente alta salidas / fuentes de 25 mA/25 mA
- Hasta cinco temporizadores / contadores de 16 bits y que opcionalmente se pueden conectar entre sí temporizadores de 16 bits para convertirse en módulos temporizadores de 32 bits
- Hasta cuatro funciones de entrada de captura de 16 bits
- Hasta cuatro funciones de salida de 16 bits del PWM
- La interfase del convertidor de datos (DCI) soporta protocolos de audio y Codecs, incluyendo I²S y AC' 97
- Módulo de 3 canales SPI (soporte de 4 modos de estructura)
- El módulo I²C soporta modos múltiples de Maestro/Esclavo y direccionamiento de 7 bits/10 bits
- Hasta dos módulos direccionables UART con buffers FIFO
- Módulo del bus CAN satisface la norma CAN 2.0B

Características Análogas:

- Convertidor análogo-digital de 12 bits con:
- Tasa de conversión de 100 ksps
- Hasta 13 canales de entrada
- Conversión disponible durante los modos "Sleep" e "Idle"
- Detección programable de bajo voltaje
- Generador programable de detección "Brown-out"

Características Especiales del Microcontrolador:

- Memoria de programación Flash mejorada
-10000 ciclos de escritura/borrado mínimos para el intervalo industrial de temperaturas, típicos 100K
 - Memoria de datos EEPROM:
-100000 ciclos escritura/borrado mínimo para el intervalo industrial de temperaturas típicas 1M
 - Auto reprogramable bajo control de software
 - Reset de encendido, temporizador de “encendido” y temporizador de inicio para el oscilador
 - Temporizador vigilante flexible con oscilador RC de bajo consumo integrado para operación rentable
 - Operación de monitoreo de ciclos a prueba de fallas
- Detecta fallas de reloj y cambia al oscilador RC de bajo consumo integrado

[19].

2.1.5. REGISTROS DE LA CPU

La CPU de dsPIC30F tiene un registro de 16 bits de estado (SR), el LSByte del cual se refiere como un registro más bajo del estado (SRL). El byte superior de SR se refiere como SRH. Una descripción detallada del SR se muestra en el registro 2-1.

SRL contiene todas las banderas de estado de la operación de MCU ALU, más los bits de estado de prioridad de la interrupción de la CPU, IPL<2:0 > y el bit de estado activo del REPET, RA (SR<4 >). Durante la excepción que procesa, SRL se concatena con el MSByte de la PC para formar un valor completo de la palabra, que entonces se apila.

SRH contiene los bits de estado de DSP Adder / Subtractor, el DO loop por el bit activo, DA (SR<9 >) y el bit del Digit Carry, D.C. (SR<8 >).

Los bits del SR son leídos / escritos con las siguientes excepciones:

1. El bit de DA (SR<8 >): DA es solamente un bit de lectura.
2. El bit RA (SR<4 >): RA es solamente un bit de lectura.
3. Los bits del OA, de OB (SR<15:14 >) y de OAB (SR<11 >): Estos bits son solamente de lectura y únicamente pueden ser modificados por el hardware del motor del DSP.
4. Los bits del SA, del SB (SR<13:12 >) y de SAB (SR<10 >): Éstos solamente son para definir y leer y pueden ser únicamente determinados por el hardware del motor del DSP. Una vez determinados, siguen estando fijos hasta ser definidos por el usuario, independientemente de los resultados de cualesquiera operaciones subsecuentes del DSP.

Nota: Claro que el bit SAB también limpiará tanto al bit SA como al SB.

CORCON: Registro De Control De la Base

El registro de CORCON contiene los bits que controlan la operación del multiplicador del DSP y el DO del hardware. El registro de CORCON también contiene el bit de estado IPL3, con el cual se concatena IPL<2:0 > (SR<7:5 >), para formar el nivel de la prioridad de la interrupción de la CPU.

Registro 2-1: SR: Registro De Estado de la CPU

Byte Superior:							
R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R-0	R/W-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15						bit 8	

Byte Inferior: (SRL)							
R/W-0 ⁽²⁾	R/W-0 ⁽²⁾	R/W-0 ⁽²⁾	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL<2:0>			RA	N	OV	Z	C
bit 7						bit 0	

bit 15 **OA:** Acumulador A de bit de Estado de salida

- 1 = Acumulador A de sobre flujo
- 0 = Acumulador A sin sobre flujo

bit 14 **OB:** Acumulador B de bit de Estado de sobre flujo

- 1 = Acumulador B de sobre flujo
- 0 = Acumulador B sin sobre flujo

bit 13 **SA:** Acumulador A de Saturación ‘Difícil’ del bit de Estado

- 1 = Acumulador A que está saturado o que ha sido saturado en alguna ocasión.
- 0 = Acumulador A que no está saturado.

Nota: Este bit puede ser de leído o definido (no determinado).

bit 12 **SB:** Acumulador B de Saturación ‘Difícil’ del bit de Estado

- 1 = Acumulador B que está saturado o que ha sido saturado en alguna ocasión.
- 0 = Acumulador B que no está saturado.

Nota: Este bit puede ser de leído o definido (no determinado).

bit 11 **OAB:** OA || OB Acumulador Combinado de bit de Estado de desborde

- 1 = Acumuladores A o B con sobre flujo
- 0 = Ninguno de los Acumuladores A o B con sobre flujo

bit 10 **SAB:** SA || SB Acumuladores Combinados con bit de Estado ‘Difícil’

1 = Acumuladores A o B están saturados o han sido saturados en alguna ocasión en el pasado.

0 = Ninguno de los Acumuladores A o B están saturados

Nota: Este bit puede ser leído o completado (no determinado). Completando este bit llenará a SA y SB.

bit 9 **DA:** DO Loop bit Activo

1 = DO loop en progreso

0 = DO loop no en progreso

bit 8 **DC:** bit Medio Carry / Borrow de MCU ALU

1 = Un carry fuera del 4º orden abajo del bit (para datos del tamaño de byte) u 8 abajo del orden de bit (para palabras del tamaño del dato) del resultado dado.

0 = Ningún carry fuera del 4o. orden abajo del bit (para el tamaño del dato del byte) u 8 abajo del orden del bit (para palabras del tamaño del dato) del resultado producido.

Registro 2-1: SR: Registro de Estado de la CPU (Continuación)

bit 7-5 **IPL<2:0>:** bits⁽¹⁾ de Estado del Nivel de Prioridad de Interrupción de la CPU.

111 = El Nivel de Prioridad de Interrupción de la CPU es 7 (15). Usar interrupciones inhabilitadas.

110 = El Nivel de Prioridad de Interrupción es 6 (14)

101 = El Nivel de Prioridad de Interrupción es 5 (13)

100 = El Nivel de Prioridad de Interrupción es 4 (12)

011 = El Nivel de Prioridad de Interrupción es 3 (11)

010 = El Nivel de Prioridad de Interrupción es 2 (10)

001 = El Nivel de Prioridad de Interrupción es 1 (9)

000 = El Nivel de Prioridad de Interrupción es 0 (8)

Nota 1: Los bits del IPL<2:0> están concatenados con el bit del IPL<3> (CORCON<3>) para formar el Nivel de Prioridad de Interrupción de la CPU.

El valor del paréntesis indica el IPL si IPL<3> = 1. Usar las interrupciones que están inhabilitadas cuando IPL<3> = 1.

2: Los bits de estado IPL<2:0> son leídos solamente cuando NSTDIS = 1 (INTCON1<15>).

bit 4 **RA:** REPEAT Loop bit Activo

1 = REPEAT loop en progreso

0 = REPEAT loop no en progreso

bit 3 **N:** Bit Negativo de MCU ALU

1 = El resultado fue negativo

0 = El resultado no fue negativo (cero o positivo)

bit 2 **OV:** Bit de sobre flujo de MCU ALU

Este bit es usado para la aritmética señalada (dos complementos). Indica un sobre flujo de la magnitud que causa la señal de bit para cambio de estado.

1 = Sobre flujo producido para la aritmética asignada (en esta operación aritmética)

0 = No produjo sobre flujo

bit1 **Z:** Bit Cero de MCU ALU

1 = Una operación que en efecto el bit ha determinado alguna vez en el pasado

0 = La más reciente operación que en efecto el bit Z ha definido (ejemplos un resultado no-cero)

bit 0 **C:** Bit Carry / Borrow de MCU ALU

1 = Un carry fuera del bit más significativo del resultado producido

0 = Ningún carry fuera del bit más significativo del resultado producido

Leyenda:		
R=Bit de lectura	W = Bit Escrito	U =Bit no implementado leído como 0
C = Definir sólo el bit	S = Solo bit determinado	-n = Valor en POR
'1' = El bit es determinado	0' = El bit está definido	x = El bit es desconocido

Registro 2-2: CORCON: Registro de Control de la Base

Byte Superior:							
U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0
—	—	—	US	EDT	DL<2:0>		
bit 15						bit 8	

Byte Inferior:							
R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF
bit 7							bit 0

bit 15-13 No implementado: Leído como '0'

bit 12 **US:** Bit DSP Multiplicar sin señal / Control de señal

1 = El motor DSP se multiplica sin señal

0 = El motor DSP se multiplica con señal

bit 11 **EDT:** Terminación temprana del bit de control DO Loop

1 = Terminar ejecutando DO loop al final de la iteración del loop actual

0 = Ningún efecto

Nota: Este bit se leerá siempre como '0'.

bit 10-8 **DL<2:0>**: Bits del Nivel de Estado DO Loop Nesting

111 = 7 DO loops activos

•

•

001 = 1 DO loop activo

000 = 0 DO loops activos

bit 7 **SATA**: Bit habilitado de saturación AccA

1 = Acumulador A habilitado de saturación

0 = Acumulador A inhabilitado de saturación

bit 6 **SATB**: Bit habilitado de Saturación AccB

1 = Acumulador B habilitado de saturación

0 = Acumulador B inhabilitado de saturación

bit 5 **SATDW**: Sitio de Información escrito desde el bit habilitado del Motor de Saturación DSP

1 = Escribe espacios de información de saturación habilitada

0 = Escribe espacios de datos de saturación inhabilitada

bit 4 **ACCSAT**: Saturación de Acumulador del bit de Modo Selecto

1 = 9.31 saturación (super saturación)

0 = 1.31 saturación (saturación normal)

bit 3 **IPL3**: Estado del Nivel de Prioridad de Interrupción del bit 3 de la CPU

1 = El nivel de prioridad de interrupción de la CPU es más grande que 7

0 = El nivel de la prioridad de interrupción de la CPU es 7 o menos

Nota: El bit IPL3 está concatenado con los bits IPL<2:0> (SR<7:5>) para formar el nivel de prioridad de interrupción de la CPU.

Registro 2-2: CORCON: Registro de Control de la Base(Continuación)

bit 2 **PSV**: Espacio de Programa visible en Espacios de Información de bit habilitados

1 = Espacio de Programas visibles en sitios de datos

0 = Espacios de Programas no visibles en sitios de datos

bit 1 **RND**: Rondando el bit de Modo Selecto

1 = Parcialidad (convencional) ronda habilitada

0 = Imparcialidad (convergente) ronda habilitada

bit 0 **IF**: Número Entero o Fraccionario Multiplicador del bit de Modo Selecto

1 = Modo de número entero habilitado para que DSP multiplique a ops

0 = Modo Fraccionario habilitado para que DSP multiplique ops

Leyenda:			
R = Bit de lectura	W = Bit escrito	U = Bit no implementado, leído como '0'	C = El bit no puede ser definido
-n = Valor en POR	'1' = El bit es determinado	'0' = El bit está definido	x = El bit es desconocido

Otros registros de Control de dsPIC30F

TBLPAG: Registro de la Página de la Tabla

El registro de TBLPAG se utiliza para llevar a cabo los 8 bits superiores de una dirección de memoria del programa durante la tabla de leer y escribir operaciones. Las instrucciones de la tabla son usadas para transferir datos entre el espacio de memoria del programa y memoria de los datos.

PSVPAG: Registro de la Página de Visibilidad del Espacio del Programa

La visibilidad del espacio del programa permite al usuario mapear una sección de 32 kbytes del espacio de memoria del programa en los 32 kilobytes superiores de espacio de dirección de los datos. Esta característica permite el acceso transparente de datos constantes a través de las instrucciones de dsPIC30F que operan en la memoria de datos. El PSVPAG, elige el registro de la región de 32 kilobytes del espacio de memoria del programa que es mapeado al espacio dirección datos.

MODCON: Registro del Módulo de Control

El registro de MODCON, se utiliza para habilitar y configurar la dirección del módulo (buffers circulares).

XMODSRT, XMODEND: Registros de dirección de inicio y fin del módulo X

Los registro de XMODSRT y XMODEND realizan el inicio y termino de las direcciones para el módulo (circular) de los buffers implementados en el espacio de dirección de memoria de los datos de X.

YMODSRT, YMODEND: Registros de dirección de inicio y fin del módulo Y

Los registros de YMODSRT y YMODEND realizan el inicio y termino de las direcciones para el módulo (circular) de los buffers implementados en el espacio de dirección de memoria de los datos de Y.

XBREV: Registro Bit-Inverso del Módulo X

El registro de XBREV se utiliza para fijar el tamaño del buffer utilizado para la dirección del bit-inverso.

DISICNT: El Registro de la Cuenta de las Interrupciones inhabilitadas

El registro de DISICNT es utilizado por la instrucción de DISI para inhabilitar interrupciones de la prioridad 1-6 para un número especificado de ciclos.

[19].

2.1.6. MAPAS DE LAS MEMORIAS INTERNAS

Mapa de direcciones de memoria del programa

Los dispositivos de dsPIC30F tienen un espacio de dirección de memoria del programa de 4M x 24-bit.

Hay tres métodos disponibles para tener acceso al espacio del programa:

1. Mediante la PC 23-bit.
2. Mediante la tabla (TBLRD) e instrucciones (TBLWT).
3. Por mostrar un segmento de 32 Kbytes de la memoria del programa en el espacio de dirección de memoria de datos.

El mapa de memoria del programa se divide en el espacio del programa de usuario y la configuración del espacio del usuario. El espacio del programa de usuario contiene el vector del reajuste, tablas del vector de la interrupción, memoria de programa y memoria de datos EEPROM. El espacio de la configuración del usuario contiene bits de configuración no-inestables para fijar opciones del dispositivo y las localizaciones de ID del dispositivo.

Básicamente la memoria del dsPIC30 está dividida en tres partes que son: la memoria de programa "P", la memoria "X" y la memoria "Y".

La memoria de programa es de 24 bits de tamaño, la cual contiene a su vez una memoria tipo FLASH y esta destinada a contener todas las instrucciones de un programa específico a ejecutar por el procesador.

Las memorias "X" e "Y" tienen un tamaño de 16 bits y están destinadas para escritura - lectura de datos temporales y también para el control de los periféricos internos mediante el uso de registros específicos. Más adelante se hará una descripción de estas memorias [19].

NOTA: Si desea conocer un poco más de estas características, por favor diríjase al apéndice 1.

CAPÍTULO III

EL EMISOR Y SENSOR ULTRASÓNICOS

3.1. PRINCIPIO FUNDAMENTAL DE FUNCIONAMIENTO

En la actualidad los emisores y sensores ultrasónicos son ampliamente usados para detectar la presencia de objetos mediante el principio de eco sonografía. Esto es, se envía un sonido de alta frecuencia y si hay presente un objeto frente a la fuente emisora, la señal rebotará en el objeto y regresará hacia la fuente emisora de nuevo pero con un retraso. Este retraso es medido por el procesador digital de señales y posteriormente será usado para generar una señal senoidal con periodo proporcional a la distancia que se midió.

3.2. DESCRIPCIÓN DEL SENSOR ULTRASÓNICO

El sensor ultrasónico mostrado en la figura 3.5 está integrado por dos circuitos idénticos los cuales tienen la misión de recibir la onda ultrasónica reflejada que originalmente es emitida por el emisor ultrasónico.

Refiriéndonos al análisis de uno de estos circuitos se puede ver que consta de un transductor piezoeléctrico X1 de 40 KHz, una etapa de amplificación, un detector de señal, un comparador de voltaje y una etapa en que se adecua la señal a niveles digitales estándares.

La etapa de amplificación está integrada por dos pasos de amplificación en donde el primer amplificador tiene una ganancia de voltaje de 100, mientras que el segundo amplificador tiene una ganancia de voltaje de 10, teniéndose entonces una ganancia final de 1000.

Como se puede ver en el diagrama de la figura 3.5, la señal proveniente del transductor X1 es amplificada 1000 veces obteniéndose así una señal que puede ser analizada y procesada adecuadamente por las siguientes etapas.

El detector de señal está integrado por el capacitor C3, los diodos D1, D2 y el capacitor C4, los cuales solo dejan pasar la parte positiva de la señal amplificada.

El capacitor C4 filtra la componente de alta frecuencia y solo deja pasar a la envolvente de dicha señal dejando así la posibilidad de ser procesada por la siguiente etapa.

El comparador de voltaje de la figura 3.5 compara los voltajes que hay en las entradas del amplificador operacional, teniéndose en la entrada no inversora a la

preferencia de voltaje y en la entrada inversora el voltaje a comparar, siendo este último el proveniente del detector de señal.

El funcionamiento es muy simple, cuando el voltaje de la señal de entrada es mayor que el voltaje de referencia a la salida del amplificador se tendrá un voltaje máximo. Cuando el voltaje de la señal de entrada es menor que el voltaje de referencia se tendrá a la salida un voltaje cero.

La etapa que adecua la señal de salida del amplificador a niveles digitales está integrada por los siguientes elementos: un divisor de tensión integrado por las resistencias R6 y R7, un Flip-flop integrado por las compuertas NAND U2.

La misión del Flip-flop es solo dejar pasar el primer impulso proveniente de la señal del comparador, siendo este Flip-flop inmune a todos los impulsos que ocurran posteriormente.

Este Flip-flop puede ser restablecido por la señal de reset proveniente de la Terminal J2 del diagrama.

El circuito transmisor está integrada por el oscilador de 40 KHz, el amplificador de corriente y el emisor ultrasónico LS1. El circuito oscilador puede ser encendido y apagado por el dsPIC30 a través de la Terminal J3.

Para poder medir el tiempo transcurrido entre la onda emitida por el transmisor y la onda reflejada recibida por el receptor X1 se tiene el siguiente principio fundamental de operación:

1. El dsPIC30 primero manda una señal de reset al Flip-flop a través de la Terminal J2.
2. En seguida es encendido el oscilador de 40 KHz.
3. Cuando la onda reflejada por un objeto se genera un impulso que es capturado por el Flip-flop, enviándose un pulso al dsPIC30 a través de la Terminal J1.
4. El procesador digital de señales calcula el tiempo transcurrido entre el encendido del oscilador y el pulso recibido.

Cabe destacar que el tiempo de medición se determinará partiendo de:

$$\lambda = \frac{34300 \left[\frac{cm}{seg} \right]}{1 \left[\frac{TM}{2} \right]}$$

Siendo la constante 34300 cm / s la velocidad del sonido

TM= tiempo medido

λ = es la distancia medida

3.3. EL EMISOR Y RECEPTOR ULTRASÓNICOS

3.3.1. EL EMISOR

El emisor esta integrado por dos elementos:

- a) El circuito oscilador
- b) El transductor ultrasónico

El circuito oscilador esta integrado por un oscilador astable que genera una señal cuadrada de 40kHz y tiene a al salida un amplificador de corriente como etapa de potencia para poder manejar posteriormente al transductor.

La etapa de potencia esta constituida por compuertas inversores en paralelo como se muestra en la figura.

Este circuito es encendido y apagado en intervalos regulares de tiempo para así establecer un punto de referencia entre la señal generada y la señal que rebota en el objeto.

El transductor es un elemento piezoeléctrico sintonizado para oscilar específicamente a una frecuencia de 40kHz y consta de un material elástico, un elemento piezoeléctrico y un resonador como se muestra en la figura 3.1.

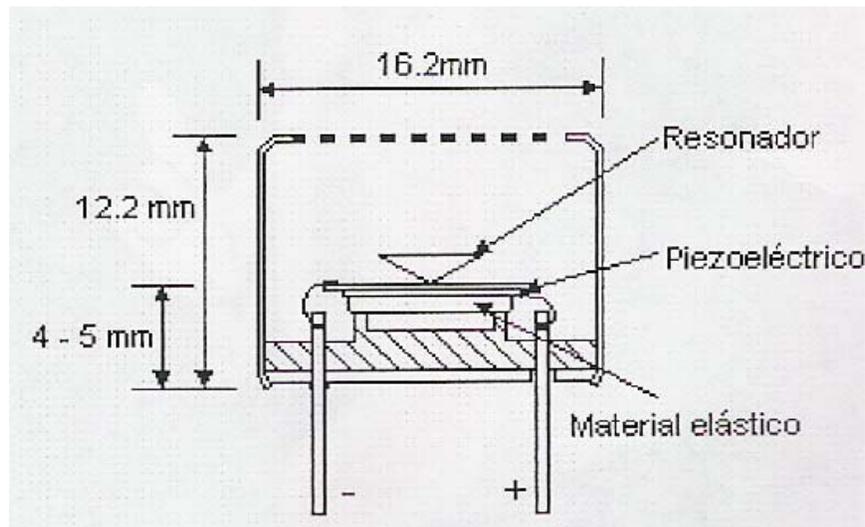


Figura 3.1 Transductor ultrasónico. [20]

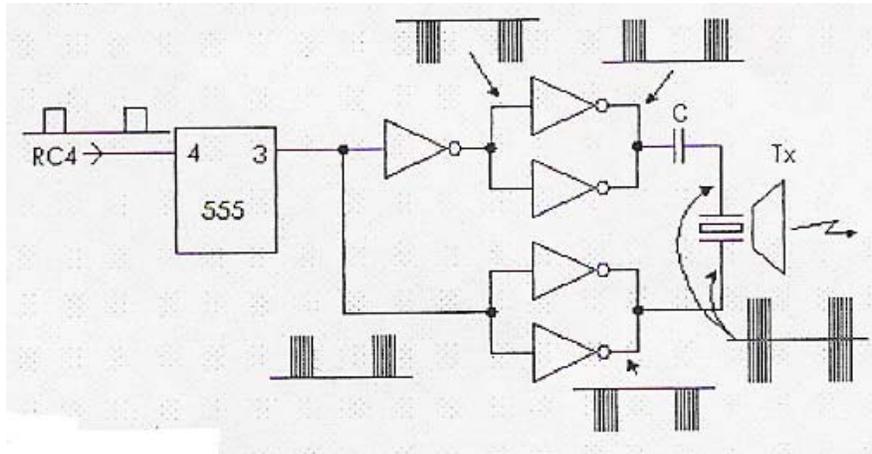


Figura 3.2 Circuito del Transmisor. [20]

3.3.2. EL RECEPTOR

El receptor consta de tres secciones las cuales son:

- a) El sensor ultrasónico
- b) El amplificador
- c) El circuito de detección

El sensor ultrasónico es idéntico al transductor ultrasónico y la diferencia entre estos radica en la forma como se usan.

Si se le aplica una señal al transductor este generará una señal de 40kHz y si se usa el mismo transductor como micrófono este será capaz de traducir una señal ultrasónica a señal eléctrica. Cabe hacer la aclaración que solo responderá a una señal que tenga precisamente una frecuencia de 40kHz.

La siguiente etapa es un amplificador que recibe la señal proveniente del sensor ultrasónico y esta integrado por dos etapas amplificadoras que dan una ganancia total de 1000 (60 dB). La primera etapa amplifica la señal 100 veces (40 dB) y la segunda amplifica 10 veces (20 dB). El circuito es mostrado en la figura 3.3.

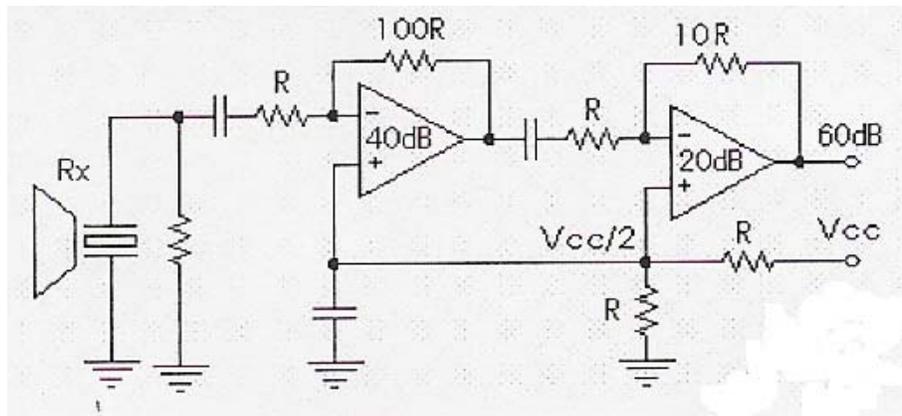


Figura 3.3 Amplificador de señal. [20]

El circuito de detección está integrado por un rectificador de media onda y un comparador. Cabe aclarar que a la salida del rectificador de media onda hay un capacitor que filtra a la señal de 40kHz, dando como resultado la aparición de un pulso cada vez que esta señal es detectada, teniéndose así justamente la señal que será comparada contra la señal generada inicialmente.

El intervalo de tiempo entre el inicio de la señal generada y la aparición del pulso entregado por el detector es proporcional a la distancia del objeto sobre el cual rebotó la señal. El circuito de detección se muestra en la figura 3.4.

La distancia real a la que se encuentra un determinado objeto es proporcional al intervalo de tiempo dividido entre 2, y la distancia por lo tanto se podrá calcular en base a la velocidad del sonido, considerando a dicha velocidad igual a 34300 cm / s (a 20 °C). La fórmula para el cálculo de la distancia es la siguiente:

$$\lambda = \frac{34300 \left[\frac{cm}{seg} \right]}{\left[\frac{TM}{2} \right]}$$

Por ejemplo, si el tiempo calculado fue de 50ms, de acuerdo a la fórmula anterior la distancia λ será:

$$\lambda = \frac{34300 \left[\frac{cm}{seg} \right]}{\frac{50^{-3}}{2}} = 857.5 \text{ cm}$$

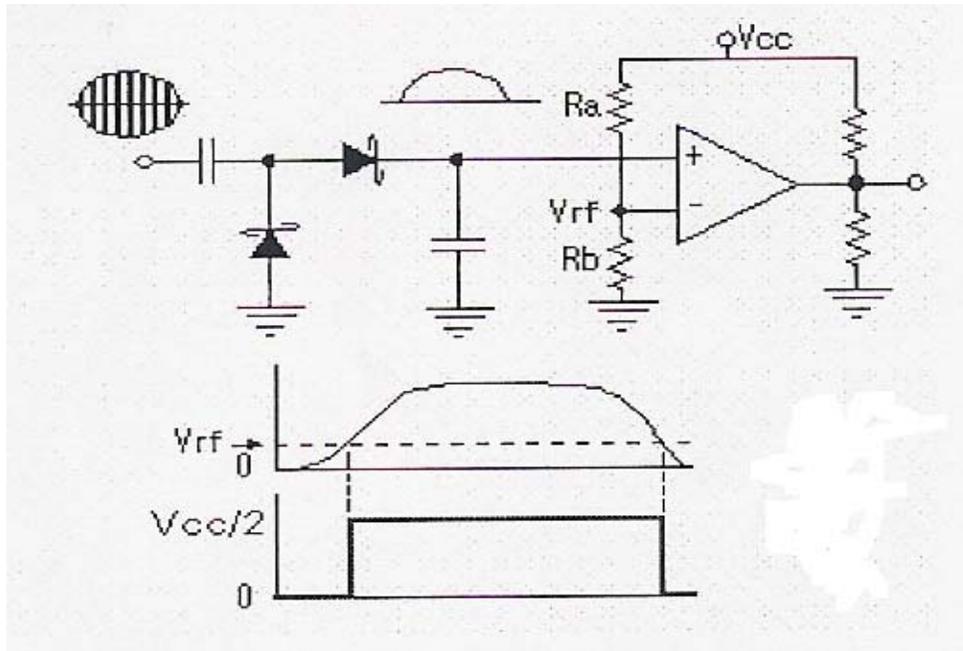


Figura 3.4 Circuito de detección. [20]

3.4 LA INTERFASE DIGITAL ANALÓGICA

El Convertidor Digital Analógico que se usó fue el DAC712P el cual tiene una entrada paralela de 16 bits.

Es un Convertidor bipolar que tiene capacidad de manejar notación en complemento a dos, es decir, que puede manejar valores positivos de voltaje a la salida cuando el bit más significativo es cero, mientras que cuando ese bit es uno los voltajes presentes a la salida serán negativos.

En la figura 3.6 puede verse las conexiones eléctricas de este dispositivo, así como su conexión con el dsPIC30 y su adecuación para poderse conectar a un amplificador de audio.

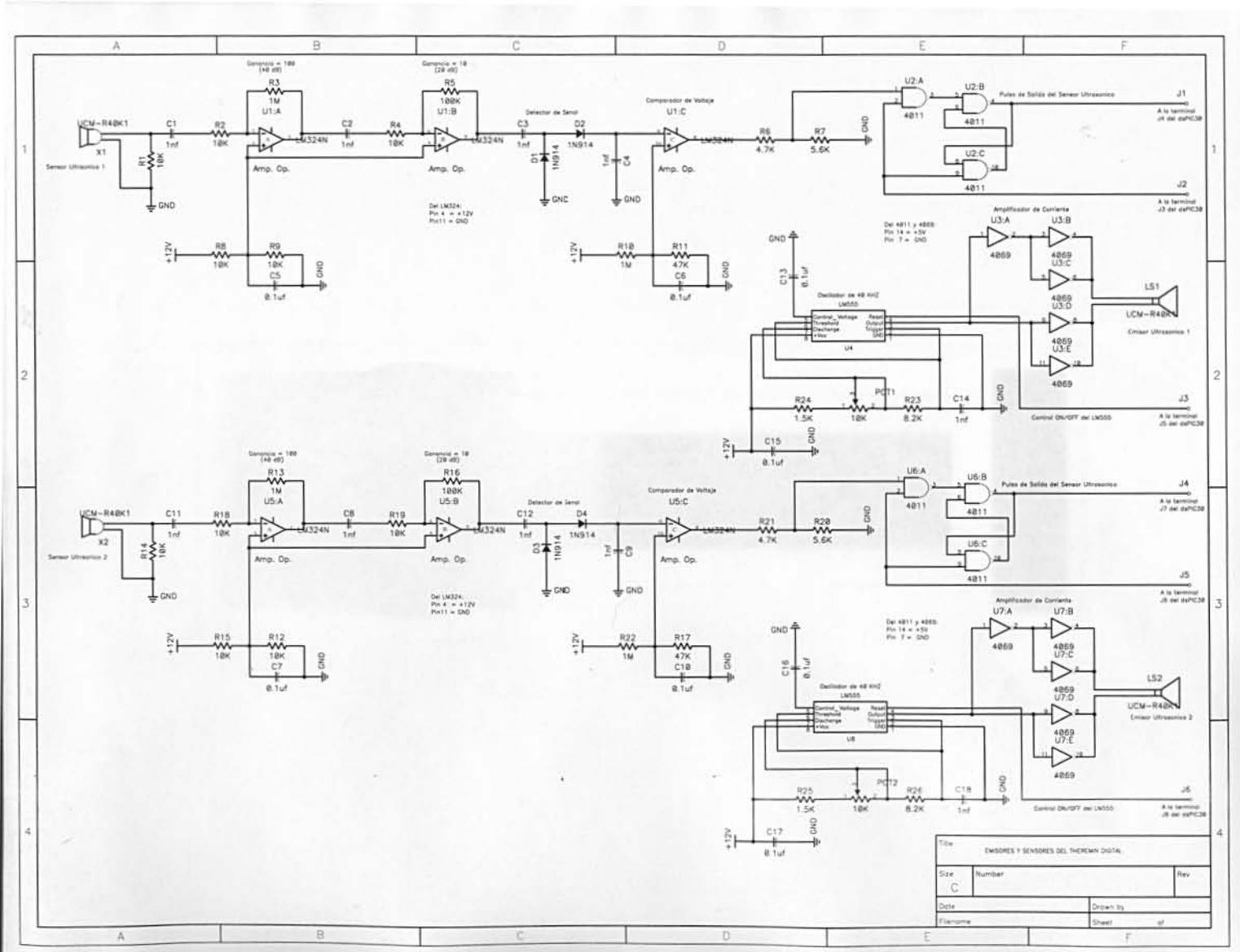


Figura 3.5 Circuito de los Emisores y Sensores del Theremin Digital

Título		
EMISORES Y SENSORES DEL THEREMIN DIGITAL		
Size	Number	Rev
C		
Date	Drawn by	
File name	Sheet	of

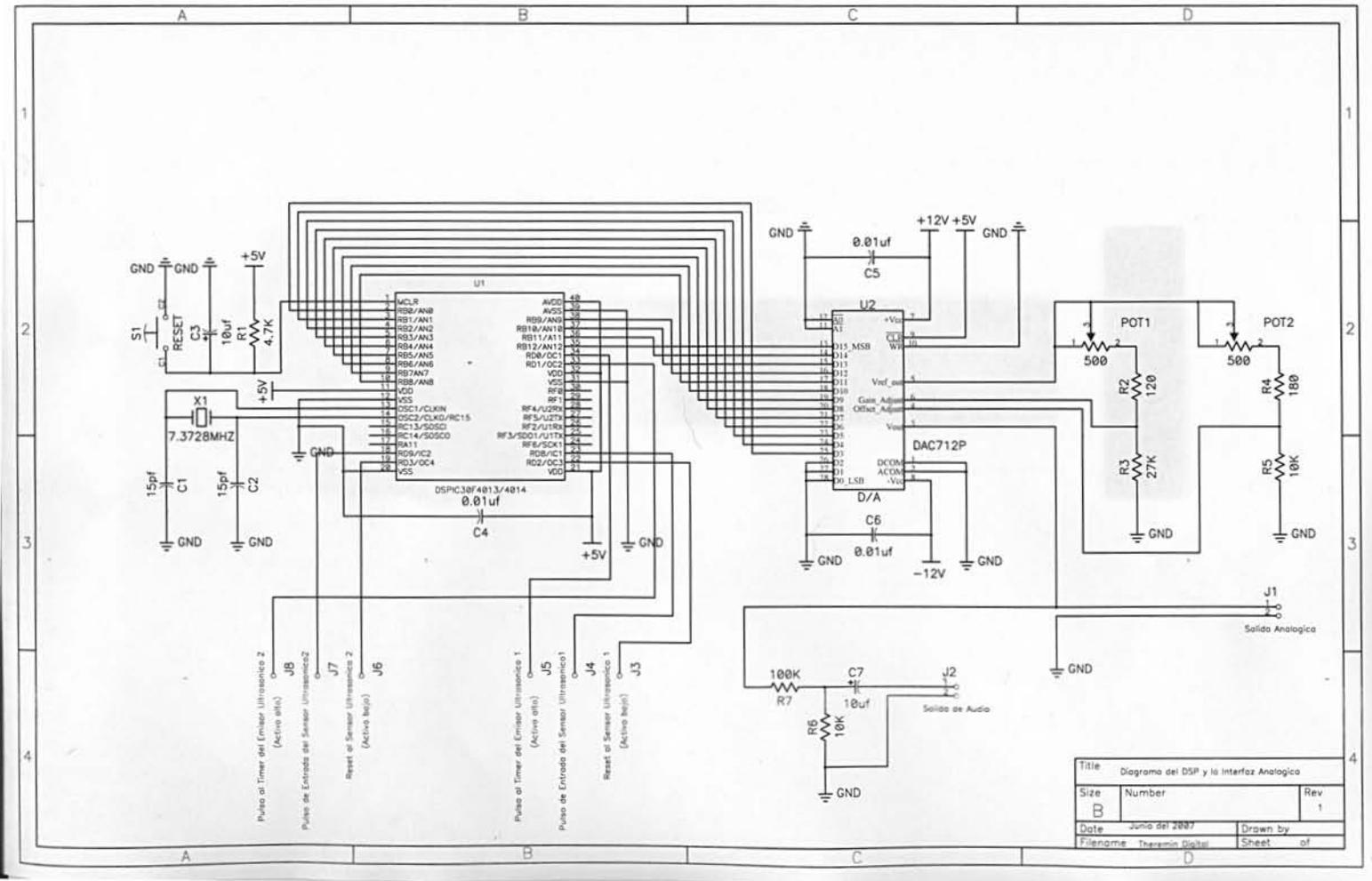


Figura 3.6 Diagrama del DSP y la Interfase Analógica

CAPÍTULO IV DISEÑO DEL SOFTWARE

4.1. EL COMPILADOR DE LENGUAJE C

El procesador digital de señales dsPIC30 en general, tiene un compilador de lenguaje C bastante poderoso que es capaz de trabajar en combinación con el ambiente MPLAB de Microchip.

Dentro de sus cualidades esta la de poder trabajar con aritmética en punto fijo de acuerdo a la arquitectura interna del procesador que el fabricante está proporcionando.

El corazón del dsPIC30 tiene la habilidad de trabajar en aritmética de punto fijo y tiene instrucciones para ese fin, sin embargo, al trabajar sobre el compilador solamente es posible trabajar en aritmética de enteros a menos que se utilice la Librería de DSP.

Para poder utilizar plenamente la aritmética en punto fijo es necesario utilizar instrucciones de lenguaje ensamblador en forma mixta o sea, mezclando instrucciones de lenguaje C e instrucciones en línea de lenguaje ensamblador.

Dentro de las instrucciones en lenguaje ensamblador se tiene el poder de mezclar operandos dados en lenguaje C dentro de los operandos en lenguaje ensamblador, sin embargo, para poder lograr este fin es necesario darle un tratamiento especial a dichos operandos

Algunas de las notaciones tales como el formato Q15, requieren de la Librería de DSPs para poder compilarla. Los resultados entregados en punto fijo por el corazón del DSP son automáticamente ajustados a esta notación de formato Q15 según se vaya necesitando.

Adicionalmente, el ambiente MPLAB tiene en su simulador comandos que nos permiten visualizar numéricamente el formato Q15 sin tener que estar haciendo conversiones de datos adicionales.

El formato Q15 es un formato que nos permite expresar cantidades numéricas fraccionales. Con este formato es posible representar cantidades numéricas dentro del rango ± 1 , siendo la palabra a representar un número de 16 bits dentro del cual el bit más significativo corresponderá al signo, y los 15 bits menos significativos representarán a la cantidad fraccional dada en formato binario.

La representación numérica de un número en punto fijo se realiza de la siguiente manera:

$$NUMFIX = (MSB) * 2^{-1} + (MSB - 1) * 2^{-2} + \dots + (LSB) * 2^{-15}$$

Siendo NUMFIX el número a representar en punto fijo, MSB el bit más significativo y el factor 2^{-1} de más peso. A continuación se sigue a partir del siguiente bit y el siguiente factor, formándose de esta manera una sumatoria que representa al número en punto fijo, siendo LSB el bit menos significativo y 2^{-15} el factor menos significativo.

Cabe hacer notar que de los 16 bits de la fracción total el bit más significativo de todos corresponderá al signo y los siguientes 15 bits representarán a la fracción decimal cuya interpretación se realiza mediante la ecuación antes expuesta y al mismo tiempo recibe el nombre de formato Q15 debido a que se están utilizando solo 15 bits.

Un 1 lógico asignado al bit más significativo corresponderá a una fracción negativa y un 0 lógico corresponderá a una fracción positiva.

Sin embargo, al integrar en forma mixta instrucciones en ensamblador y en C, nos da la posibilidad de trabajar con fracciones decimales en punto fijo de una manera muy flexible.

Una de las ventajas de trabajar en punto fijo es que el procesador puede trabajar en sus máximas posibilidades, pero una desventaja podría ser la interpretación de datos sino se dispone de las herramientas necesarias.

Cabe hacer notar que el procesador también tiene herramientas en lenguaje C desde las cuales se puede hacer trabajar al procesador en punto flotante. Sin embargo, como el DSP no es de punto flotante, se va a tardar más tiempo en realizar los cálculos matemáticos.

En las rutinas de software se está utilizando una multiplicación en punto fijo, siendo ésta la que controla el volumen o amplitud de la onda senoidal que se está generando por el Theremin.

Cabe hacer notar que este compilador resulta ser demasiado amigable y fácil de manejar, ya que se parece a otros compiladores usados por otros DSPs más grandes.

4.2. RUTINA PRINCIPAL DEL SISTEMA

El Theremin Digital para su funcionamiento está conformado por una serie de rutinas en lenguaje C que le dan las características operativas al hardware que está predestinado para tal fin.

En la Rutina Principal se está considerando las condiciones iniciales que se presentan inmediatamente después del reset, así como la inicialización de los periféricos que se están utilizando para hacer posible el funcionamiento del Theremin.

Las rutinas de inicialización de periféricos involucrados en la propia operación del Theremin son las siguientes:

- a) Reset.
- b) Rutina de Inicialización del Módulo de Captura 1.
- c) Rutina de Inicialización del Módulo de Captura 2.
- d) Rutina de Inicialización del Timer 2.
- e) Rutina de Inicialización del Timer 4.

La inicialización después del reset involucra a la configuración del Puerto B para que sea este un puerto de salida digital de datos. Cabe hacer notar que este puerto es de 13 bits, mientras que el convertidor Digital/Analógico es de 16 bits. Para hacer compatible la salida del Puerto B con el convertidor se conectó este de la siguiente manera:

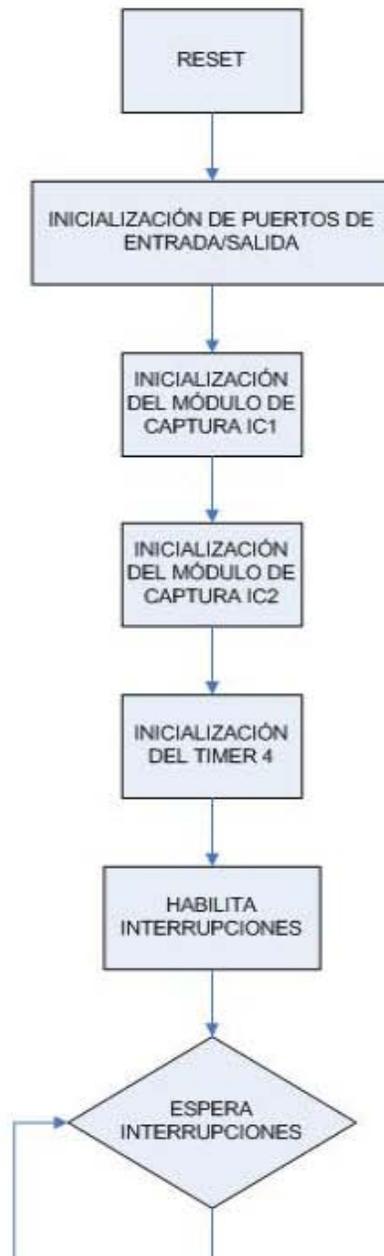
Pin del Puerto B		Pin del convertidor D/A
PB12	=	D15
PB11	=	D14
PB10	=	D13
PB9	=	D12
PB8	=	D11
PB7	=	D10
PB6	=	D9
PB5	=	D8
PB4	=	D7
PB3	=	D6
PB2	=	D5
PB1	=	D4
PB0	=	D3
Tierra Física	=	D2
Tierra Física	=	D1
Tierra Física	=	D0

Tabla 4.1. Conexiones del Puerto B con el convertidor D/A

Como los datos que maneja el dsPIC30F4013 son de 16 bits, para hacer compatibles estos datos, al mandar una escritura al Puerto B se realiza previamente un corrimiento de tres bits a la derecha para ajustar la palabra al tamaño del puerto quedando truncados los

tres bits menos significantes. No obstante que hay una pérdida de tres bits el funcionamiento del sistema es bastante aceptable.

RUTINA PRINCIPAL



Respectivamente cada rutina de inicialización llama específicamente a una subrutina que se encarga de configurar a cada uno de los periféricos del DSP para poder funcionar acordeamente con las señales de entrada/salida del dispositivo.

4.2.1. SUBRUTINAS LLAMADAS POR LA RUTINA PRINCIPAL

A continuación se presenta a los diagramas de flujo de cada una de estas subrutinas:

SUBROUTINA DE INICIALIZACIÓN DE IC1



SUBROUTINA DE INICIALIZACIÓN DE IC2



SUBROUTINA DE INICIALIZACIÓN DEL TIMER 4



4.3. RUTINA DE INTERRUPCIÓN DEL MÓDULO DE CAPTURA DE ENTRADA IC1.

La misión del Módulo de Captura IC el cual tiene conectado el hardware del receptor ultrasónico es detectar el rebote de la señal ultrasónica que es emitida por el transmisor ultrasónico.

Hay que recordar que el Sistema del Theremin contiene dos transmisores ultrasónicos y dos receptores capaces de detectar las señales emitidas por los transmisores.

Dentro de las acciones que realizan cada par de emisores y receptores ultrasónicos están las siguientes: a) Modulación en frecuencia de la onda senoidal de salida del Theremin, b) Modulación en amplitud de esa onda senoidal.

El Módulo de Captura IC1 tiene la misión de modular en frecuencia a la onda senoidal proporcionalmente al intervalo de tiempo medido de dicho módulo.

Los transmisores ultrasónicos pueden ser encendidos o apagados por el dsPIC30 y es necesario establecer un marco de referencia para poder medir un intervalo de tiempo entre la señal emitida y la señal reflejada. Como ya se sabe, el Módulo de Captura IC1 e IC2 tienen como base de tiempo al Timer 2 del procesador y todas las funciones de medición están hechas en base a ese Timer.

Cuando el emisor ultrasónico se enciende, al mismo tiempo se toma la lectura inicial del Timer 2 a la que se le llama “Flanco 1” e inmediatamente se activa la rutina de interrupción para poder medir el tiempo en que la señal emitida es reflejada. Cuando ésta señal es detectada inmediatamente entra en operación la rutina de interrupción para poder leer el “Flanco 2” que corresponde a una segunda lectura al Timer 2.

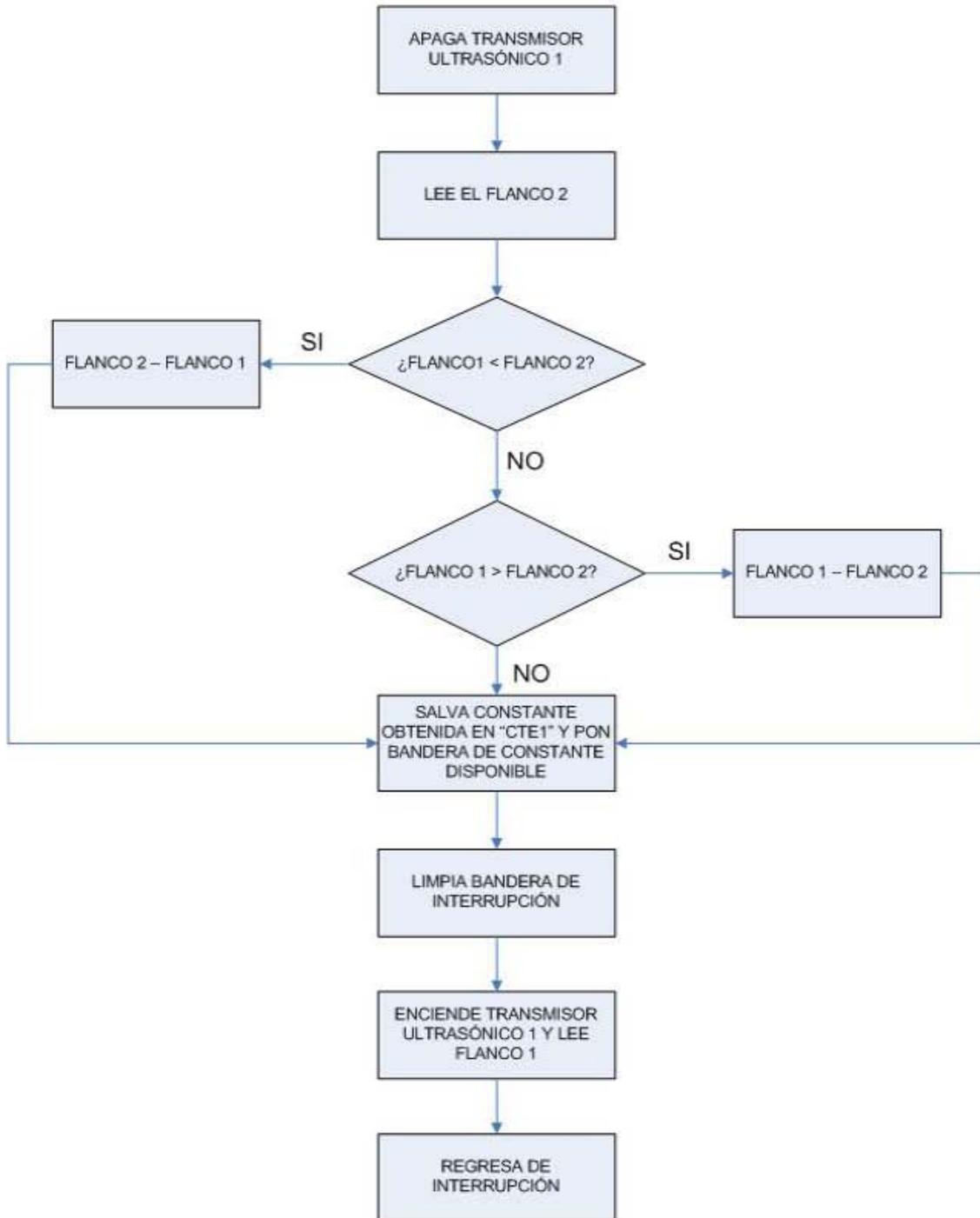
Como el Timer 2 es de 16 bits y siempre está funcionando, puede darse el caso de que la lectura del Flanco 1 se haya tomado cuando el Timer 2 se encontraba cerca del tope de su máxima lectura mientras que la lectura del Flanco 2 se estuviera dando al inicio de la cuenta de dicho Timer. Como consecuencia de esto se tendrá que la lectura del Flanco 1 es mayor que la lectura del Flanco 2, para lo cual la lógica nos diría que el Flanco 2 debe ser mayor que el Flanco 1.

Cabe aclarar que el Timer 2 cuenta de 0x0000 hasta 0xFFFF y luego cuenta de nuevo a partir de cero, repitiéndose así de esta manera el ciclo de cuenta.

Para solucionar el problema antes expuesto en la Rutina de Interrupción se dio una solución viable en la que primero se determina cual de los flancos es mayor y finalmente se realiza una resta para obtener el intervalo de tiempo medido entre los flancos. Después esta constante obtenida es salvada para ser utilizada por la Rutina de Interrupción del Timer 4 que se encarga de generar la onda senoidal y modularla en frecuencia en base a la constante obtenida previamente.

A continuación se presenta el diagrama de flujo de la rutina de interrupción que regula a todas las acciones antes descritas.

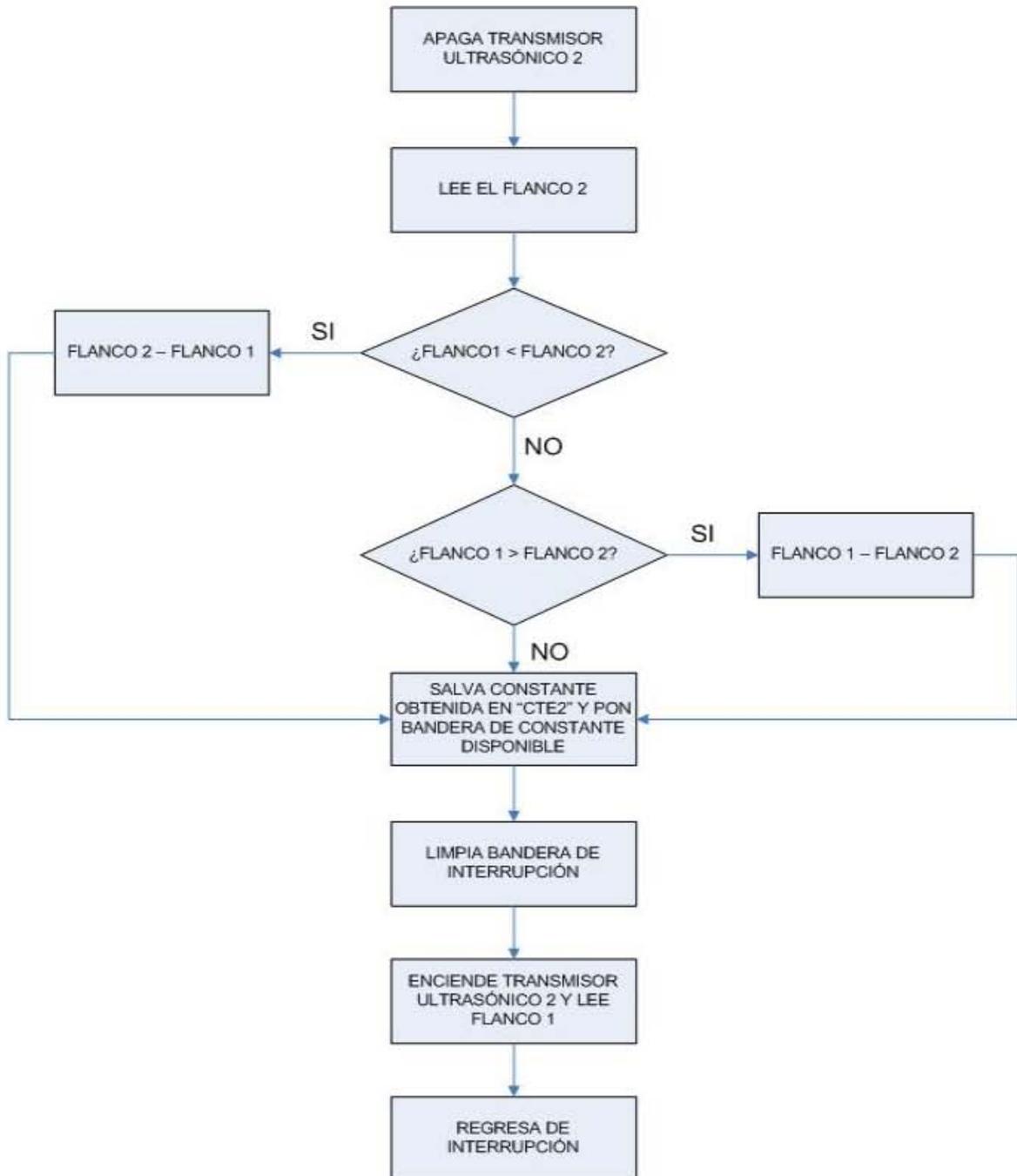
RUTINA DE INTERRUPCIÓN DEL MÓDULO DE CAPTURA IC1



4.4. RUTINA DE INTERRUPCIÓN DEL MÓDULO DE CAPTURA DE ENTRADA IC2.

Esta rutina funciona de manera similar a la anterior, pero la diferencia radica en que la constante obtenida es usada por la Rutina de Interrupción del Timer 4 para modular en amplitud a la onda senoidal que se está generando.

RUTINA DE INTERRUPCIÓN DEL MÓDULO DE CAPTURA IC2



4.5. RUTINA DE INTERRUPCIÓN DEL TIMER 4

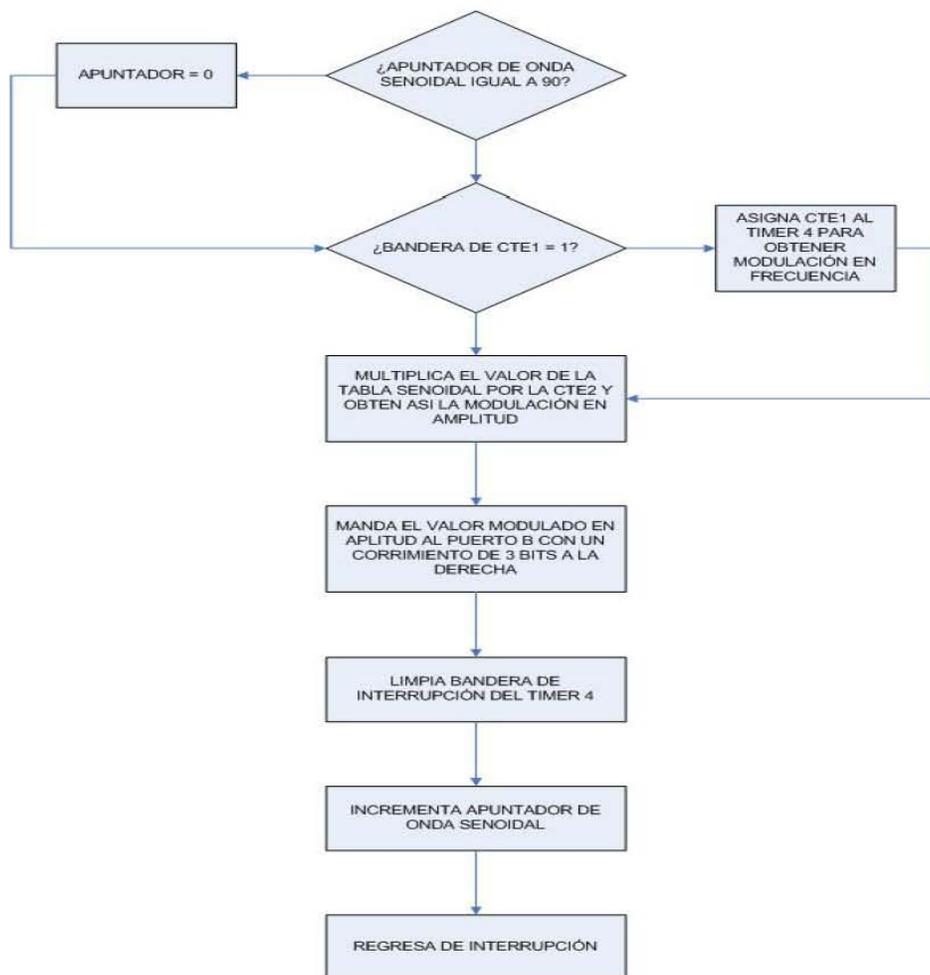
Esta rutina de interrupción tiene la misión de generar una onda senoidal con una resolución de 90 puntos y también la de actuar como modulador en frecuencia y amplitud de dicha señal.

La señal senoidal de salida es aplicada a través del Puerto B a un convertidor digital-analógico para tener así una onda senoidal con características de periodo y amplitud.

Cada vez que las rutinas de interrupción de IC1 e IC2 detectan la presencia de un objeto, la rutina de interrupción del Timer 4 es actualizada en frecuencia y amplitud.

La multiplicación hecha en la rutina de interrupción para lograr la modulación en amplitud aprovecha las características del procesador dsPIC30 de aritmética en punto fijo en donde el número fraccional siempre estará dentro del rango de ± 1.0

RUTINA DE INTERRUPCIÓN DEL TIMER 4



NOTA: Si desea conocer el listado de programa, por favor diríjase al apéndice 2.

CAPÍTULO V RESULTADOS

5.1. PRUEBAS Y ANÁLISIS

El Theremin, registra una senoidal discreta con resolución de 90 puntos como se muestra en la siguiente figura, la cual es una prueba realizada a partir de la conexión de la salida de audio de nuestro Theremin y uno de los canales del osciloscopio.

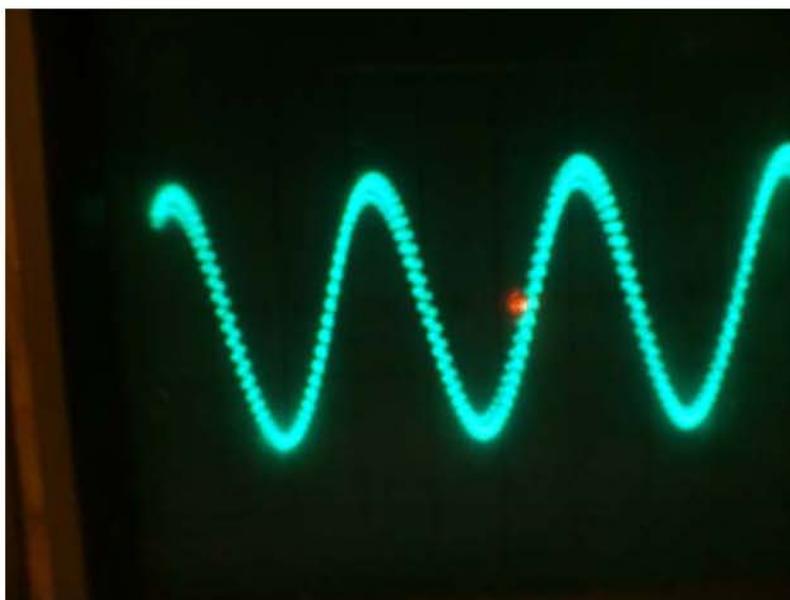


Figura 5.1. Senoidal discreta del Theremin

Debido a lo anterior es preciso mencionar que para la existencia de armónicos en el Theremin, es necesaria la interacción de dos Theremins. Una convolución genera armónicos, pero en la realidad no es el hecho de que exista una multiplicación de dos sonidos, si no que para el ambiente es una suma de dos sonidos. Por ejemplo:

$$F(t) = [\text{sen } 2\pi f_{la} t] [\text{sen } 2\pi f_{si} t]$$

Donde: f_{la} , es la frecuencia de la nota la y f_{si} , la frecuencia de la nota si.

La ecuación anterior es un ejemplo de una convolución entre las notas musicales, la y si, en la realidad es simplemente es una suma y un claro ejemplo lo podemos notar en un concierto, la interacción que existe entre los diferentes instrumentos musicales y sus sonidos, es simple y sencillamente una adición de sonidos.

Asimismo, es preciso mencionar que las notas musicales trabajan a diferentes frecuencias, así por ejemplo:

$f_{la} = 440 \text{ Hz}$

$$la^\# = la \left(\sqrt[12]{2} \right) = 466.16 \text{ Hz}$$

$$si = la^\# \left(\sqrt[12]{2} \right) = 493.88 \text{ Hz}$$

Con todo lo anterior, es preciso mencionar que los sonidos agradables al oído generalmente tienen una estructura complicada y los tonos puros, representados por ondas senoidales, son muy desagradables al oído [21].

5.2. FOTOGRAFÍAS DE NUESTRO PROYECTO DEL THEREMIN

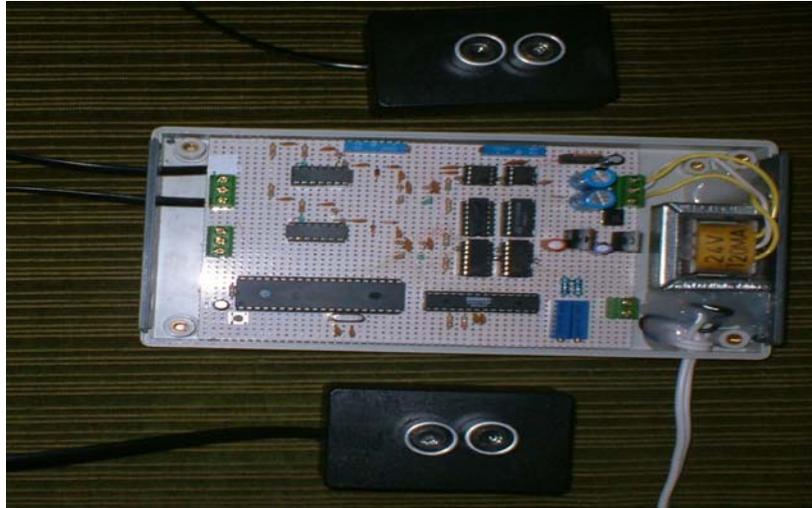


Figura 5.2. Circuito Interno del Theremin con base al dsPIC30F4013



Figura 5.3. Realizando pruebas de sonido con el Theremin

CONCLUSIONES

Hoy por hoy y sin duda alguna, cada día me doy cuenta de que si bien aún no tenemos muy en claro el origen evolutivo de la música y que en la actualidad se han generado cientos de debates entre los investigadores para explicarlo, sí me queda claro y sobre todo con el desarrollo de este proyecto que la tecnología y la ciencia prestan servicio al arte y en gran medida, sobre todo con los avances surgidos en el siglo pasado y con los que siguen surgiendo en el actual y seguirán surgiendo.

Cabe mencionar que la música es innata, puesto que nacemos dotados para apreciarla sin que se nos enseñe. Además es preciso destacar que la música es fundamental en nuestra vida y que el gusto que desarrollamos por ella es debido a que nos inspira emociones que van desde lo abstracto hasta la nostalgia.

Así pues, el sonido además de sus características medibles también entiende de los detalles de nuestra sensación auditiva, con lo que la percepción hace en cierta forma colaboración entre el órgano que capta tal estímulo y el cerebro, que lo interpreta. Luego entonces, es bueno mencionar que el oído es un analizador de ondas sonoras y que cuando se produce algún sonido, este realiza ciertas tareas con lo que el sonido está lleno de información útil acerca del entorno y para interpretarla el cerebro no actúa como una grabadora, que recibe y registra una señal nada más, sino que también reparte el estímulo a las diversas regiones del encéfalo, donde se llevan los procesos de reconocimiento e interpretación.

Con el desarrollo de mi tesis viví de cerca la experiencia de que para producir sonido hay que poner a vibrar algún objeto y que las cualidades del sonido dependen en gran medida de las propiedades de las ondas que produce el objeto al vibrar en el aire. El volumen o intensidad del sonido actúan como función de la amplitud. El tono depende del número de veces que se repite la vibración en un segundo, lo que nos da la sensación de la nota y a lo que se le conoce como frecuencia. También cabe destacar como característica al timbre, que es lo que nos permite distinguir el sonido de un instrumento y de otro. Asimismo el timbre depende de ciertas características, entre las que se encuentran: una frecuencia baja que sabemos se conoce como fundamental y que es la que nos da la nota que escuchamos, y posteriormente todos los múltiplos de esa frecuencia.

Esas frecuencias superiores se conocen como armónicos. Por lo general, los armónicos son más débiles y la frecuencia fundamental es más intensa. Con esto último es bueno hacer mención y referencia al gran impacto que han tenido tanto la Física como las Matemáticas al mundo de la Música, puesto que Fourier, fue el que descubrió una forma de armar y desarmar funciones periódicas complicadas, entre ellas las ondas sonoras. Este método nos permite considerar las ondas complicadas como superposiciones, o sumas, de ondas senoidales simples, llamadas como ya se explicó, armónicos y con ello toda onda sonora se puede desarmar y sintetizar en términos de tonos puros.

También he podido observar mediante las pruebas realizadas que la forma más sencilla de sonido es la que se puede representar por una onda senoidal y a los sonidos que representan se llaman tonos puros.

Ahora bien, dentro de este proyecto debo destacar la importancia que tienen las diferentes técnicas de procesamiento utilizadas para crear música electrónica como las que se usan para crear nuevos sonidos y las que se utilizan para transformar o modificar sonidos que ya están registrados.

La aplicación de la Ingeniería y de la Tecnología en el desarrollo de este proyecto y en función de las aplicaciones que brinda a este tipo de técnicas es fundamental, puesto que ya sabemos que nosotros como ingenieros debemos de innovar y de hacer uso de los recursos para dar aplicación y solución a los problemas optimizando el funcionamiento de los dispositivos o de la infraestructura que se este utilizando.

En nuestro caso llegamos a una simplificación óptima de un instrumento musical muy poco usado, pero que sin duda alguna llegó a revolucionar a los instrumentos musicales electrónicos y que hoy lo presentamos de forma reducida, pero no por ello sin trascendencia, por el contrario es este un claro ejemplo de la Ingeniería en Computación, Ingeniería Eléctrica y de tecnología modernas.

De lo anterior, tenemos que en épocas anteriores eran muy usados los filtros de control por voltaje y los osciladores, pero hoy en día estas funciones también se realizan en conjunto con la computadora o simplemente con esta sola.

Así pues, que la música por computadora se realiza empleando tecnología digital, como números que por medio de procesos matemáticos pueden manipularse mediante el uso de software, de aquí la importancia en gran medida de la Ingeniería de Software.

Así hoy en día la música electrónica acapara varios ámbitos que van desde la creación de obras que se conforman con la combinación de sonidos electrónicos y acústicos, además de aplicarla al video, el cine o la danza, como lo vemos en la actualidad. Puede ser que en el futuro, una vez que la informática y la tecnología electrónica queden totalmente integradas en el mundo del compositor el término de música electrónica desaparezca.

Por otro lado en la elaboración del proyecto han sido de suma importancia tanto el sensor ultrasónico, como el procesador digital de señales, puesto que son estos los encargados de cierta forma de la mayor parte del funcionamiento de nuestro Theremin y con ello se hace una combinación de la parte analógica y digital.

En tanto, la tarea desempeñada por el ultrasónico ha sido la de comportarse como un medidor a distancia, tanto por el hecho de transmitir como de reflejar ultrasonidos, la función del dsPIC, ha sido la de procesar y analizar nuestra información de manera digital, ayudado en consecuencia por las instrucciones del MPLAB y el compilador del lenguaje C.

En resumen, el sonido es una alteración repetitiva de la presión del aire y con ello la música no deja rastros duraderos una vez que se termina.

En consecuencia, no me queda duda alguna y dado este proyecto de que la música es un movimiento organizado de sonidos a través de un espacio de tiempo. La música desempeña un papel importante en todas las sociedades y existe en una gran cantidad de estilos y la música electrónica en relación a la música ha sabido adaptarse a los avances tecnológicos, los cuales han sido de gran influencia para su desarrollo.

En este proyecto como se ha podido observar el Theremin es un instrumento difícil de tocar al igual que el violín, ya que no tiene trastes como una guitarra y que para producir una nota se requiere de mucha práctica.

Por último, me queda la firme certeza de que la electricidad juega un papel sumamente importante, ya que ella interactúa con el sonido y de esta manera podremos darle a las notas musicales diferentes matices y obviamente por medio de la electricidad podrán reproducirse.

Además tengo que confesar que estoy muy conforme y feliz con los resultados obtenidos independientemente de que el instrumento no es meramente melódico, pero creo que esto es al principio, me parece que con la práctica la forma de ejecutarlo se vuelve mejor y más agraciado será tal vez su sonido y con ello se verá francamente las posibilidades que tenga el instrumento y la aptitud del ejecutante.

APÉNDICE 1

CAPÍTULO I ANTECEDENTES

* Las frecuencias a las cuales operaba el Theremin original estaban entre 200 y 600 kHz de la banda de radio frecuencia, el problema es que la frecuencia intermedia de un radio de AM aproximadamente está entre 500 y 600 kHz y el inconveniente con los osciladores locales de los receptores de AM es que generan armónicas que van dentro del campo radial del Theremin, dando como resultado el que se escuchen voces o música.

El Theremin original cuenta con dos antenas, con lo cual se modifica el campo electromagnético por la presencia o ausencia de las manos; a su vez cada antena está ligada a un oscilador local fijo y el otro es un oscilador variable cuya frecuencia varía en función de la presencia de las manos.

Internamente hay una sumatoria de señales entre el oscilador local y el oscilador variable, generándose dos armónicas principalmente una de alta frecuencia y una de baja frecuencia, pero la que nos es útil es la de baja frecuencia, ya que es audible, la otra no se escucha.

En nuestro caso la operación la realiza el DSP y es una señal ultrasónica tipo sonar.

CAPÍTULO II EL PROCESADOR DIGITAL DE SEÑALES dsPIC30

Memoria “X” y memoria “Y”

Introducción

La anchura de los datos de dsPIC30F es de 16-bits. Se organizan todos los registros y memoria del espacio de datos internos como 16-bits de ancho. El dsPIC30F ofrece dos espacios de datos. Los espacios de los datos pueden ser alcanzados por separado (por algunas instrucciones DSP) o junto un rango de dirección lineal de 64-Kbyte (para Instrucciones de MCU). Los espacios de los datos son alcanzados usando dos unidades de la generación de la dirección (AGUs) y trayectorias de datos separados.

Un ejemplo del mapa de memoria del espacio de datos se muestra en la figura 2.3.

Las direcciones de memoria de los datos entre 0x0000 y 0x07FF son reservadas para el dispositivo de registros de función especial (SFRs). El SFRs incluye el control y bits de estado para la CPU y periféricos en el dispositivo.

La RAM, comienza en la dirección 0x0800 y está partida en dos bloques, X e Y del espacio de datos. Para los datos escritos, los espacios de datos X e Y, están alcanzados siempre como sólo, espacio de datos lineales. Para los datos leídos, los espacios de memoria X e Y, se pueden alcanzar independientemente o como sólo, espacio lineal. Los datos leen siempre para instrucciones de clase MCU accediendo a los espacios de datos X e Y como sólo espacio combinado de dos datos. En las instrucciones duales del operando DSP de la fuente, tales como la instrucción MAC, hay acceso al espacio de datos X e Y por separado para simultáneamente apoyar lecturas para las dos fuentes del operando.

Las instrucciones de MCU pueden utilizar cualquier registro de W como indicador de la dirección para los datos leídos u operación escrita.

W10 y W11, son usados como indicadores de dirección para lecturas desde los espacios de datos Y. El resto de los espacios de datos se refieren como espacio X, pero se podrían describir más exactamente como espacio "X menos Y". Se utilizan W8 y W9 mientras que los indicadores de la dirección para los datos leen en el espacio de los datos de X en instrucciones de la clase DSP.

No perder de vista que es el número del registro de W y el tipo de instrucción que se determina como el espacio de dirección y es alcanzado para lecturas de datos. En detalle, las instrucciones de MCU tratan a la memoria de X e Y como sólo espacio combinado de los datos. Las instrucciones de MCU pueden utilizar cualquier registro de W como el indicador de la dirección para lecturas y escritos. Las instrucciones de DSP que pueden simultáneamente el pre-buscar dos operandos de datos, partieron la memoria de los datos en dos espacios. Los registros específicos de W se deben utilizar para leer indicadores de la dirección en este caso.

Algunas instrucciones de DSP tienen la capacidad de almacenar el acumulador que no está dirigido por la instrucción a la memoria de datos. Esta función es llamada "acumulador que escribe detrás". W13 debe ser utilizado como un indicador de dirección al espacio de memoria combinada de los datos para el "acumulador de operaciones escribe detrás".

Las instrucciones de la clase de DSP, W8 y W9 deben señalar al espacio de memoria implementada X para todas las lecturas de memoria. Si W8 o W9 señala al espacio de memoria de Y, los ceros serán devueltos. Si W8 o W9 señala a una dirección de memoria no implementada, una trampa del error de dirección será generada.

Para las instrucciones de la clase de DSP, W10 y W11 deben señalar al espacio de memoria implementada Y para todas las lecturas de memoria. Si W10 o W11 señalan al espacio de memoria implementada X, todos los ceros serán devueltos. Si W10 o W11 señala a una dirección de memoria no implementada, una trampa de error de dirección será generada.

Nota: El mapa de memoria de datos y la partición entre los espacios de datos X y Y es un dispositivo específico.

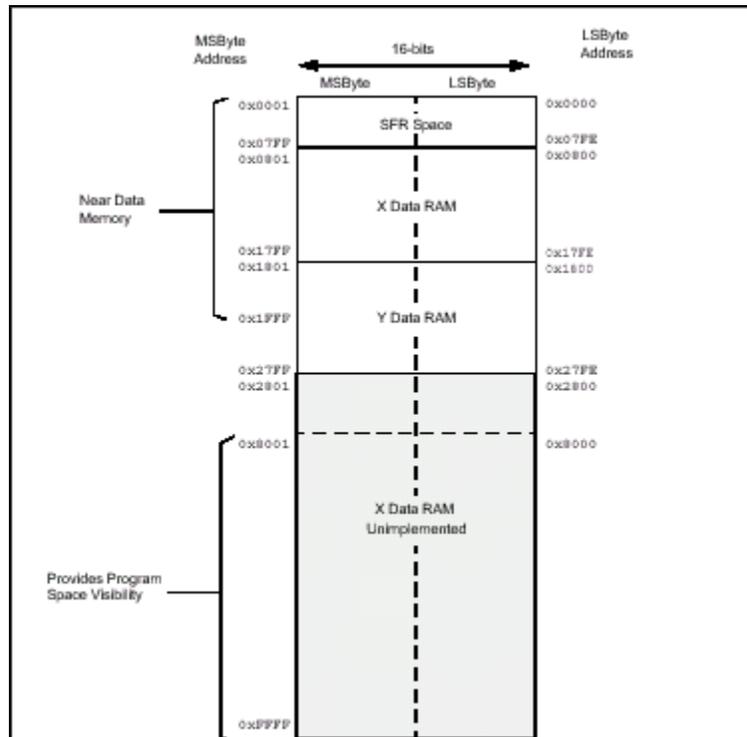


Figura 2.3. Ejemplo de Mapa De Memoria De Datos

Nota 1: La partición entre los espacios de los datos de X y de Y es específico del dispositivo. Los límites del espacio de los datos indicados aquí se utilizan por ejemplo para propósitos únicamente.

2: La memoria cercana de los datos se puede alcanzar directamente mediante las instrucciones del registro del archivo que codifican a unos 13-bit en la dirección en el opcode. En un mínimo, la región de memoria cercana de los datos traslapa todo el espacio SFR y una porción de espacio de memoria de X. Todo el espacio de memoria de X y algo o todo el espacio de memoria de Y, pueden ser incluidos en la región de memoria cercana de los datos, dependiendo de la variante del dispositivo.

3: Toda la memoria de datos se puede alcanzar indirectamente mediante los registros de W o directamente usando la instrucción MOV.

4: La mitad superior del mapa de memoria de datos se puede mapear en un segmento del espacio de la memoria del programa para visibilidad del espacio del programa.

Memoria cercana de los Datos

Un espacio de dirección 8 Kbytes, designado a la memoria cercana de datos, se reserva en el espacio de memoria de datos entre 0x0000 y 0x1FFF. La memoria cercana de datos es directamente direccionable mediante unos 13 bits de campo de dirección absoluta dentro de todas las instrucciones del registro del archivo.

Las regiones de memoria incluidas en la región cercana de los datos dependerán de la cantidad de memoria de los datos implementados para cada variante del dispositivo dsPIC30F. En un mínimo, la región cercana de los datos incluirá todo el SFRs y algo de la memoria de datos de X. Para los dispositivos que tienen cantidades más pequeñas de memoria de datos, la región cercana de datos puede incluir todos los espacios de memoria de X y posiblemente algo o todo el espacio de memoria de Y.

Nota: El espacio entero de los datos 64K puede ser direccionado usando la instrucción MOV.

Unidades del Generador de Dirección del Espacio de Datos (AGUs)

El dsPIC30F contiene un X AGU y una Y AGU para generar direcciones de memoria de datos. Ambos X y Y AGUs pueden generar cualquier dirección efectiva (EA) dentro de un rango de 64 Kbytes. Sin embargo, EAs que está fuera de la memoria física proporcionada volverá todos los ceros para lecturas de datos y datos escritos a esas localizaciones y no tendrá ningún efecto. Además, una trampa del error de dirección será generada.

Unidad del Generador de Dirección de X

El X AGU es utilizado por todas las instrucciones y apoya todos los modos de direccionamiento. El X AGU consiste de una lectura de AGU (X RAGU) y de una escritura de AGU (X WAGU), que funcionan independientemente en separado con buses de lectura y escritura durante diversas fases del ciclo de la instrucción. El bus de datos de lectura X es la trayectoria de datos de regreso para todas las instrucciones que son vistas en el espacio de datos como espacio de dirección combinada de X y de Y. Es también la trayectoria de datos del espacio de dirección de X para las instrucciones de lectura del operando dual (clase de instrucción de DSP). X escribe el bus de datos y es el único que describe la trayectoria para el espacio de los datos combinados de X y de Y para todas las instrucciones.

El X RAGU comienza su cálculo efectivo de dirección durante el ciclo previo de la instrucción, usando información derivada de la instrucción exacta pre-buscada. El X RAGU EA es presentado al bus de dirección al principio del ciclo de la instrucción.

El X WAGU comienza su cálculo efectivo de dirección al principio del ciclo de la instrucción. El EA es presentado al bus de dirección durante la fase de escritura de la instrucción.

El X RAGU y el X WAGU apoyan el direccionamiento del módulo.

El direccionamiento Bit-Invertido solamente es apoyado por el X WAGU.

Unidad del Generador de Dirección de Y

La memoria de los datos de Y tiene un AGU que apoyan las lecturas de datos del espacio de memoria de datos de Y. El bus de memoria de Y nunca se utiliza para los datos escritos. La función del bus de la memoria de Y AGU y de Y es apoyar datos de lectura concurrentes para las instrucciones de la clase de DSP.

La sincronización de Y AGU es idéntica a la del X RAGU, en que su cálculo efectivo de dirección comienza antes del ciclo de la instrucción, usando la información derivada de la instrucción pre-buscada. El EA es presentado al bus de dirección al principio del ciclo de la instrucción.

La Y AGU apoya los modos de direccionamiento del módulo y de direccionamiento de la Post-modificación para las clases de DSP de las instrucciones que la utilizan.

Nota: La Y AGU no apoya datos escritos. Todos los datos escritos ocurren mediante el X WAGU a espacios de datos combinados de X y de Y. La Y AGU se utiliza solamente durante datos que leen para instrucciones DSP de operandos de la fuente dual.

Unidades del generador de dirección e instrucciones de la clase de DSP

La trayectoria de datos de la memoria de Y AGU y de Y es utilizada en concreto con el X RAGU por la clase de DSP de instrucciones para proporcionar dos trayectorias de lectura de datos concurrentes. Por ejemplo, la instrucción MAC puede simultáneamente pre-alcanzar dos operandos en la multiplicación siguiente.

La clase de DSP de instrucciones dedica dos indicadores del registro de W, W8 y W9, para funcionar siempre con el X RAGU y espacio de datos de la dirección X independientemente del espacio de los datos de Y, más dos indicadores del registro W, W10 y W11, para funcionar siempre con el Y AGU y direcciones del espacio de los datos de Y independientemente de espacio de los datos de X. Cualquier dato escrito realizado por una instrucción de la clase DSP que tomará lugar en el espacio de los datos combinados de X y de Y y el escribir ocurrirá a través del bus X. Por lo tanto, el escribir puede estar en cualquier dirección sin consideración de donde se dirige el EA.

La Y AGU apoya solamente los modos de dirección de la Post-modificación asociados a la clase de instrucciones de DSP. La Y AGU también apoya el direccionamiento del módulo para buffers circulares automatizados. El resto de las instrucciones de la clase (MCU) pueden tener acceso al espacio de dirección de los datos de Y con el X AGU cuando es considerado como parte del espacio lineal compuesto.

Alineación de los Datos

El ISA apoya tanto a las operaciones de la palabra como del byte para todas las instrucciones de MCU que tengan acceso a datos con la memoria AGU de X. El LSB de una dirección de 16 bits de los datos es ignorado para las operaciones de la palabra. Los datos

de la palabra se alinean en el formato little-endian con el LSByte incluso la dirección (LSB = 0) y el MSByte en la dirección impar (LSB = 1).

En las operaciones del byte, el LSB de la dirección de datos se utiliza para seleccionar el byte que es alcanzado. El byte de dirección se pone en los 8 bits más bajos del bus de datos internos.

Todos los cálculos efectivos de dirección se ajustan automáticamente dependiendo de si un byte o un acceso de palabra se realiza. Por ejemplo, una dirección será incrementada por 2 para una operación de palabra que post-incremente el indicador de la dirección.

Nota: Todos los accesos de la palabra se deben alinear incluso a una dirección (LSB = 0). Para una palabra mal alineada los alcances de los datos no se apoyan, así que el cuidado debe ser tomado al mezclar byte y operaciones de palabra o traduciendo del código existente de PICmicro. Si una palabra mal alineada de lectura o escritura están atentadas, una trampa del error de dirección ocurrirá. Una operación de lectura mal alineada terminará, pero una escritura mal alineada no ocurrirá. La trampa entonces será tomada, permitiendo que el sistema examine el estado de la máquina antes de la ejecución de la falla de dirección.

Direccionamiento del Módulo

El módulo, o direccionamiento circular proporciona medios automatizados para apoyar buffers circulares de datos y usar el hardware. El objetivo es quitar la necesidad del software para realizar el límite de la dirección de los datos y revisar al ejecutar código firmemente colocado al igual que el típico en muchos algoritmos de DSP.

Cualquier registro de W, excepto W15, puede ser seleccionado como el indicador al buffer del módulo. El hardware del módulo realiza revisiones de límite en la dirección llevada a cabo en el registro seleccionado de W y ajusta automáticamente el valor del puntero en los límites del buffer, cuando es requerido.

El direccionamiento del módulo de dsPIC30F puede funcionar en datos o espacio del programa (desde los datos el mecanismo del indicador es esencialmente igual para ambos). Un buffer circular puede estar apoyado en cada uno de los espacios de los datos de X (que también proporciona los indicadores dentro del espacio del programa) y de Y.

La longitud del buffer de los datos del modulo puede ser de cualquier tamaño hasta palabras de 32K. La lógica del buffer del módulo apoya buffers usando palabra o datos del tamaño del byte. Sin embargo, la lógica del módulo realiza solamente revisiones del límite de la dirección en los límites de la dirección de la palabra, así que la longitud de un buffer del módulo del byte debe ser uniforme. En adición, los buffers del módulo byte-sized no pueden ser implementados usando la Y AGU porque el acceso del byte no se apoya por el bus de datos de la memoria de Y.

2.2. PERIFÉRICOS INTERNOS

2.2.1. TIMER (TEMPORIZADOR)

Esta sección describe el módulo de 16-bits del Timer1 de propósitos generales y modos operacionales asociados.

La figura 2.4 representa el diagrama de bloque simplificado de los 16 bits del Módulo Timer1.

Nota: Timer1 es ‘contador de tiempo del tipo A’.

El módulo Timer1, es un contador de tiempo de 16 bits que puede servir como el contador del tiempo para el reloj en tiempo real, o funciona como intervalo corriente libre timer / counter. El contador de tiempo de 16 bits tiene los modos siguientes:

- Contador de tiempo de 16 bits
- Contador Síncrono de 16 bits
- Contador Asíncrono de 16 bits

Además, se apoyan las características operacionales siguientes:

- Operación de la puerta del contador de tiempo
- Ajustes seleccionables del prescaler
- Operación del contador de tiempo durante modos idle y sleep de la CPU.

Interrupción del complemento del registro del período en 16-bits o el borde de descenso de la señal externa de la puerta.

El modo de funcionamiento es determinado fijando bit(s) apropiado en el SFR 16-bit, T1CON. La figura 2.4 presenta un diagrama de bloque de módulo 16-bit del contador de tiempo.

Modo de 16 bits del Contador de Tiempo

En el modo del contador de tiempo de 16 bits, los incrementos de este contador de tiempo en cada ciclo de la instrucción van hasta un valor del complemento, cargado en el PR1 del registro del período, entonces los reajustes a 0 continúan contando.

Cuando la CPU entra en modo idle, el contador de tiempo parará el incremento, a menos que el TSIDL(T1CON<13>) bit = 0. Si TSIDL = 1, la lógica del módulo del

contador de tiempo retomará la secuencia de incremento sobre la terminación del modo idle de la CPU.

Modo de Contador Síncrono de 16 bits

En el modo de contador síncrono de 16 bits, los incrementos del contador de tiempo en el borde del levantamiento de la señal externa aplicada al reloj se sincronizan con los relojes internos de la fase.

El contador de tiempo cuenta hasta un valor del complemento cargado en PR1, entonces el reajuste a 0 continúa.

Cuando la CPU va dentro del modo idle, el contador de tiempo parará el incremento, a menos que el respectivo de TSIDL bit = 0. Si TSIDL = 1, la lógica del módulo del contador de tiempo retomará la secuencia de incremento sobre la terminación del modo lento de la CPU.

Modo de Contador Asíncrono de 16 bits

En el modo de contador asíncrono de 16 bits, los incrementos del contador de tiempo en cada borde de levantamiento de la señal externa aplicada al reloj, el contador de tiempo cuenta hasta un valor del complemento cargado en PR1, entonces los reajustes a 0 continúan.

Cuando el contador de tiempo se configura para el modo de operación asíncrona y la CPU va dentro del modo idle, el contador de tiempo parará el incremento si TSIDL = 1.

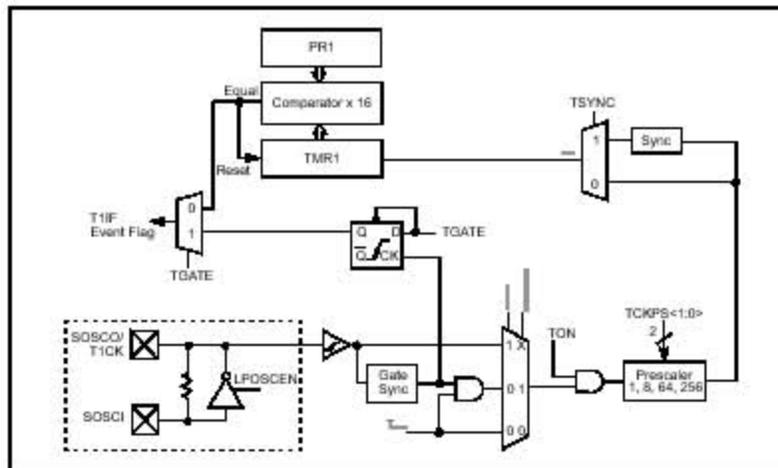


Figura 2.4. Diagrama de Bloque del Módulo de 16 bits Timer 1(Contador de Tiempo Tipo A)

Operación del Temporizador durante el Modo Sleep

Durante el modo sleep de la CPU, el contador de tiempo funcionará si:

- El módulo del contador de tiempo es habilitado (TON = 1) y
- La fuente del reloj del contador de tiempo se selecciona como externa (TCS = 1) y
- El bit de TSYNC (T1CON<2>) se afirma a una lógica 0, que define la fuente externa del reloj como asincrónico.

Cuando las tres condiciones son verdaderas, el contador de tiempo continuará contando hasta el registro del período y siendo reajustado a 0x0000.

Cuando ocurre un complemento entre el contador de tiempo y el registro de período, una interrupción se puede generar, si se mantiene el bit respectivo habilitado de la interrupción del contador de tiempo.

Interrupción del Temporizador

El contador de tiempo de 16-bit tiene la habilidad para generar una interrupción en el complemento del período. Cuando la cuenta del contador de tiempo empareja el registro del período, el bit T1IF se afirma y una interrupción será generada, si está permitida. El bit T1IF debe ser definido en software. La bandera de interrupción del contador de tiempo T1IF está situada en el registro del control IFS0 en el regulador de interrupción.

Cuando el modo de Acumulación del tiempo de bloqueo es habilitado, una interrupción también será generada en el borde que cae de la señal de la puerta (en el extremo del ciclo de la acumulación).

Habilitar una interrupción se logra mediante el bit habilitado de la interrupción del contador de tiempo, T1IE. La interrupción del contador de tiempo habilita el bit que está situado en el registro de control IEC0 en el regulador de interrupción.

Reloj en Tiempo Real

Timer1, al funcionar en modo del reloj en tiempo real (RTC), proporciona capacidades de time-of-day y de tiempo del acontecimiento. Las características operacionales claves del RTC son:

- Operación a partir del oscilador de 32 KHz LP
- Prescaler de 8 bits
- Energía baja
- Interrupciones del reloj en tiempo real

Estos modos de operación son determinados fijando el bit (s) apropiado en el registro de control T1CON.

Operación del Oscilador RTC

Cuando la TON = 1, TCS = 1 y TGATE = 0, los incrementos del contador de tiempo en el borde de levantamiento de la señal de salida del oscilador de 32 KHz LP, van hasta el valor especificado en el registro del período, y es entonces reajustado a '0'.

El bit de TSYNC se debe afirmar a un '0' lógico (Modo Asíncrono) para la operación correcta.

Permitir LPOSCEN (OSCCON<1>) inhabilitará el contador de tiempo normal y los modos de contador y permitirá un contador de tiempo carry-out del wake-up.

Cuando la CPU entra en modo sleep, el RTC continuará para operar, con tal de que el cristal externo de 32 KHz del oscilador se active y los bits de control no sean cambiados. El bit TSIDL debe ser definido para '0' en orden para que RTC continúe la operación en modo idle.

Interrupciones de RTC

Cuando un suceso de interrupción ocurre, la bandera de interrupción respectiva, T1IF es mantenida y una interrupción será generada, si está permitido. El bit T1IF debe ser definido en software.

La bandera de interrupción respectiva del control de tiempo, T1IF, es localizada en el registro de estado IFS0 en el regulador de interrupción.

Permitir una interrupción se logra mediante el bit habilitado de la interrupción del contador de tiempo, T1IE. La interrupción del contador de tiempo habilita el bit que está situado en el registro del control IEC0 en el regulador de interrupción.

Módulo timer 2 / 3

Esta sección describe el módulo del control de tiempo (Timer 2 / 3) de propósitos generales (GP) de 32 bits y modos operacionales asociados.

El módulo Timer 2 / 3 es un contador de tiempo de 32-bit, que puede ser configurado como dos contadores de tiempo de 16-bit, con modos de funcionamiento seleccionable. Estos contadores de tiempo son utilizados por otros módulos periféricos por ejemplo:

- Captura de la entrada
- Salida Compare / Simple PWM

El contador de tiempo de 32 bits tiene los modos siguientes:

- Dos contadores de tiempo de 16-bit independientes (Timer 2 y Timer 3) con todos los modos de funcionamiento de 16-bit (excepto modo de contador asíncrono)
- Sólo operación de 32 bits del contador de tiempo
- Sólo contador síncrono de 32 bits

Además, las características operacionales siguientes están apoyadas:

- Disparador del acontecimiento ADC
- Operación de la puerta del contador de tiempo
- Ajustes seleccionables del prescaler
- Operación del contador de tiempo durante modos idle y sleep
- Interrupción en un complemento de 32 bits del registro del período

Este modo de funcionamiento es determinado fijando el bit (s) apropiado en el T2CON y el T3CON SFRs de 16-bit.

Para la operación de 32 bits timer / counter , Timer2 es la palabra del LS y Timer3 es la palabra MS del contador de tiempo de 32 bits.

Nota: Para la operación de 32 bits del contador de tiempo, los bits de control T3CON son ignorados. Solamente los bits de control T2CON se utilizan para la disposición y el control. El reloj Timer2 y las puertas de entrada son utilizados para el módulo de 32-bit del contador de tiempo, pero una interrupción es generada con la bandera de interrupción Timer3 (T3IF) y la interrupción es habilitada con la interrupción Timer3 que habilita el bit (T3IE).

Modo de 16 bits

En el modo de 16 bits, Timer2 y Timer3 pueden ser configurados como dos contadores de tiempo independientes de 16-bits cada contador de tiempo puede ser instalado en cualquiera de los dos modos de contador de tiempo de 16-bit o modo de contador síncrono de 16-bit.

La única diferencia funcional entre Timer2 y Timer3 es que Timer2 proporciona la sincronización de la salida del prescaler del reloj. Esta es útil para las entradas de reloj externas de alta frecuencia.

Modo de 32 bits del Contador de tiempo

En el modo de 32-bit del contador de tiempo, los incrementos del contador de tiempo en cada instrucción completan un ciclo hasta un valor del complemento, cargado en el registro de período de 32-bit combinado PR3/PR2 , entonces los reajustes a 0 continúan contando.

Para lecturas síncronas de 32-bit del par Timer2/Timer3, leyendo la palabra del LS (registro TMR2) causará la palabra MS que se lee y se parará en un registro de 16-bit, llamado TMR3HLD.

Para escritura síncrona de 32-bit, el registro de (TMR3HLD) debe primero ser escrito. Cuando seguido por un escrito al registro TMR2, los contenidos de TMR3HLD serán transferidos y detenidos en el MSB del contador de tiempo 32-bit (TMR3).

Modo de Contador Síncrono de 32 bits

En el modo de contador síncrono de 32-bit, los incrementos del contador de tiempo en el borde de levantamiento de la señal externa aplicada al reloj, se sincronizan con los relojes internos de la fase.

El contador de tiempo cuenta hasta un valor del complemento cargado en el registro de período de 32-bit combinando PR3/PR2, entonces los reajustes a '0' continúan. Cuando el contador de tiempo se configura para el modo de operación de contador síncrono y la CPU entra en modo idle, el contador de tiempo parará el incremento, a menos que el TSIDL (T2CON<13 >) bit = 0. Si TSIDL = 1, el módulo del contador de tiempo lógico retomará la secuencia de incremento sobre la terminación del modo lento de la CPU.

Operación del Temporizador durante Modo Sleep

Durante el modo sleep de la CPU, el contador de tiempo no funcionará, porque los relojes internos están inhabilitados.

Interrupción del Temporizador

El módulo de 32 bits del contador de tiempo puede generar una interrupción en el complemento del período, o en el borde que cae de la señal externa de la puerta. Cuando la cuenta de 32 bits del contador de tiempo empareja el registro de período de 32 bits respectivos, o el borde cayendo y la señal externa de la "puerta" es detectada, se mantiene el bit de T3IF (IFS0<7 >) y una interrupción será generada si está permitido. En este modo, la bandera de interrupción de T3IF se utiliza como la fuente de la interrupción. El bit de T3IF debe ser definido en software.

El permitir una interrupción se logra mediante el contador de tiempo respectivo que el bit de interrupción, T3IF (IEC0<7 >), permite.

Idle mode

Modo inactivo. Cuando la CPU, está inactiva el periférico puede seguir generando trabajos de PWM.

Nota: Para Sleep Mode, todos los relojes del sistema están deshabilitados.

Para Idle Mode, Todos los relojes están funcionando, pero la CPU está parada.

Módulo de captura de entrada

Esta sección describe el módulo de la captura de entrada y modos operacionales asociados. Las características proporcionadas por este módulo son útiles en las aplicaciones que requieren frecuencia (período) y medida del pulso. La figura 2.5, representa un diagrama de bloque del módulo de captura de entrada.

La captura de entrada es útil para los modos tales como:

- Medidas de Frecuencia / Periodo / Pulso
- Fuentes adicionales de interrupciones externas

Las características operacionales dominantes del módulo de captura de entrada son:

- Modo simple de evento de captura
- Modo de selección Timer2 y Timer3
- Interrupción del evento de captura de entrada

Estos modos de funcionamiento son determinados fijando los bits apropiados en el registro ICxCON (donde $x = 1, 2, \dots, N$). Los dispositivos del dsPIC contienen la captura hasta 8 canales, (es decir, el valor máximo de N es 8).

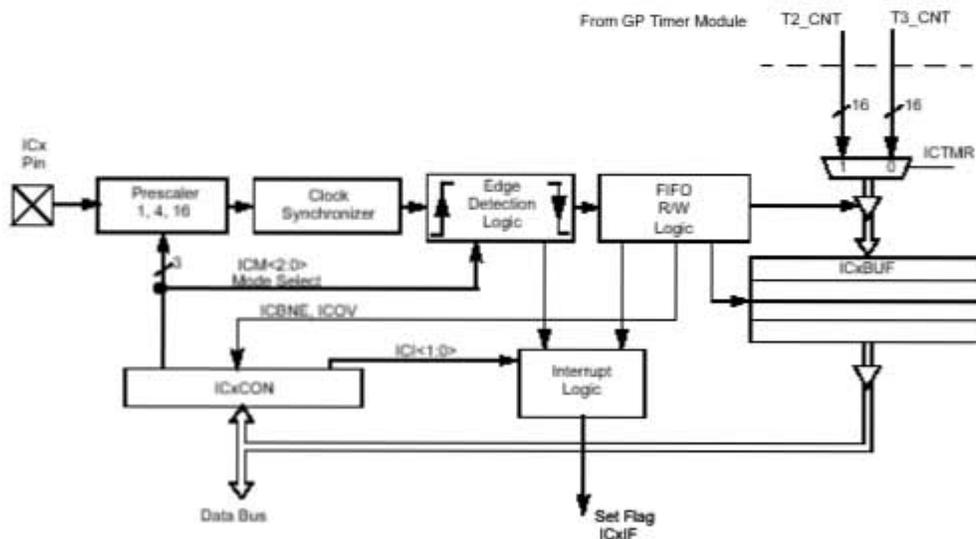


Figura 2.5. Diagrama de Bloques del Módulo de Captura de Entrada.

2.2.2. EL CONVERTIDOR ANALÓGICO/DIGITAL

Una conversión analógica-digital consiste en la transcripción de señales analógicas en señales digitales, con el propósito de facilitar su procesamiento (encriptación, compresión, etc.) y hacer la señal resultante (la digital) más inmune al ruido y otras interferencias a las que son más sensibles las señales analógicas.

Una señal analógica es aquella que puede tomar una infinidad de valores (frecuencia y amplitud) dentro de un límite superior e inferior.

En cambio, una señal digital es aquella señal cuyos valores (frecuencia y amplitud) no son continuos sino discretos, lo que significa que la señal necesariamente ha de tomar unos determinados valores fijos predeterminados. Estos valores fijos se toman del sistema binario, lo que significa que la señal va a quedar convertida en una combinación de ceros y unos, que ya no se parece en nada a la señal original. Precisamente, el término digital tiene su origen en esto, en que la señal se construye a partir de números (dígitos).

Módulo del Convertidor (A/D) de Alta Velocidad de 10 bits

El convertidor de analógico a digital de alta velocidad de 10 bits (A/D) permite la conversión de una señal de entrada análoga a un número digital de 10 bits. Este módulo está basado en una arquitectura sucesiva del registro de la aproximación (SAR), y proporciona un índice máximo del muestreo de 500 ksp. El módulo A/D tiene hasta 16 entradas análogas las cuales están multiplexadas dentro de 4 muestras y amplificadores. La salida de la muestra y del asimiento son la entrada en el convertidor, que genera el resultado. Los voltajes análogos de la referencia son software seleccionable para cualquier voltaje del dispositivo (AV_{DD}/AV_{SS}) o el nivel voltaico en el pin de (V_{REF+}/V_{REF-}). El convertidor A/D, tiene una característica única de poder funcionar mientras que el dispositivo está en modo sleep.

El módulo A/D tiene seis registros de 16 bits:

- Control A/D Registro1 (ADCON1)
- Control A/D Registro2 (ADCON2)
- Control A/D Registro3 (ADCON3)
- Registro Selecto de la entrada A/D (ADCCHS)
- Registro de la configuración de Puerto (ADPCFG)
- Registro de la entrada de la selección del explorador (ADCSSL)

El ADCON1, ADCON2 y los registros ADCON3 controlan la operación del módulo A/D. El registro de ADCHS selecciona los canales de entrada para ser convertidos.

El registro de ADPCFG configura los pines portuarios como entradas analógicas o como I/O digital. El registro de ADCSSL selecciona las entradas para la exploración.

Nota: SSRC<2:0>, ASAM, SIMSAM, SMPI<3:0>, los bits de BUFM y de ALTS, así como los registros de ADCON3 y de ADCSSL, no deben ser escritos mientras que ADON = 1. Esto conducirá resultados indeterminados.

El diagrama de bloque de un módulo A/D de 12 bits se muestra en la figura 2.6

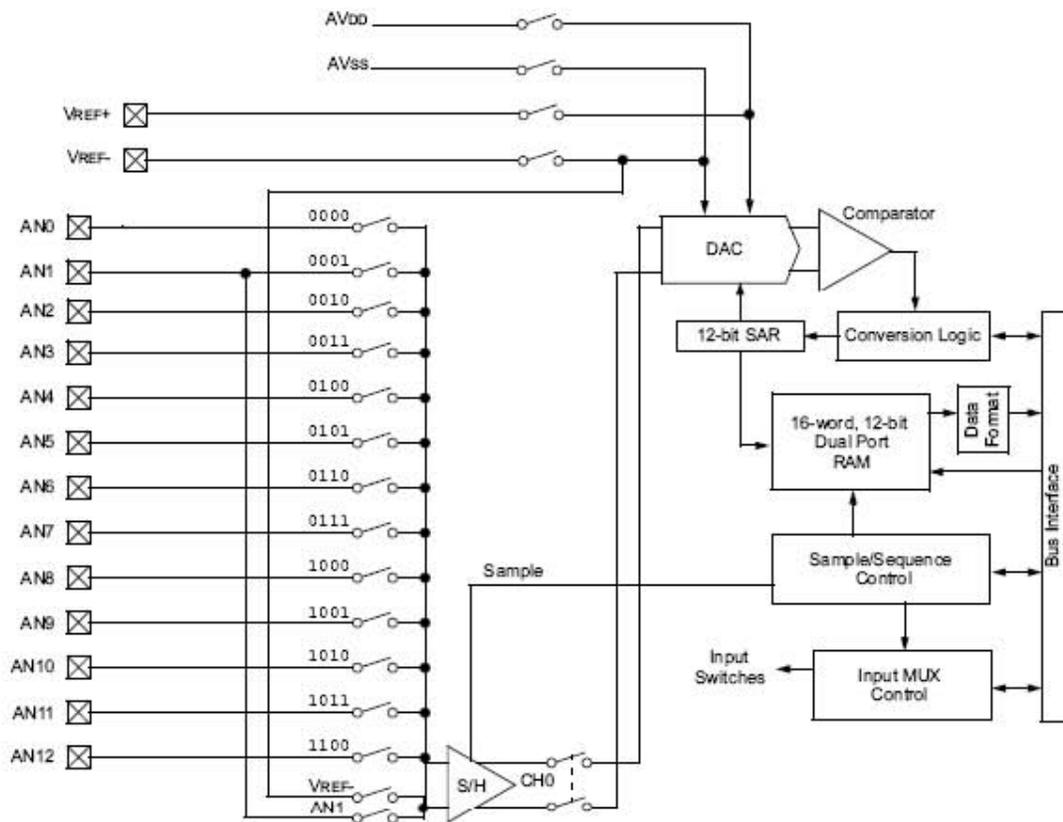


Figura 2.6. Diagrama de bloque funcional A/D de alta velocidad en este caso de 12 bits

Operación de la Conversión

Después de que se haya configurado el módulo A/D, la adquisición de la muestra es iniciada fijando el bit de SAMP. Varias fuentes, tales como un bit programable, descansos del contador de tiempo y acontecimientos externos, terminarán la adquisición y comenzarán una conversión.

Cuando la conversión A/D se completa, el resultado se carga en ADCBUF0... ADCBUFF, la bandera de interrupción ADIF del A/D y el bit DONE se fijan después del número de las muestras especificadas por el bit de SMPI.

Los pasos siguientes se deben seguir para hacer una conversión A/D:

1. Configurar el módulo A/D:

- Configurar los pines analógicos, la referencia del voltaje e I/O digital
- Seleccionar canales de entrada A/D
- Seleccionar reloj de conversión A/D
- Seleccionar disparador de conversión A/D
- Girar el módulo A/D

2. Configurar la interrupción A/D (se requiere):

- Bit claro de ADIF
- Seleccionar la prioridad de interrupción A/D

3. Comenzar a muestrear

4. Esperar el tiempo requerido de la adquisición

5. Accionar el extremo de la adquisición, comenzar la conversión

6. Esperar la conversión A/D para terminar, por cualquier cosa:

- Esperar la interrupción A/D

7. Leer el buffer del resultado A/D, ADIF claro si está requerido

Operación A/D durante los Modos Sleep e Idle de la CPU

Operación A/D durante el Modo Sleep de la CPU

Cuando el dispositivo entra en modo sleep en todas las fuentes del reloj al módulo se mantienen detenidas y permanecen en el '0' lógico.

Si el descanso ocurre en el centro de una conversión, se termina la conversión. El convertidor no continuará con una conversión parcialmente terminada en la salida del modo sleep.

El contenido del registro no es afectado por el dispositivo que entra en o que sale del modo sleep.

El módulo A/D puede operar durante el modo sleep si la fuente A/D del reloj se fija a RC (ADRC = 1). Cuando la fuente del reloj de RC es seleccionada, el módulo A/D, espera un ciclo de instrucción antes de comenzar la conversión. Esto permite que la instrucción del SLEEP sea ejecutada y que elimine todo el ruido digital de la conmutación de la conversión.

Cuando termina la conversión, el bit DONE será fijado y el resultado será cargado en el registro ADCBUF.

Si la interrupción A/D es habilitada el dispositivo regresará del descanso. Si la interrupción A/D no se habilita el módulo A/D entonces será apagado, aunque el bit de ADON seguirá siendo determinado.

Operación A/D durante el Modo Idle de la CPU

El bit de ADSIDL selecciona si el módulo detecta o se detiene en modo lento o bien continuará en modo lento. Si ADSIDL = 0, el módulo continuará la operación en la aserción del modo lento. Si ADSIDL = 1, el módulo parará en modo lento.

Efectos de un reajuste

Un dispositivo reajusta fuerzas a todos los registros para su estado de reajuste. Esto fuerza el módulo A/D para ser apagado, y alguna conversión y secuencia de adquisición es abortada. Los valores que están en los registros ADCBUF no se modifican. El registro A/D del resultando contendrá datos desconocidos después de una energía en Reajuste.

Formatos de la Salida

El resultado es 10-bit extenso. La RAM del buffer de datos es también de 10-bits extensos. Los datos de 10-bit pueden ser leídos en uno o cuatro diversos formatos. Los bits FORM <1:0> seleccionan el formato. Cada uno de los formatos de la salida traducen a resultados de 16 bits en el bus de datos.

Escribir los datos siempre será en formato justificado.

2.2.3. PUERTOS DE ENTRADA/SALIDA (I/O)

Todos los pines del dispositivo (excepto V_{DD} , V_{SS} , \overline{MCLR} y OSC1/CLKIN) se comparten entre los periféricos y los puertos paralelos de I/O.

Todas las entradas de I/O de características de entradas del Schmitt Trigger son para mejorar la inmunidad del ruido.

Puertos paralelos de I/O (PIO)

Cuando un periférico está habilitado y el periférico está conduciendo activamente un pin asociado, el uso del pin como un pin de salida de fines generales es inhabilitado. El pin de I/O se puede leer, pero el conductor de la salida para el bit de puerto paralelo será inhabilitado. Si un periférico es habilitado, solamente el periférico no está conduciendo activamente un pin y el pin puede ser conducido por un puerto.

Todos los pines portuarios tienen tres registros asociados directamente a la operación del pin portuario. El registro de dirección de los datos (TRISx) determina si el pin es una entrada o una salida. Si el bit de la dirección de los datos es un '1', entonces el pin es una entrada. Todos los pines portuarios se definen como entradas después de un reajuste.

Cualquier bit y sus datos asociados y los registros de control que son inválidos para un dispositivo particular serán inhabilitados.

Un puerto paralelo I/O (PIO) que comparte un pin con un periférico es en general, servicial al periférico.

Los datos del buffer de la salida del periférico y las señales de control son proporcionadas a un par de multiplexores. Los multiplexores seleccionan si el periférico o el puerto asociado tienen propiedad de los datos de salida y de las señales de control la celda de I/O.

La figura 2.7 muestra como los puertos se comparten con otros periféricos, y la celda asociada de I/O a la cual están conectadas.

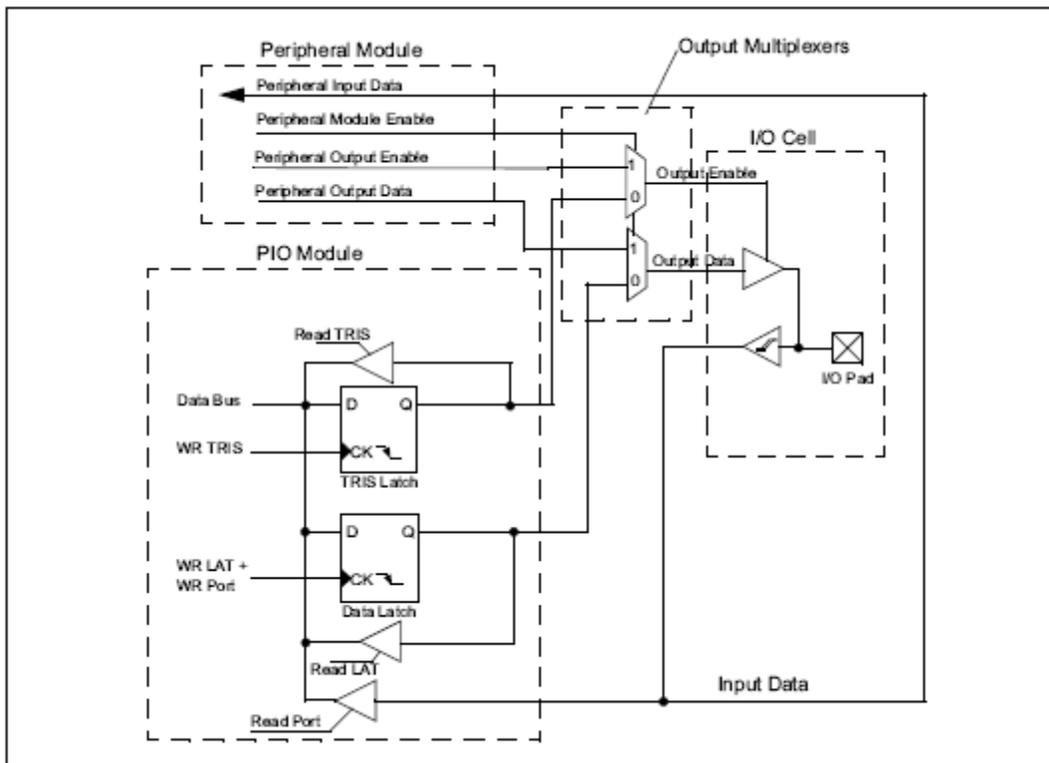


Figura 2.7. Diagrama de Bloque de una Estructura Portuaria Compartida

2.2.4. Módulo SPI(SERIAL PERIPHERAL INTERFACE)

El módulo de interfase de serie de periféricos (SPI) es una interfase en serie síncrona. Es útil para comunicarse con otros dispositivos periféricos tales como EEPROMs, registros de cambio, conductores de la exhibición y convertidores A/D, y otros microcontroladores. Es compatible con las interfases de SPI y de SIOP de Motorola.

Operación de SPI durante el Modo Sleep de las CPU

Durante el modo sleep, el modo de SPI es detenido. Si la CPU entra en modo sleep mientras que una transacción de SPI está en marcha, entonces se terminan la transmisión y la recepción.

El transmisor y el receptor pararán en modo sleep. Sin embargo, los contenidos del registro no son afectados por entradas o salidas del modo sleep.

Operación de SPI durante el Modo Idle de la CPU

Cuando se da el modo idle en las entradas del dispositivo, todas las fuentes del reloj siguen funcionando. El bit de SPISIDL (SPIxSTAT<13>) selecciona si el módulo de SPI continuará inactivo. Si SPISIDL = 0, el módulo continúa funcionando cuando la CPU entra en modo idle. Si SPISIDL = 1, el módulo se detendrá cuando la CPU entra en este modo.

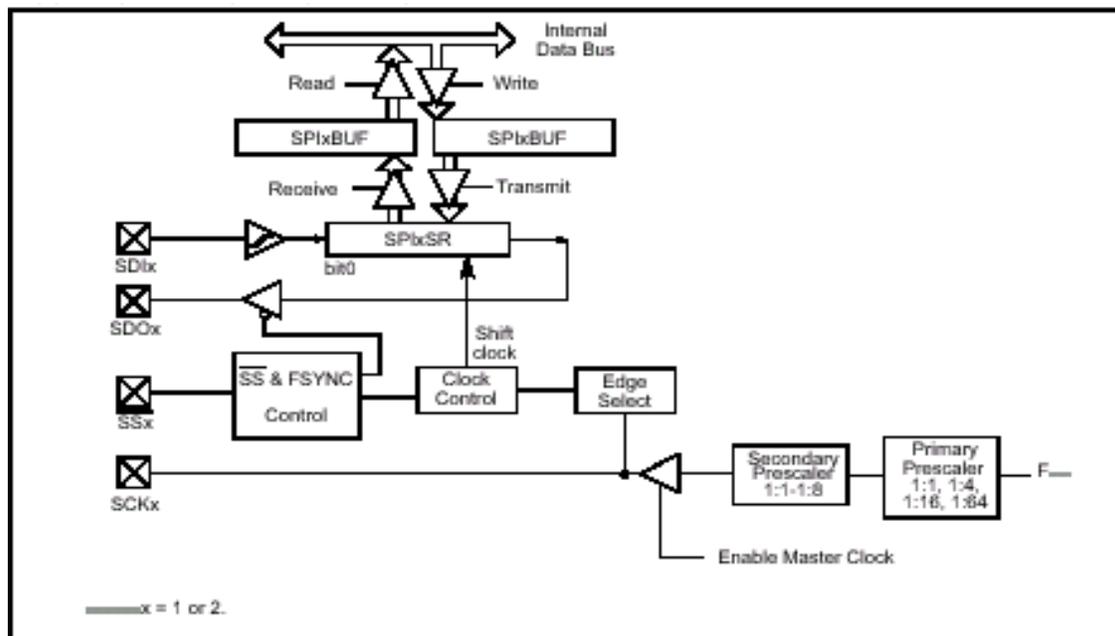


Figura 2.8. Diagrama de Bloque del SPI

2.2.5. Módulo de I²C

El módulo del circuito (I²C) proporciona la ayuda completa del hardware para los modos del esclavo y del maestro del estándar de la comunicación en serie de I²C con una interfase de 16 bits.

Este módulo ofrece las características siguientes:

- Apoyo de la interfase I²C tanto para operaciones del maestro y del esclavo
- Apoyo al modo esclavo de I²C para 7 y 10-bit de dirección
- Apoyo al modo maestro de I²C para 7 y 10-bit de dirección
- El puerto de I²C no prohíbe transferencias bidireccionales entre maestros y esclavos

- La sincronización serial del reloj para el puerto de I²C se puede utilizar como mecanismo de presión para suspender y reanudar la transferencia serial (control de SCLREL).
- I²C auxiliar de operación multimaestra; detecta la colisión del bus.

Descripción de la Función de Operación

El hardware completamente implementa todas las funciones del maestro y del esclavo y especificaciones rápidas del I²C, así como el direccionamiento de 7 a 10 bits. Así, el módulo de I²C puede operar como un esclavo o un maestro en un bus de I²C.

Varios Modos de I²C

Los siguientes son tipos de operación de I²C:

- Operación de esclavo de I²C con la dirección de 7 bits
- Operación de esclavo de I²C con la dirección de 10 bits
- Operación maestra de I²C con dirección de 7 ó 10 bits

Registros de I²C

El I2CCON e I2CSTAT son registros de estado y control respectivamente. Los 6 bits más bajos de I2CSTAT se leen solamente. Los bits restantes del I2CSTAT son de lectura/escritura.

I2CRSR, es el registro de cambio usado para cambiar datos, mientras que I2CRCV es el registro del buffer al cual los bytes de los datos son escritos o los bytes de datos son leídos. I2CRCV, es el buffer de recepción.

I2CTRN, es el registro de transmisión del cual los bytes son escritos durante una operación de la transmisión según lo mostrado en la figura 2.9.

El registro de I2CADD, lleva a cabo la dirección auxiliar o esclava. Un bit de estado, ADD10, indica el modo de dirección de 10-bit. El I2CBRG, actúa como el valor de la recarga del generador de velocidad.

En operaciones recibidas, I2CRSR e I2CRCV forman un receptor protegido doble. Cuando I2CRSR, recibe un byte completo se transfiere a I2CRCV y se genera un pulso de la interrupción. Durante la transmisión, el I2CTRN, no está protegido.

Nota: Siguiendo una condición del reinicio en el modo de 10 bits, el usuario necesita solamente emparejar los primeros 7 bits de dirección.

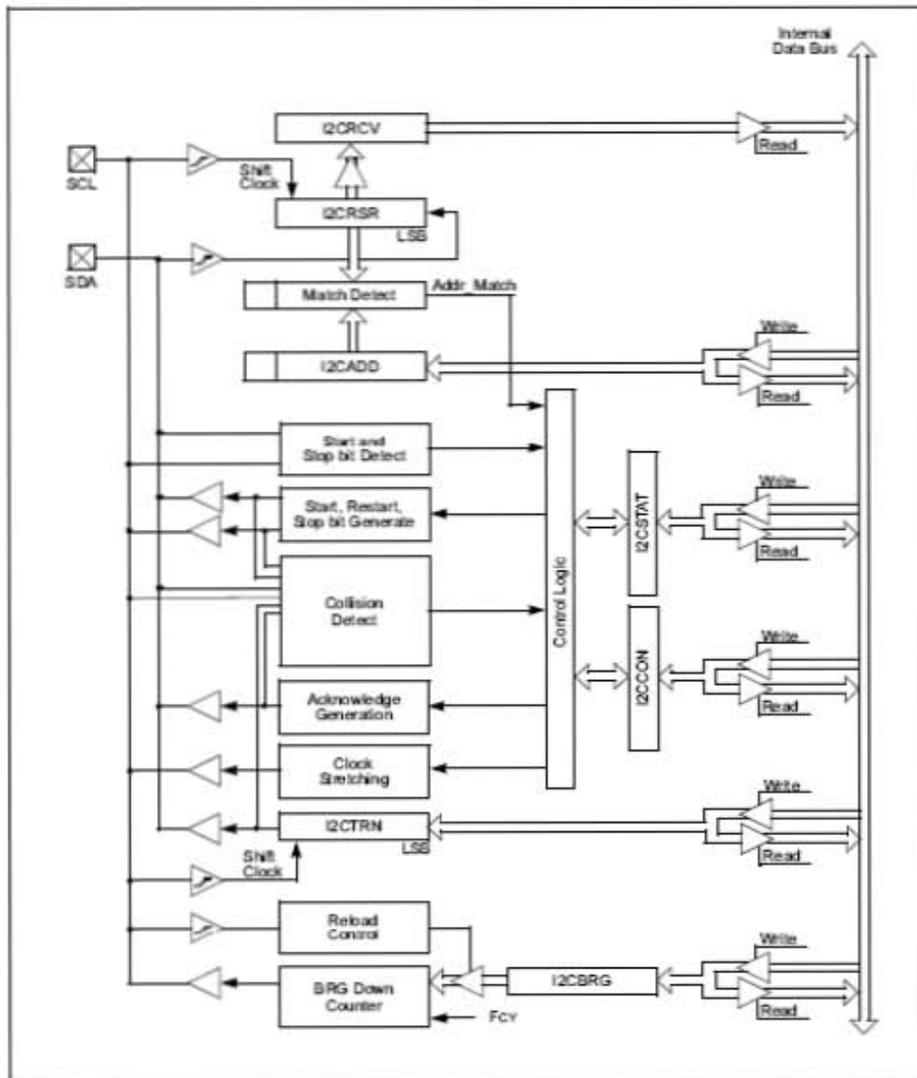


Figura 2.9. Diagrama de bloque de I²C

Interrupciones

Los módulos I²C, generan dos banderas de interrupción, MI2CIF (bandera de interrupción principal de I²C) y SI2CIF (bandera de interrupción auxiliar de I²C). La bandera de interrupción de MI2CIF se activa en la terminación de un acontecimiento principal del mensaje. La bandera de interrupción de SI2CIF, es activada en la detección de un mensaje dirigido al esclavo o auxiliar.

Operación principal de I²C

Los dispositivos principales generan todos los pulsos seriales de reloj y las condiciones del comienzo y fin. Una transferencia es finalizada con una condición de

término o con una condición de repetición del inicio. Puesto que la condición de repetición del inicio es también el principio de la transferencia serial siguiente, el bus de I²C no será lanzado.

En el modo transmitir maestro, los datos seriales son salidas con SDA, mientras que los relojes seriales con salidas SCL, el primer byte transmitido contiene las direcciones auxiliares del dispositivo de recepción (7 bits) y del bit de la dirección de los datos. En este caso, el bit de la dirección de los datos (R_W) es un 0 lógico. Los datos seriales son transmitidos 8 bits a la vez. Después de cada byte que se transmite, se recibe un bit del

ACK. Las condiciones del inicio y de término están afuera para indicar el principio y el fin de una transferencia serial. En el modo de receptor maestro, el primer byte transmitido contiene la dirección auxiliar del dispositivo que transmite (7 bits) y del bit de la dirección de los datos. En este caso, el bit de la dirección de los datos (R_W) es un 1 lógico. Así, el primer byte transmitido es una dirección del esclavo de 7-bit seguida por un '1' para indicar el bit recibido. Los datos seriales se reciben mediante SDA, mientras que los relojes seriales son salidas SCL. Los datos seriales reciben 8 bits a la vez. Después de que se reciba cada byte,

se transmite un bit del ACK. Las condiciones del inicio y de término indican el principio y el fin de la transmisión.

Transmisión Principal de I²C

La transmisión de un byte de datos, de una dirección de 7-bit, o de la segunda mitad de una dirección de 10-bit se logra simplemente por escribir un valor al registro de I2CTRN. El usuario debe escribir solamente a I2CTRN cuando el módulo está en un estado de ESPERA.

Esta acción fijará la bandera completa del buffer (TBF) y permitirá que el generador de velocidad comience a contar e inicie la transmisión siguiente. La bandera de estado de transmisión, TRSTAT (I2CSTAT<14>), indica que un maestro que transmite está en progreso.

Recepción Principal de I²C

La recepción del modo principal es habilitada por la programación al recibir el bit habilitado (RCEN) (I2CCON<11>). El módulo I²C debe estar inactivo antes de que se fije el bit de RCEN, de lo contrario el bit RCEN será desatendido. La velocidad del generador comienza el conteo, y en cada vuelta que da el estado del pin de SCL los datos se cambian de lugar dentro de I2CRSR.

Operación del Módulo I²C durante los Modos de Sleep e Idle de la CPU

Operación I²C durante el modo de sleep de la CPU

Cuando el dispositivo entra en modo sleep, todas las fuentes de reloj al módulo son cerradas y permanecen en '0' lógico. Si el descanso ocurre en el centro de una transmisión, y la máquina de estado está parcialmente en una transmisión detenida de relojes, entonces se abandona la transmisión. Similarmente, si el descanso ocurre en el centro de una recepción, entonces la recepción es también abandonada.

Operación de I²C durante el modo idle de la CPU

Para el I²C, el bit I2CSIDL selecciona si el módulo se detiene en modo idle o continúa en este modo. Si I2CSIDL = 0, el módulo continuará la operación del modo idle. Si I2CSIDL = 1, el módulo se detendrá en modo inactivo.

2.2.6. INTERRUPCIONES

El dsPIC30F, tiene 24 fuentes de interrupción y 4 (traps) del procesador, que deben ser cuidadas de acuerdo con un esquema de prioridad.

La CPU, es responsable de leer la tabla del vector de la interrupción (IVT) y de transferir la dirección contenida en el vector de la interrupción al contador de programa. El vector de la interrupción es transferido del bus de datos al contador del programa, por medio de un multiplexor amplio de 24 bits en la entrada del contador de programa.

La tabla del vector de la interrupción (IVT) y la tabla alterna del vector de la interrupción (AIVT) son puestas cerca del principio de la memoria del programa (0x000004). Los IVT y los AIVT, se muestran en la tabla 2.1.

El regulador de interrupción es responsable de procesar las interrupciones y las excepciones del procesador, antes de ser presentadas a la base del procesador. El periférico interrumpe y se permiten las trampas, cuidando y controlando, usando registros de función centralizada:

- IFS0<15:0>, IFS1<15:0>, IFS2<15:0>

Todas las banderas de la petición de la interrupción se mantienen en éstos tres registros. Las banderas son fijadas por sus respectivos periféricos o las señales externas, y se definen por medio de software.

- IEC0<15:0>, IEC1<15:0>, IEC2<15:0>

Toda la interrupción habilita bits de control que son mantenidos en estos tres registros. Estos bits de control se utilizan para permitir individualmente interrupciones de los periféricos o de las señales externas.

- IPC0<15:0>... IPC11<7:0>

El usuario asigna el nivel de prioridad asociado a cada una de éstas interrupciones centralmente en éstos doce registros.

- IPL<3:0> el nivel actual de la prioridad de la CPU se almacena explícitamente en los bits del IPL. IPL3, está presente en el registro de CORCON, mientras que IPL<2:0> están presentes en el registro del estado (SR) en la base del procesador.
- INTCON1<15:0>, INTCON2<15:0>

Las funciones globales del control de la interrupción se derivan de estos dos registros. INTCON1, contiene el control y bandera de estado para las excepciones del procesador. El registro INTCON2, controla el comportamiento de la interrupción externa solicitada y el uso de la tabla alterna del vector.

Nota: Se consigue fijar los bits de bandera de interrupción cuando ocurre una condición de interrupción, sin importar el estado de su bit habilitado correspondiente. El software del usuario debe asegurar que los bits de bandera apropiados de interrupción estén bien antes de permitir una interrupción.

Todas las fuentes de interrupción pueden usarse asignando a uno de los 7 niveles de la prioridad, 1 a 7, por medio de los registros IPCx. Cada fuente de la interrupción se asocia a un vector de la interrupción, como se muestra en la tabla 2.1. Los niveles 7 y 1 representan las prioridades más altas y más bajas, respectivamente.

Nota: Asignar un nivel de prioridad de 0 a una fuente de la interrupción es equivalente a deshabilitar esa interrupción.

Si se fija el bit de NSTDIS (INTCON1<15>), la jerarquización de interrupciones se previene. Así, si una interrupción está actualmente siendo utilizada, la programación de una nueva interrupción se previene, incluso si la nueva interrupción es de una prioridad más alta que la única que es mantenida actualmente.

Nota: Los bits del IPL llegan a ser sólo de lectura siempre que el bit de NSTDIS se haya fijado a '1'.

La instrucción DISI, puede ser usada para inhabilitar el proceso de interrupciones de las prioridades 6 y más bajas para un cierto número de instrucciones, durante las cuales los restos del bit de DISI (INTCON2<14>) son fijados.

Cuando una interrupción es utilizada, la PC se carga con la dirección almacenada en la localidad del vector en la memoria del programa que corresponda a la interrupción. Hay 63 diversos vectores dentro del IVT (tabla 2.1). Estos vectores están contenidos en la localidad 0x000004 con 0x0000FE de la memoria del programa (tabla 2.1). Estas localidades contienen las direcciones de 24-bit, y para preservar robustez en orden, una trampa de error de dirección ocurrirá si hay tentativa de la PC de traer cualquiera de estas palabras durante la ejecución normal. Esto previene la ejecución de datos al azar como un resultado accidental del decremento de una PC en espacio del vector, un accidente de mapeo de una dirección del espacio de los datos en espacio del vector, o el balanceo de la PC sobre los 0x000000 después de alcanzar el final del espacio de memoria de programa implementado.

La ejecución de una instrucción GOTO a este espacio del vector también generará una trampa de error de dirección.

Prioridad de la Interrupción

El usuario asigna los bits de prioridad de interrupción (IP<2:0>) y para cada fuente individual de la interrupción están situados en el LS de 3 bits de cada parte, dentro del registro(s) de IPCx. Los 3 bits de cada parte son utilizados y son leídos como '0'. Estos bits definen el nivel de prioridad asignado a una interrupción particular del usuario.

Nota: El usuario selecciona el inicio del nivel de prioridad en 0, como la prioridad más baja, y nivel 7, como la prioridad más alta.

Puesto que más de una fuente de petición de la interrupción se puede asignar a un usuario específico de un nivel particular de prioridad, algunos medios son proporcionados para asignar prioridad dentro de un nivel dado. Este método se llama "prioridad natural de la orden" y es definitivo.

La prioridad natural de la orden es determinada por la posición de una interrupción en la tabla del vector, y afecta solamente la operación de la interrupción cuando las interrupciones múltiples con la misma prioridad que el usuario asigna llegan a estar pendientes al mismo tiempo.

La tabla 2.1 enumera las interrupciones y las fuentes de la interrupción para los dispositivos del dsPIC y sus números de vector asociados.

Nota 1: El esquema de la prioridad natural de la orden tiene 0 como la prioridad más alta y 53 como la prioridad más baja.

2: El número de prioridad natural de la orden es el mismo que el número INT.

La capacidad del usuario para asignar cada interrupción a uno de siete niveles de prioridad implica que el usuario puede asignar un nivel total muy alto de prioridad a una interrupción con una prioridad natural baja de la orden. Por ejemplo, para el PLVD (Detector de Voltaje Bajo) puede ser dada una prioridad de 7. El INT0 (interrupción

externa 0) puede ser asignado al nivel de prioridad 1, así que le da una prioridad eficaz muy baja.

INT Number	Vector Number	Interrupt Source
Highest Natural Order Priority		
0	8	INT0 - External Interrupt 0
1	9	IC1 - Input Capture 1
2	10	OC1 - Output Compare 1
3	11	T1 - Timer 1
4	12	IC2 - Input Capture 2
5	13	OC2 - Output Compare 2
6	14	T2 - Timer 2
7	15	T3 - Timer 3
8	16	SPI1
9	17	U1RX - UART1 Receiver
10	18	U1TX - UART1 Transmitter
11	19	ADC - ADC Convert Done
12	20	NVM - NVM Write Complete
13	21	SI2C - I ² C Slave Interrupt
14	22	MI2C - I ² C Master Interrupt
15	23	Input Change Interrupt
16	24	INT1 - External Interrupt 1
17	25	IC7 - Input Capture 7
18	26	IC8 - Input Capture 8
19	27	OC3 - Output Compare 3
20	28	OC4 - Output Compare 4
21	29	T4 - Timer 4
22	30	T5 - Timer 5
23	31	INT2 - External Interrupt 2
24	32	U2RX - UART2 Receiver
25	33	U2TX - UART2 Transmitter
26	34	Reserved
27	35	C1 - Combined IRQ for CAN1
28-40	36-48	Reserved
41	49	DCI - CODEC Transfer Done
42	50	LVD - Low Voltage Detect
43-53	51-61	Reserved
Lowest Natural Order Priority		

Tabla 2.1. Tabla del vector de interrupción dsPIC30F4013

Secuencia de la Interrupción

Todos los indicadores de suceso de la interrupción son muestreados al principio de cada ciclo de la instrucción por los registros de IFSx. Una petición pendiente de la interrupción (IRQ) es indicada por el bit de bandera que es igual a un '1' en un registro de IFSx. El IRQ causará una interrupción si el bit de la correspondencia en la interrupción permite el registro (IECx) y se determina. Para el resto del ciclo de la instrucción, las prioridades de todas las peticiones de la interrupción pendiente son evaluadas.

Si hay IRQ pendientes con un nivel de prioridad mayor que el nivel actual de la prioridad del procesador en los bits del IPL, el procesador será interrumpido.

El byte bajo del registro de estado contiene el nivel de prioridad del procesador en ese momento, antes de que inicie el ciclo de la interrupción. El procesador entonces carga el nivel de la prioridad para esta interrupción en el registro del estado. Esta acción inhabilitará todas las interrupciones de prioridad más bajas hasta la terminación de la rutina del servicio de la interrupción.

Nota 1: El usuario puede bajar siempre el nivel de la prioridad escribiendo un nuevo valor en el SR. La rutina del servicio de la interrupción debe definir los bits de bandera de interrupción en el registro de IFSx antes de bajar la prioridad de la interrupción del procesador, en orden para evitar interrupciones recurrentes.

2: El bit IPL3 (CORCON<3>) está siempre definido cuando se están procesando las interrupciones. Son fijadas solamente durante la ejecución de trampas.

La instrucción RETFIE (Regreso de la interrupción) no juntará al contador de programa y registros de estado para regresar el procesador a su prioridad de estado para la secuencia de la interrupción.

2.2.7. TRAPS (TRAMPAS)

Las trampas pueden ser consideradas como interrupciones que indican un error de software o de hardware y que adhieren a una prioridad predefinida según lo mostrado en la tabla 2.1. Están atentas para proporcionar al usuario los medios para corregir la operación errónea y funcionar dentro de la aplicación.

2.3. SET DE INSTRUCCIONES

2.3.1 MODOS DE DIRECCIONAMIENTO

Los modos de direccionamiento en la tabla 2.2 forman la base de los modos de direccionamiento optimizados para apoyar las características específicas de las instrucciones individuales.

Addressing Mode	Description
File Register Direct	The address of the file register is specified explicitly.
Register Direct	The contents of a register are accessed directly.
Register Indirect	The contents of Wn forms the EA.
Register Indirect Post-modified	The contents of Wn forms the EA. Wn is post-modified (incremented or decremented) by a constant value.
Register Indirect Pre-modified	Wn is pre-modified (incremented or decremented) by a signed constant value to form the EA.
Register Indirect with Register Offset	The sum of Wn and Wb forms the EA.
Register Indirect with Literal Offset	The sum of Wn and a literal forms the EA.

Tabla 2.2. Apoyos Fundamentales de los Modos de Direccionamiento

Instrucciones MCU

Las instrucciones MCU, de tres operandos son de la forma: Operando 3= Operando 1<función>, Operando 2 donde está siempre un registro, el Operando 1 del funcionamiento (es decir, el modo de direccionamiento puede solamente ser registro directo), que es designado Wb. EL operando 2 puede ser un registro de W, traído de la memoria de los datos o literalmente de 5 bits. La localidad de resultado puede ser un registro de W o una localidad de la dirección. Los modos de dirección siguientes son apoyados por las instrucciones de MCU:

- Registro Directo
- Registro Indirecto
- Registro Indirecto Post-Modificado
- Registro Indirecto Pre-Modificado
- Literalmente de 5 bits o de 10 bits

Nota: No todas las instrucciones apoyan a todos los modos de direccionamiento de arriba. Las instrucciones individuales pueden apoyar diversos subconjuntos de estos modos de direccionamiento.

Instrucciones de Movimiento y de Acumulador

Las instrucciones de movimiento y la clase del acumulador de instrucciones del DSP proporcionan un mayor grado de flexibilidad de direccionamiento que otras instrucciones. Además de los modos de direccionamiento apoyados por la mayoría de las instrucciones de MCU, las instrucciones de movimiento y del acumulador también apoyan el registro indirecto con el modo de direccionamiento del registro offset, también designado modo índice del registro.

Nota: Para las instrucciones, el modo de direccionamiento especificado en la instrucción puede diferenciar a la fuente y el destino EA. Sin embargo, el campo de 4 bits Wb (registro offset) se comparte entre la fuente y el destino (pero utilizado típicamente solamente por una).

En resumen, los modos de direccionamiento siguientes son apoyados por instrucciones de movimiento y del acumulador:

- Registro Directo
- Registro Indirecto
- Registro Indirecto Post-Modificado
- Registro Indirecto Pre-Modificado
- Registro Indirecto con el Registro Offset (Índice)
- Registro Indirecto con Offset literalmente
- Literalmente de 8-bit
- Literalmente de 16-bit

Nota: No todas las instrucciones apoyan a todos los modos de direccionamiento dados arriba. Las instrucciones individuales pueden apoyar diversos subconjuntos de estos modos de direccionamiento.

Instrucciones MAC

Las instrucciones duales del operando DSP de la fuente (CLR, ED, EDAC, MAC, MPY, MPY.N, MOVSAC y MSC), también designado como instrucciones MAC, utilizan un sistema simplificado de modos de direccionamiento para permitir que el usuario manipule con eficacia los indicadores de datos a través de las tablas indirectas del registro.

Los dos registros del pre-fetch o de la prebúsqueda del operando de la fuente deben ser un miembro del sistema {W8, W9, W10, W11}. Para los datos de lectura, W8 y W9 serán dirigidos siempre al X RAGU, y W10 y W11 serán dirigidos siempre a Y AGU. Los direccionamientos efectivos generados (antes y después modificados) deben, por lo tanto, ser direcciones válidas dentro del espacio de los datos de X para espacio de los datos de W8 y W9 e Y para W10 y W11.

Nota: El registro indirecto con la dirección del registro offset está solamente disponible para W9 (en espacio de X) y W11 (en espacio de Y).

En resumen, los modos de direccionamiento siguientes son apoyados por la clase MAC de instrucciones:

- Registro Indirecto
- Registro Indirecto Post-Modificado por 2
- Registro Indirecto Post-Modificado por 4
- Registro Indirecto Post-Modificado por 6
- Registro Indirecto con Registro Offset (Índice)

Otras Instrucciones

Además de los varios modos de dirección dados arriba, algunas instrucciones de varios tamaños de uso literal, por ejemplo, instrucciones BRA (rama) usan 16-bit de literales señaladas para especificar el destino de rama directamente, mientras que la instrucción DISI, utiliza un campo sin firmar de 14-bit. Ciertas operaciones, tales como NOP, no tienen ningún tipo de operando.

APÉNDICE 2

CAPÍTULO IV DISEÑO DEL SOFTWARE

4.6. LISTADO DE PROGRAMA

```
////////////////////////////////////
// THEREMIN DIGITAL BASADO EN LA EMISION Y RECEPCION DE UNA SEÑAL //
// DE SONAR DE 40 KHZ CON EL APOYO DE UN dspIC30F4013 PARA LA //
// DETECCION DEL INTERVALO DE TIEMPO TRANSCURRIDO ENTRE LA EMISION //
// Y LA DETECCION DEL REBOTE DE LA SEÑAL ULTRASONICA. //
// EL INTERVALO DE TIEMPO ES PROCESADO Y MODULA EN FRECUENCIA A //
// SEÑAL SENOIDAL QUE ES GENERADA POR EL "DSP", SIENDO ESTA SALIDA //
// APLICADA A UN CONVERTIDOR ANALOGICO-DIGITAL Y DESPUES A UN //
// AMPLIFICADOR DE AUDIO. //
////////////////////////////////////

#include <p30f4013.h> // Incluye la librería del dsPIC30F4013
_FOSC(CSW_FSCM_OFF & XT_PLL16); // Corre este programa usando oscilador
// externo de 7.3728 MHZ, el cual será
// multiplicado por 16, obteniéndose un
// reloj interno de 117.9648 MHZ.
// Además, la frecuencia

real de // operación de la CPU del dsPIC30 es
// (reloj_interno/4)= 29.4912 MHZ
// que equivale a 29.4912 MIPS.

_FWDT(WDT_OFF); // Watchdog timer disabled.
_FBORPOR(PBOR_OFF & MCLR_EN); // Brown-out reset disabled,
// MCLR reset enabled.
_FGS(CODE_PROT_OFF); // Code protect disabled.

// DECLARA VARIABLES GLOBALES DEL PROGRAMA
unsigned int Dato; // Contiene el dato obtenido de la Ta-
// bla de
// funciones Seno.
unsigned int i; // Apuntador de la Tabla de funciones
// Seno.
unsigned int Flanco1; // Es el marco de referencia usado para
// el
// cálculo de la distancia del objeto
// detectado por el Transductor
// Ultrasónico.
unsigned int Flanco2; // Aqui se deposita la cuenta del TIMER
// 2 que
// corresponde al tiempo en que fue
// detectado
// un objeto por el Transductor
// Ultrasónico.
unsigned int CTE; // Es la constante de tiempo neta
// resultante
```

```

// a la detección de un objeto por el
// Transductor Ultrasónico.
unsigned int Contador; // Contador del detector de veces que
// es
// sensada la presencia de un objeto
// por el
// Transductor Ultrasónico.
// Tiene la finalidad de evitar estar
// escribiendo
// en cada ciclo el valor actual de la
// distancia
// a la cual fue detectado el objeto,
// ya que esto
// provocaría inestabilidad en la señal
// senoidal
// que se está generando

// TABLA SENOIDAL PARA UN PERIODO COMPLETO, CON 90 PUNTOS TOMADOS
// A 16 BITS DE RESOLUCION.
// Nota: Se están considerando la naturaleza bipolar de la señal.
int sinetable[] __attribute__((far,section(".const,r")))=
{0x0000,0x08ed,0x11d0,0x1a9c,0x2347,0x2bc6,0x340f,0x3c17,0x43d3,
0x4b3b,0x5246,0x58e9,0x5f1e,0x64dc,0x6ald,0x6ed9,0x730a,0x7691,
0x79bb,0x7c31,0x7e0d,0x7f4b,0x7feb,0x7feb,0x7f4b,0x7e0d,0x7c31,
0x79bb,0x7691,0x730a,0x6ed9,0x6ald,0x64dc,0x5f1e,0x58e9,0x5246,
0x4b3b,0x43d3,0x3c17,0x340f,0x2bc6,0x2347,0x1a9c,0x11d0,0x08ed,
0x0000,0xf712,0xee2f,0xe563,0xdc8,0xd439,0xcbf0,0xc3e8,0xbc2c,
0xb4c4,0xad9,0xa716,0xa0e1,0x9b23,0x95e2,0x9126,0x8cf5,0x896e,
0x8644,0x83ce,0x81f2,0x80b4,0x8014,0x8014,0x80b4,0x81f2,0x83ce,
0x8644,0x896e,0x8cf5,0x9126,0x95e2,0x9b23,0xa0e1,0xa716,0xad9,
0xb4c4,0xbc2c,0xc3e8,0xcbf0,0xd439,0xdc8,0xe563,0xee2f,0xf712};

// INICIALIZACION DEL MODULO IC1 DE CAPTURA DE ENTRADA.
void IC1_INI()
{
    IC1CON=0x0000; // Apaga el Módulo de Captura IC1.
    IPC0bits.IC1IP=0x1; // Asigna Prioridad 1 a la interrupción
// del Módulo de Captura IC1.
    IEC0bits.IC1IE=0; // Deshabilita interrupciones de IC1.
    IFS0bits.IC1IF=0; //Limpia la bandera de interrupción
// de IC1.
    T2CON=0x0000; // Apaga Timer2.
    TMR2=0; // Limpia contador del TIMER 2.
    IEC0bits.T2IE=0; // Deshabilita interrupciones del Timer
// 2.
    T2CONbits.TCS=0; // La fuente de reloj del Timer 2 es el
// reloj interno TCY (Clock/4).
    T2CONbits.T32=0; // Timer 2 como contador de 16 bits.
    T2CONbits.TCKPS=0x0; // Pre escalamiento 1:1
    T2CONbits.TGATE=0; // Disparo de Tiempo de Acumulación
// deshabilitado.
    T2CONbits.TSIDL=0; // Continúa operando el Timer2 aun en
// la condición IDLE.
    T2CONbits.TON=1; // Inicia operación del Timer 2.
    IEC0bits.IC1IE=1; // Habilita interrupciones de IC1.
    IC1CONbits.ICSIDL=0; // El Módulo de Captura IC1 continúa

```

```

    IC1CONbits.ICTMR=1;           // trabajando aun en el modo IDLE.
    IC1CONbits.ICI=0x0;          // El Módulo de Captura de Entrada IC1
    IC1CONbits.ICM=0x3;          // trabajará con el Timer 2.
    // El Módulo de Captura de Entrada IC1
    // siempre interrumpirá cuando ocurra
    // un
    // flanco de disparo en la entrada IC1.
    // El Modo de Captura será siempre con
    // flanco de subida de la entrada, sin
    // ningún pre escalamiento.
}

```

```

// INICIALIZACION DEL TIMER 4 PARA MANEJO DE INTERRUPCIONES Y AIGNACION
// DE CONSTANTE DE TIEMPO A COMPARAR.

```

```

void Ini_Timer4()
{
    T4CON=0x0000;                // Apaga el TIMER 4.
    TMR4=0;                      // Limpia contador del TIMER 4.
    IEC1bits.T4IE=0;            // Deshabilita interrupciones del
    // Timer 4.
    IPC5bits.T4IP=0x2;          // El nivel de prioridad de
    // interrupción
    // del TIMER 4 será de 2.
    T4CONbits.TCS=0;            // La fuente de reloj del TIMER 4 es
    // el
    // reloj interno TCY (Clock/4).
    T4CONbits.T32=0;            // TIMER 4 como contador de 16 bits.
    T4CONbits.TCKPS=0x0;        // Pre escalamiento 1:1
    T4CONbits.TGATE=0;          // Disparo de Tiempo de Acumulación
    // deshabilitado.
    T4CONbits.TSIDL=0;          // Continua operando el TIMER 2 aun
    // en
    // la condición IDLE.
    PR4=0x0080;                // Carga constante inicial de
    // comparación
    // del TIMER 4.
}

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
// Nota: Para la simulación se está
// asignando
// un número muy pequeño, dando así
// un tiempo
// razonable para observar el
// comportamiento.
IFS1bits.T4IF=0;              // Limpia bandera de interrupción
// del TIMER 4.
IEC1bits.T4IE=1;              // Habilita interrupciones
// del TIMER 4.
T4CONbits.TON=1;              // Inicia operación del Timer 4.
}

```

```

// MANEJO DE LA INTERRUPCION DE CAPTURA DE ENTRADA IC1.
// Rutina de servicio de interrupción.
void __attribute__((__interrupt__)) _IC1Interrupt(void)

```

```

{
    PORTDbits.RD9=0; // Apaga el Transductor Ultrasónico.
    Flanco2=TMR2; // Lee cual es el valor de cuenta
                  // actual
                  // del TIMER 2 que corresponde a la
                  // detección del pulso reflejado
                  // detectado
                  // por el Transductor Ultrasónico.
    if(Flanco1<Flanco2) // Compara si el Flanco1 es menor que
                       // Flanco 2.
    {
        CTE=Flanco2-Flanco1; // Obtén la nueva constante de tiempo
                             // que será
                             // asignada al TIMER 4.
    }
    if(Flanco1>Flanco2) // Compara si el Flanco1 es mayor que
                       // Flanco 2.
    {
        CTE=Flanco1-Flanco2; // Obtén la nueva constante de tiempo
                             // que será
                             // asignada al TIMER 4.
    }
    if(Contador==90)
    {
        TMR4=0; // Pon en reset al contador del
                // Timer 4.
        PR4=CTE; // Asigna al TIMER 4 la nueva
                // constante de
                // tiempo calculada.
        Contador=0; // Limpia Contador del número de
                  // detecciones.
    }
    Contador=Contador+1; // Incrementa el Contador del número
                        // de
                        // detecciones.
    TMR2=0; // Reset al contador del TIMER 2.
    PORTDbits.RD9=1; // Enciende el Transductor
                    // Ultrasónico.
    Flanco1=TMR2; // Lee el estado actual del TIMER 2
                 // para
                 // establecer el marco de referencia
                 // entre
                 // el pulso transmitido por el
                 // Transductor
                 // Ultrasónico y el pulso que será
                 // reflejado.
    IFS0bits.IC1IF=0; // Pon la bandera de IC1 en cero
                     // (limpia
                     // interrupción).
}

```

```

// MANEJO DE LA INTERRUPCION DEL TIMER 4.
// Rutina de servicio de interrupción.
void __attribute__((__interrupt__)) _T4Interrupt(void)
{
    if(i==90)

```

```

    {
        i=0;                // Si el apuntador de la Tabla
                           // Senoidal es 90
                           // pon entonces apuntador en cero.
    }
    Dato=sinetable[i];    // Lee dato actual apuntado por "i"
                           // de la

                           // Tabla Senoidal.
    PORTB=(Dato>>3);     // Como el Puerto B es de 13 bits y
                           // la Tabla
                           // Senoidal es de 16 bits, realiza un
                           // corrimiento
                           // de 3 bits para ajustar dato al
                           // tamaño del
                           // Puerto B.
    i=i+1;                // Incrementa el apuntador para el
                           // próximo evento.
    IFS1bits.T4IF=0;     // Pon la bandera del TIMER 4 en cero
                           // (limpia
                           // interrupción).
}

```

```

// ***** //
// **      RUTINA PRINCIPAL DEL THEREMIN DIGITAL.      ** //
// ***** //

```

```

main()
{
    i=0;                // Pon en ceros al apuntador de la tabla
                           // de coeficientes "Seno".
    Contador=0;        // Pon el Contador de veces de detección
                           // en cero.
    ADPCFG=0XFFFF;    // Asegúrate que los pins del Puerto B
                           // relacionados con el convertidor A/D
                           // están en modo digital.
    TRISB=0x0000;     // Configura el Puerto B como salida.
    TRISDbits.TRISD8=1; // Configura el bit RD8 como entrada.
                           // Nota: este bit está destinado para la
                           // recepción del pulso del Transductor
                           // Ultrasónico.
    TRISDbits.TRISD9=0; // El bit RD9 del Puerto D es
                           // configurado
                           // como salida.
                           // Nota: El bit RD9 controla el
                           // encendido
                           // del módulo que controla al
                           // Transductor
                           // Ultrasónico de 40 KHZ.
    PORTDbits.RD9=0;   // Apaga el Transductor Ultrasónico.
    IC1_INI();        // Rutina de inicialización del Módulo
                           // Captura de Entrada IC1.
    Ini_Timer4();     // Rutina de inicialización del TIMER 4.
    PORTDbits.RD9=1;   // Enciende el Transductor Ultrasónico.
    Flancol=TMR2;     // Lee el estado actual del TIMER 2 para
                           // establecer el marco de referencia
}

```

```

// entre
// el pulso transmitido por el
// Transductor
// Ultrasónico y el pulso que será
// reflejado.
for(;;)
{
//     IFS0bits.IC1IF=0; // Pon la bandera de IC1 en cero (limpia
//     IFS0bits.IC1IF=1; // interrupción).
//     IFS0bits.IC1IF=0;
}
}
```

REFERENCIAS

- [1] http://es.wikipedia.org/wiki/M%C3%BAsica_docta
- [2] <http://presencias.net/indpdm.html?http://presencias.net/histor/ht2013b.html>
- [3] http://www.xtec.es/centres/a8019411/caixa/musica_es.htm
- [4] NOVARO Augusto, Teoría de la Música. Sistema Natural. Base del Natural Aproximado. México, D. F., 1927
- [5] http://es.wikipedia.org/wiki/Instrumento_musical_electr%C3%B3nico
- [6] Microsoft ® Encarta ® 2006. © 1993-2005 Microsoft Corporation. Música Electrónica. Sintetizador. Robert Moog.
- [7] http://es.wikipedia.org/wiki/Ondas_Martenot
- [8] http://es.wikipedia.org/wiki/Moog_modular
- [9] <http://es.wikipedia.org/wiki/Minimoog>
- [10] <http://www.musicaytecnologia.com/Articulos.asp?Accion=verArticulo&IDArticulo=49&PagComentario=1&PagArticulo=3>
- [11] <http://www.geocities.com/SunsetStrip/Stadium/1675/htm/sintetizador.html>
- [12] <http://es.wikipedia.org/wiki/Therem%C3%ADn>
- [13] <http://www.stickcenter.com/Magazine/Mayo05.pdf>
- [14] <http://es.wikipedia.org/wiki/RISC>
- [15] http://es.wikipedia.org/wiki/Arquitectura_Harvard
- [16] <http://www.terra.es/personal/fremiro/arquitectura.htm>
- [17] http://es.wikipedia.org/wiki/Procesador_digital_de_se%C3%B1al
- [18] <http://www.redeya.com/electronica/tutoriales/dsp1/dsp1.htm#AB>
- [19] Apuntes de Microchip, dsPIC30F3014, dsPIC30F4013. Data Sheet. High-Performance. Digital Signal Controllers

[20] www.agelectronica.com.mx

[21] DELAHAY Francisco. Artículo: Los Sintetizadores: a la conquista del espacio de los sonidos