

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE INGENIERÍA**

### **DETERMINACIÓN DEL TORQUE DE UNA MÁQUINA TRIFÁSICA DE INDUCCIÓN UTILIZANDO PROCESAMIENTO DIGITAL DE SEÑALES**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y CONTROL**

**ANDRÉS HUMBERTO LUZURIAGA ZHINDÓN**

**ELENA SALOMÉ UQUILLAS ANDRADE**

**DIRECTOR: PABLO RIVERA, MSc.**

**Quito, noviembre 2001**

## DECLARACIÓN

Nosotros, Andrés Humberto Luzuriaga Zhindón y Salomé Uquillas Andrade, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado, o clasificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



---

**Andrés H. Luzuriaga Z.**



---

**E. Salomé Uquillas A.**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Andrés Humberto Luzuriaga Zhindón y Elena Salomé Uquillas Andrade, bajo mi supervisión.

A handwritten signature in black ink, appearing to read 'Pablo Rivera', is positioned above a horizontal line.

MSc. Pablo Rivera

**DIRECTOR DEL PROYECTO**

## **AGRADECIMIENTO**

Mi eterna gratitud a Martha y Humberto, mis padres, quienes han sabido “soportar” estos años de estudio y a mis tías Bertha, Rosa y Laura por el desinteresado apoyo que me han brindado.

A mis profesores, especialmente al Ing. Pablo Rivera por la confianza y guía paciente.

A mi hermano Paúl, por su aliento constante.

A Boris, Jenny, Santiago, Salomé, Piedad, Christian, Víctor, Julio y Luis, que con su amistad han hecho estos años muy enriquecedores.

**Andrés H. Luzuriaga Z.**

## **AGRADECIMIENTO**

A Dios.

A mis padres, hermanos y a Pablo por su amor y comprensión.

A mis profesores, en especial al Ing. Pablo Rivera por su amistad y guía.

Al Ing. Eduardo Freire por su paciencia y apoyo.

A mis amigos, en especial Fernando Carrera, Andrés Luzuriaga y Andrés Pulla por su amistad incondicional.

**E. Salomé Uquillas A.**

## DEDICATORIA

*A mi Abuelita Leticia, su amor y  
cariño siempre será un ejemplo e impulso en mi vida.*

**Andrés H. Luzuriaga Z.**

## **DEDICATORIA**

*A mis padres, Roberto y Helena, y a mi hermano David  
quienes con su amor y dulzura ha llenado mi vida de alegría.*

**E. Salomé Uquillas A.**

## RESUMEN

En el presente proyecto se obtiene el torque de la máquina trifásica de inducción mediante la realización de un algoritmo a partir del modelo matemático de la misma y se visualiza gráficamente los resultados para distintos pasos de carga con el objetivo de realizar una comparación con resultados obtenidos analógicamente. Tomando en cuenta que la industria de Control de Motores es un sector fuertemente competitivo, para mantener la competitividad debe reducir costos, pero también tiene que responder a una reducción del consumo de energía. Así, los resultados de estos factores constituyen la necesidad de implementar algoritmos mejorados. La tecnología DSP permite obtener ambos, un alto nivel de rendimiento así como también aprovechamiento inteligente para reducir los costos completos del sistema.

A continuación se presenta el resumen del contenido de cada capítulo:

En el capítulo I se presenta una visión general de la máquina de inducción trifásica. Además, se justifica la utilización de los accionamientos de velocidad variable, se mencionan las consideraciones que se deben tomar en cuenta para el control de la velocidad, y se describen brevemente los tipos de control escalar y por vector de campo.

En el capítulo II se elabora un estudio matemático de la máquina de inducción en variables  $abc$ . A partir de éste se reduce la complejidad de este modelo mediante un cambio de variables a variables  $qd0$  en un marco de referencia arbitrario. Finalmente, se obtienen las ecuaciones de la máquina en variables  $qd0$  en un marco de referencia estacionario, que será el marco de referencia a utilizar en el desarrollo del algoritmo para la obtención del torque.

En el capítulo III se presenta la definición de procesamiento digital de señales, se describen los conceptos básicos, y los elementos que intervienen en un sistema de procesamiento digital; así como, las ventajas que tiene frente al procesamiento

analógico de señales. A continuación, se realiza una descripción general del módulo de evaluación, del procesador digital de señales, y el circuito de interfase análogo-digital incluidos en este módulo; definiendo sus características principales y presentando los diagramas de bloques correspondientes a cada uno de ellos.

En el capítulo IV se describe el software y lenguajes de programación utilizados. Luego se plantea el acondicionamiento y desacoplamiento de las señales de voltaje y velocidad, la conversión análoga - digital y la estructura del programa describiendo el algoritmo desarrollado para el procesamiento de los datos adquiridos. Por último, se presenta el desarrollo de la interfaz para la interpretación y visualización gráfica del resultado.

En el capítulo V se presentan los resultados en tiempo real de la obtención del torque de la máquina de inducción trifásica mediante los algoritmos implementados. Se visualiza el torque para distintos pasos de carga haciendo una comparación con las mediciones del dispositivo analógico. Para finalizar, se extrae conclusiones de los resultados obtenidos del trabajo global.

## PRESENTACIÓN

Durante los últimos años el campo de los accionamientos eléctricos controlados ha experimentado una rápida expansión debido principalmente a las ventajas de los semiconductores culminando en microprocesadores microelectrónicos y DSPs (Procesadores Digitales de Señales). Estos adelantos tecnológicos han permitido el desarrollo de un control realmente efectivo de accionamientos de la máquina de alterna, utilizando hardware con menor disipación de potencia y estructuras de control más precisas.

Debido a dichos adelantos tecnológicos, en la actualidad los variadores de velocidad utilizan nuevas técnicas de control como: control por vector de campo, control con técnica difusa, entre otros. El esquema más eficiente de control por vector se denomina: Control por Campo Orientado. Éste está basado en tres puntos importantes: los vectores de corriente y voltaje de la máquina, la transformación de un sistema trifásico dependiente de la velocidad y el tiempo en un sistema de dos coordenadas invariante con el tiempo, y la generación de una modulación por ancho de pulso efectiva. Gracias a estos factores, el control de la máquina de AC adquiere las ventajas del control de la máquina de DC y se libera a sí misma de los inconvenientes de la conmutación mecánica. Adicionalmente, esta estructura de control en estado estable y transitorio muy preciso, conduce a un alto rendimiento dinámico en términos de respuesta de tiempo y conversión de energía.

El propósito del presente trabajo se basa en sentar las bases para una posterior aplicación en el control del motor trifásico de inducción por Campo Orientado, utilizando para ello el módulo de evaluación que incorpora un procesador digital de señales. Se adquieren las señales de voltaje de línea y velocidad de la máquina de inducción mediante un conversor A/D y se procesan en el DSP, en el cual se resuelve el modelo matemático de la máquina con el fin de obtener el torque para distintos pasos de carga y visualizarlo gráficamente

# CONTENIDO

## RESUMEN

## PRESENTACIÓN

<b>1 VISIÓN GENERAL DE LA MÁQUINA DE INDUCCIÓN</b>	2
1.1 DESCRIPCIÓN DE LA MÁQUINA DE INDUCCIÓN	2
1.2 CONTROL DE VELOCIDAD PARA MÁQUINAS DE INDUCCIÓN	5
1.2.1 EL PORQUÉ DE LOS ACCIONAMIENTOS DE VELOCIDAD VARIABLE	5
1.2.2 CONSIDERACIONES DEL CONTROL DE VELOCIDAD VARIABLE	7
1.2.3 TIPOS DE CONTROL	9
<b>2 MODELO DE LA MÁQUINA DE INDUCCIÓN TRIFÁSICA</b>	16
2.1 MODELO DE LA MÁQUINA DE INDUCCIÓN TRIFÁSICA EN VARIABLES $abc^{[2]}$	16
2.1.1 CAMPO MAGNÉTICO GIRATORIO Y DESLIZAMIENTO	16
2.1.2 ECUACIONES DE VOLTAJE EN VARIABLES $abc$	18
2.1.3 ECUACIONES DE CONCATENACIÓN DE FLUJO $abc$	20
2.2 MODELO DE LA MÁQUINA DE INDUCCIÓN UTILIZANDO UN MARCO DE REFERENCIA	24
2.2.1 MODELO DE LA MÁQUINA EN EL MARCO DE REFERENCIA ARBITRARIO $qd0^{[6]}$	24
2.2.2 ECUACIONES DE VOLTAJE $qd0$	28
2.2.3 ECUACIONES DE CONCATENACIÓN DE FLUJO $qd0$	29
2.2.4 ECUACIÓN DE TORQUE $qd0$	34
2.2.5 MARCO DE REFERENCIA ESTACIONARIO	36
<b>3 PROCESAMIENTO DIGITAL DE SEÑALES UTILIZANDO EL DSP TMS320C31</b>	46
3.1 PROCESAMIENTO DIGITAL DE SEÑALES	46
3.1.1 ELEMENTOS BÁSICOS DE UN SISTEMA DE PROCESADO DIGITAL DE SEÑALES	47
3.1.2 PROCESADOR DIGITAL DE SEÑALES	49
3.1.3 CONVERSIÓN ANÁLOGA - DIGITAL	50
3.1.4 VENTAJAS DEL PROCESAMIENTO DIGITAL DE SEÑALES FRENTE AL ANALÓGICO	50
3.2 MÓDULO DE EVALUACIÓN DSK STARTER KIT	52
3.2.1 VISIÓN GENERAL Y CARACTERÍSTICAS DEL DSK	52
3.2.2 VISIÓN FUNCIONAL DEL DSK	59
3.2.3 DESCRIPCIÓN GENERAL DEL DSP TMS320C31	68
3.2.4 ARQUITECTURA DEL TMS320C31	70
3.2.5 ORGANIZACIÓN DE MEMORIA DEL TMS320C31	75
3.2.6 SET DE INSTRUCCIONES	79
3.2.7 DESCRIPCIÓN DEL CIRCUITO DE INTERFAZ ANALÓGICO	87

## **4 DESARROLLO DEL ALGORITMO PARA LA OBTENCIÓN DEL TORQUE**

<b><u>4.1</u></b>	<b><u>DESCRIPCIÓN DEL SOFTWARE DEL DSK</u></b>	94
4.1.1	<u>DESCRIPCIÓN DEL ENSAMBLADOR DEL DSK</u>	94
4.1.2	<u>DESCRIPCIÓN DEL DEPURADOR DEL DSK</u>	99
<b><u>4.2</u></b>	<b><u>ACONDICIONAMIENTO DE SEÑALES ADQUIRIDAS</u></b>	104
4.2.1	<u>SEÑAL DE VOLTAJE</u>	105
4.2.2	<u>SEÑAL DE VELOCIDAD</u>	106
<b><u>4.3</u></b>	<b><u>CONVERSIÓN ANÁLOGA-DIGITAL DE LAS SEÑALES DE VOLTAJE Y VELOCIDAD</u></b>	108
4.3.1	<u>INICIALIZACIÓN DEL AIC TLC32040</u>	109
<b><u>4.4</u></b>	<b><u>DESCRIPCIÓN DEL ALGORITMO IMPLEMENTADO EN EL DSP TMS320C31</u></b>	116
4.4.1	<u>INICIALIZACIÓN</u>	116
4.4.2	<u>ESTABLECIMIENTO DE PARÁMETROS DEL AIC</u>	117
4.4.3	<u>ASIGNACIÓN DE VALORES A CONSTANTES Y PARÁMETROS DEL MOTOR</u>	119
4.4.4	<u>ESTABLECIMIENTO CONDICIONES INICIALES DEL MOTOR</u>	121
4.4.5	<u>INICIALIZACIÓN DEL TEMPORIZADOR DEL DSP</u>	121
4.4.6	<u>INICIALIZACIÓN Y PROGRAMACIÓN DEL AIC</u>	122
4.4.7	<u>ADQUISICIÓN DE DATOS MEDIANTE EL AIC</u>	125
4.4.8	<u>OBTENCIÓN DEL TORQUE UTILIZANDO DATOS DE VOLTAJE Y VELOCIDAD MUESTREADOS</u>	125
4.4.9	<u>TRANSMISIÓN DEL TORQUE MEDIANTE EL CONVERTOR D/A</u>	133
4.5	<u>INTERFAZ GRÁFICA PARA LA VISUALIZACIÓN DEL RESULTADO</u>	134
<b><u>5 PRUEBAS Y CONCLUSIONES</u></b>		136
<b><u>5.1</u></b>	<b><u>PRUEBAS EXPERIMENTALES</u></b>	136
5.1.1	<u>ARRANQUE SIN CARGA</u>	136
5.1.2	<u>AÑADIR PASOS DE CARGA</u>	137
5.1.3	<u>ARRANQUE CON CARGA</u>	139
5.1.4	<u>DISMINUCIÓN DE LA CARGA</u>	140
5.1.5	<u>AUMENTO DE LA CARGA</u>	142
5.1.6	<u>CAMBIANDO PASOS DE CARGA</u>	144
<b><u>5.2</u></b>	<b><u>CONCLUSIONES Y RECOMENDACIONES</u></b>	146
5.2.1	<u>CONCLUSIONES</u>	146
5.2.2	<u>RECOMENDACIONES</u>	148

## **BIBLIOGRAFÍA**

## **ANEXOS**

ANEXO A: DIMENSIONES DE LA TARJETA DSK Y DIAGRAMAS ESQUE- A-1  
MÁTICOS

ANEXO B: HOJAS DE DATOS DEL DSP TMS320C31

B-1

ANEXO C: SET DE INSTRUCCIONES DEL TMS320C31

C-1

ANEXO D: HOJAS DEL DATOS DEL AIC TLC32040

D-1

# **CAPÍTULO 1**

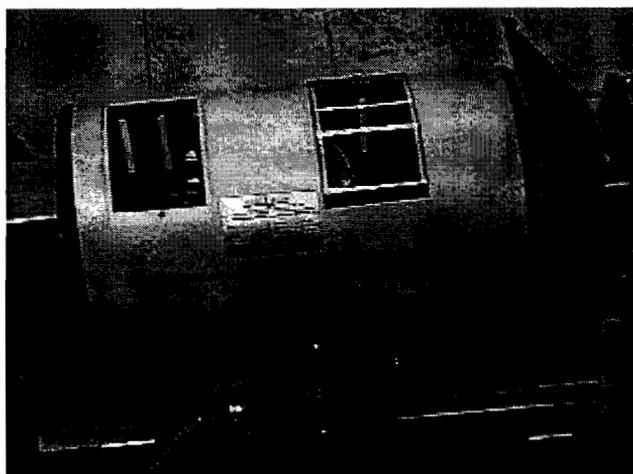
## **VISIÓN GENERAL DE LA MÁQUINA DE INDUCCIÓN**

# 1 VISIÓN GENERAL DE LA MÁQUINA DE INDUCCIÓN

## 1.1 DESCRIPCIÓN DE LA MÁQUINA DE INDUCCIÓN

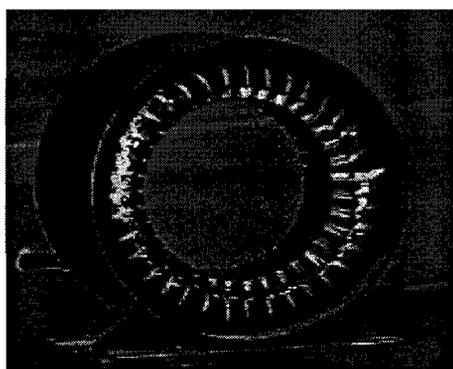
Las máquinas de inducción trifásicas son máquinas de velocidad asincrónica, que operan a velocidades bajo la velocidad sincrónica cuando funcionan como motor y sobre la velocidad sincrónica cuando funcionan como generador. Son comparativamente menos caras que las máquinas sincrónicas o de corriente continua equivalentes en tamaño, y varían en rangos de algunos vatios hasta cientos de HPs.

La máquina de inducción es usada en una amplia variedad de aplicaciones como un medio para convertir potencia eléctrica en trabajo mecánico. Ellas son, efectivamente, los caballos de trabajo de la industria en la actualidad. Como motores, son robustos y requieren muy poco mantenimiento. Sin embargo, sus velocidades no son tan fácilmente controlables como con los motores de DC. Sus corrientes de arranque son altas, típicamente alrededor de seis a ocho veces su valor a carga nominal, y operan con un pobre factor de potencia en retraso cuando son ligeramente cargadas.



**Fig. 1.1** Vista de un Motor de inducción

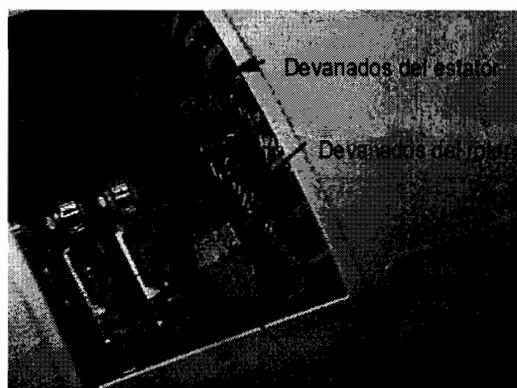
La mayoría de motores de inducción son de tipo giratorio, básicamente con un estator estacionario y un rotor giratorio. El estator tiene un núcleo magnético cilíndrico de anillo que está ubicado dentro de una armadura de metal sobre la cual están montadas las capas de láminas, los blindajes finales y una caja de terminales. En el centro de los blindajes se encuentran los cojinetes que constituyen el soporte del eje del rotor. El núcleo magnético del estator está formado por delgadas laminaciones eléctricas de acero, apiladas con ranuras espaciadas uniformemente acunadas en el interior de la circunferencia, para acomodar los tres devanados distribuidos del estator. Para máquinas de 60 Hz, la laminación es aproximadamente de 0.5 mm de ancho. Los devanados del estator están formados por bobinas acopladas, de conductores de cobre o aluminio. Las bobinas están aisladas de las paredes de la ranura. Los ejes de los devanados del estator de una máquina de P polos están espaciados  $(2/P)(2\pi/3)$  radianes mecánicos, con cada fase ocupando el mismo número de ranuras. Los terminales de los tres devanados de fase del estator pueden estar conectados en delta o en estrella.



**Fig. 1.2** Devanado del estator

El rotor consiste de un núcleo cilíndrico laminado de hierro con ranuras periféricas uniformemente espaciadas para acomodar los devanados del rotor. Existen dos tipos principales de devanados del rotor: bobinado o jaula de ardilla.

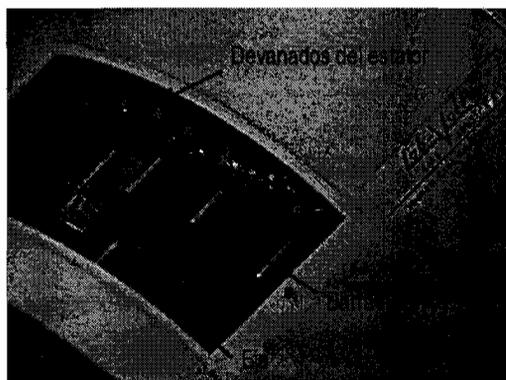
En una máquina de rotor bobinado, los devanados, semejantes a los devanados distribuidos del estator, son bobinados en un núcleo cilíndrico laminado con ranuras uniformemente espaciadas en la periferia exterior. Los terminales de los devanados del rotor pueden ser sacados vía anillos deslizantes y escobillas. Estos anillos deslizantes están aislados entre sí y del eje. Una cantidad variable de resistencia externa puede ser conectada a los devanados del rotor por medio de los anillos deslizantes para obtener un mayor torque de arranque, o un limitado control de la velocidad bajo la velocidad sincrónica. En accionamientos grandes y costosos, los terminales de los devanados del rotor son conectados a una fuente de frecuencia de deslizamiento para recuperar parte de la potencia que fluye hacia el circuito del rotor.



**Fig. 1.3** Devanados de motor de inducción con rotor bobinado

El motor de inducción menos costoso y más ampliamente difundido es el de jaula de ardilla. El rotor jaula de ardilla tiene barras axiales uniformemente espaciadas que son soldadas sobre los anillos comunes en ambos extremos. Después de que las laminaciones del núcleo del rotor son apiladas en un molde, el molde es llenado con aluminio derretido. Las barras del rotor, anillos y aspas del ventilador son fundidas en un único proceso económico. En máquinas grandes, las barras del rotor son de una aleación de cobre y son conducidas sobre ranuras uniformemente espaciadas en la periferia del rotor; las barras son aseguradas

sobre los anillos comunes en ambos extremos. No existe aislamiento entre las barras y paredes de las ranuras del rotor.



**Fig. 1.4** Devanados del motor de inducción jaula de ardilla

Mientras más pequeño es el entrehierro, mejor es la inducción electromagnética mutua entre los bobinados del estator y del rotor. Sin embargo, la extensión a la que el entrehierro puede ser reducido es fijada por las tolerancias y costos de fabricación, y por las pérdidas de núcleo permisibles.

## **1.2 CONTROL DE VELOCIDAD PARA MÁQUINAS DE INDUCCIÓN**

### **1.2.1 EL PORQUÉ DE LOS ACCIONAMIENTOS DE VELOCIDAD VARIABLE.**

Actualmente se utiliza una gran cantidad de mecanismos de producción que trabajan a velocidad variable, como por ejemplo, las máquinas – herramienta, trenes de laminado, grúas y aparatos de transporte, mecanismos de la industria papelera, textil y otras ramas industriales. En todos estos mecanismos, variando la velocidad se consigue una elevada productividad y una buena calidad de trabajo.

La variación de velocidad es el cambio forzado de la velocidad del accionamiento eléctrico para satisfacer las distintas exigencias del proceso industrial. Sin embargo, no debe confundirse con un cambio de velocidad cuando se varía las condiciones de la carga que esta acoplada al árbol de la máquina.

Cabe mencionar, que antes de que se desarrollen los variadores de velocidad eléctricos, la variación de velocidad se lograba por métodos mecánicos con la ayuda de juegos de engranajes, poleas, transmisiones de fricción regulables. En la actualidad, se van reemplazando estos mecanismos porque la regulación electrónica tiene una serie de ventajas tanto técnicas como económicas.

El objetivo de abaratar costos, transmitir las funciones de variación y control directamente al accionamiento eléctrico, y de mejorar la calidad de la producción, a llevado a la industria a desarrollar mejores accionamientos de velocidad. Cada vez la tecnología permite un mayor rango de variación y suavidad de los saltos de velocidad a las máquinas de inducción, con las ventajas conocidas de confiabilidad, factor de potencia, entre otras, sobre las de corriente continua.

En esta parte de la literatura se pretende dar un enfoque global de los diversos métodos para realizar la variación de velocidad de la máquina de corriente alterna y una mejor comprensión del método de variación por vector de campo. Debido a que en muchas aplicaciones industriales se requiere el control del torque a bajas velocidades, aprovechando los prototipos construidos por medio de inversores PWM se pretende dar las bases para que se desarrolle a futuro un control de velocidad aplicando la técnica de control por vector de campo. Para esto, es necesario que se pueda determinar el torque de la máquina, que es el propósito del presente trabajo; el método de vector de campo actualmente constituye uno de los métodos más modernos que se está desarrollando con una serie de ventajas sobre los otros que se mencionará más adelante.

## **1.2.2 CONSIDERACIONES DEL CONTROL DE VELOCIDAD VARIABLE**

El problema del control de velocidad tiene gran significado dentro de la economía de un sistema de producción industrial, ya que como metas se traza una abundante producción y una alta calidad, para solucionar este problema, cada vez se adelanta la técnica y los procedimientos industriales se vuelven más económicos y fiables.

Los principales índices que caracterizan los accionamientos de velocidad eléctricos son:

- Límites o campo de variación
- Suavidad
- Economía
- Estabilidad de trabajo a la velocidad preestablecida
- Carga admisible en distintas velocidades.

A continuación se menciona cada uno de estos tópicos para un conocimiento general de lo que se debe tomar en cuenta para que el accionamiento sea óptimo.

### **1.2.2.1 Los límites o campo de variación de velocidad**

Esto se refiere a determinar el campo o gama de la variación de velocidad para que los requisitos relacionados con el salto admisible de velocidad suplan la carga que está conectada al eje. Por ejemplo, a bajas velocidades se crea el problema de la elevación de la corriente en los elementos eléctricos. Hay que considerar en este caso la eficiencia y el factor de potencia de la máquina; sin embargo, en la máquina de inducción estos parámetros son fiables debido a la robustez que presenta.

### **1.2.2.2 Suavidad de la variación**

Se caracteriza por el número de velocidades estables obtenidas en la gama de variación. La suavidad es más elevada cuanto menor sea el salto de la velocidad, al pasar de una velocidad dada a la próxima admisible. El paso gradual de una etapa de variación a otra, en muchos casos, determina la calidad de la producción.

### **1.2.2.3 Economía del accionamiento**

Se relaciona con los gastos de los elementos a considerar en la implementación de la variación de velocidad. En un accionamiento se persigue un alto rendimiento, obteniéndose así una alta calidad del proceso tecnológico y se cubren con rapidez los costos de llevarlo a cabo.

### **1.2.2.4 La estabilidad de trabajo a una velocidad preestablecida**

Esto es el cambio de la velocidad de rotación cuando hay una desviación del par de carga y depende de la rigidez de la característica mecánica; esta estabilidad es tanto mayor cuanto mayor sea la rigidez de la característica.

### **1.2.2.5 Carga admisible del motor**

Esto depende del método de variación. La variación del par de carga según sea la velocidad, es distinta en los diferentes mecanismos de producción, algunos exigen la variación manteniendo el momento o torque constante, otros no. Como regla general la capacidad de sostener una carga por parte del motor está determinado por el calentamiento que sufra éste cuando se le aplican varios pasos de carga. La carga admisible al trabajar en características variables, se define por la magnitud de corriente nominal y será distinta para los diversos sistemas de variación.

### 1.2.3 TIPOS DE CONTROL

Existen algunas maneras de controlar un motor de inducción, en torque, velocidad o posición, éstas pueden ser categorizadas en dos grupos: el control escalar y el control por vector.

#### 1.2.3.1 Control Escalar

Control escalar significa que las variables son controladas solo en magnitud y las señales de realimentación y comando son proporcionales a cantidades de DC. El método de control escalar solo puede manejar la frecuencia del estator usando un voltaje o una corriente como comando. En medio del método escalar conocido se asume que variando los voltajes del estator en proporción con la frecuencia, el torque se mantiene constante.

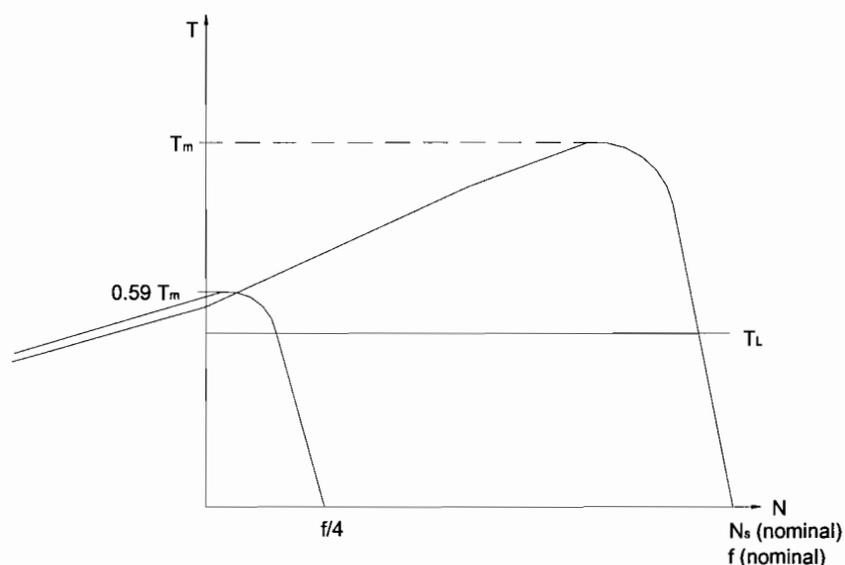
Este método está basado en características en estado estable del motor y el supuesto de que los voltajes y corrientes del estator son sinusoidales. Las ventajas de esta técnica de control son su simplicidad, es fácil y rápida de programar y requiere solo ciertas aptitudes de cálculo. Los inconvenientes son la pobre capacidad de respuesta a cambios en la carga y la eficiencia durante estos puntos de operación. Un controlador de velocidad que toma en consideración cambios de torque y evite lapsos indeseados puede raramente ser obtenido con un control de lazo abierto  $V / f$ .

Se necesita que las características torque-velocidad se distingan por su alta rigidez y el motor disponga de una alta capacidad de sobrecarga, esto se consigue manteniendo el flujo magnético constante, por esto es necesario variar la frecuencia manteniendo la siguiente relación invariable:

$$\frac{V}{f} = \Phi = \text{constante.} \quad (1.1)$$

Donde  $V$  es voltaje  
 $f$  es frecuencia  
 $\phi$  es flujo

Las características mecánicas ilustrativas cuando se mantiene la relación de flujo constante, se puede observar en la siguiente figura:



**Fig. 1.5** Características mecánicas de un motor asincrónico con control de la velocidad de rotación variando la frecuencia.

Matemáticamente, la desventaja sustancial de este tipo de variación ocurre a bajas velocidades, es decir, a bajas frecuencias ocurre una disminución sustancial del flujo magnético y por consiguiente también se disminuye el torque máximo, analizando la siguiente fórmula<sup>[1]</sup>:

$$T_{\max} = \frac{3}{2N_s} \frac{V_1^2}{R_1 \pm \sqrt{R_1^2 + X_c^2}} \quad (1.2)$$

*siendo*

$$X_C = X_1 + X'_2$$

$V_1$  = voltaje de fase del estator

$N_s$  = velocidad de sincronismo

$X'_2$  = reactancia del rotor referida al estator

$X_1$  = reactancia del estator

Cuando las frecuencias son grandes, la magnitud  $R_1$  respecto a  $X_C$  es considerablemente menor y por esto se puede considerar que:

$$T_{\max} = \frac{3V_1^2}{2N_s X_C}$$

Puesto que  $X_C$  y  $N_s$  son proporcionales a la frecuencia  $f$ , resulta:

$$T_{\max} \approx \frac{V_1^2}{f^2}$$

Y esta relación es constante.

Al disminuir la frecuencia las relaciones mencionadas se alteran, ya que la reactancia de dispersión  $X_C = X_1 + X'_2$  resulta de magnitud comparable con la resistencia del estator  $R_1$  o incluso menor. La influencia de la caída de tensión en el estator repercute intensamente y esto hace que disminuya el torque máximo. El mantenimiento de la capacidad suficiente de sobrecarga del motor requiere que a bajas frecuencias la tensión disminuya en menor grado que la misma frecuencia.

Si la frecuencia de alimentación es reducida de su valor nominal, manteniendo el voltaje primario constante el flujo debería incrementar.

$$E_1 = 4.44\phi_m \cdot fK_n$$

$$V_1 \approx K\phi_m f$$

siendo

$$E_1 = f.e.m. \text{ del motor}$$

La operación arriba del nivel del flujo nominal resulta en un incremento de las pérdidas en el cobre y una alta corriente de magnetización debido al indeseable alto nivel de saturación magnética.

Con el propósito de mantener la operación a la densidad de flujo nominal cuando la velocidad es variada, es necesario variar la f.e.m.  $E_1$  proporcional a la variación de  $f$ .

$$\frac{E_1}{f} = 4.44\phi_m \cdot K_n$$

La fuerza electromotriz  $E_1$  se varía indirectamente variando el voltaje de alimentación al estator  $V_1$ . Por lo tanto la relación  $V_1/f_1$  se debe mantener constante.

A bajas frecuencias el problema en la práctica se ha solucionado realizando una compensación de la caída  $I_xR$ .

En síntesis, este tipo de control se percibe como tres controles de sistemas monofásicos separados, en lugar de un control de un sistema trifásico. Algunos de los mayores inconvenientes se presentan a continuación:

- Los modelos y características de la máquina usados son válidos solo en estado estable. Esto causa que el control permita altos picos de voltaje y corrientes transitorios. Esto perjudica no solo el funcionamiento dinámico del accionamiento sino también la eficiencia en la conversión de energía.

Adicionalmente, los componentes de potencia deben ser sobredimensionados para soportar los picos eléctricos transitorios.

- No se considera interacciones entre las fases.
- Finalmente, la estructura del control debe ser dedicada acorde al tipo de motor (sincrónico o asincrónico).

### **1.2.3.2 Control por Vector de Campo**

El control por vector de campo se refiere no solamente a la magnitud sino también a la fase de las variables. Matrices y vectores son usados para representar las cantidades del control. Este método toma en consideración no solamente sucesivos estados estables sino ecuaciones matemáticas reales que describen el motor. Los resultados del control obtenido tienen una mejor dinámica para variaciones de torque en un extenso rango de variación de velocidad.

Aunque el motor de inducción tiene una muy simple estructura, el modelo matemático es complejo debido a factores asociados entre un gran número de variables y a las no-linealidades. El control de campo orientado (FOC) ofrece una solución para evadir la necesidad de solucionar ecuaciones de alto orden y lograr un eficiente control con muy buena dinámica.

Este método necesita más cálculos que el esquema de control estándar  $V/f$  constante. Esto puede ser solucionado utilizando cálculos incluidos en el procesamiento digital de señales (DSP) y obteniendo las siguientes ventajas:

- Capacidad de torque nominal para bajos valores de velocidad.
- Mejor funcionamiento dinámico.
- Mejor eficiencia para cada punto de operación en un amplio rango de velocidad.
- Desacoplar el control del torque y flujo.
- Pequeño período de capacidad de sobrecarga
- Cuatro cuadrantes de operación.

El control por vector de campo consiste en controlar las componentes de la corriente en el estator, representados por un vector, en un eje de referencia  $qd$ . El sistema de control por vector requiere resolver las ecuaciones del modelo dinámico del motor de inducción y retomar los valores instantáneos de corriente y voltaje para calcular las variables a ser controladas. Las máquinas controladas por campo orientado necesitan dos constantes como referencias de entrada: la componente del torque (alineada con el eje  $q$ ) y la componente del flujo (alineada con el eje  $d$ ). Como el control por campo orientado está simplemente basado en proyecciones, la estructura de control maneja cantidades eléctricas instantáneas. Esto hace el control preciso, ya sea en operación en estado estable o transitorio.

El torque eléctrico del motor de inducción puede ser descrito por medio de la interacción entre las corrientes del rotor y las ondas de flujo que resultan de las corrientes inducidas del estator. Una amplia explicación de este método que es el que se empleará para la determinación del torque de la máquina de inducción, se lo hará con detalle mientras se desarrolla la literatura del trabajo.

## **CAPÍTULO 2**

# **MODELO DE LA MÁQUINA DE INDUCCIÓN TRIFÁSICA**

## 2 MODELO DE LA MÁQUINA DE INDUCCIÓN TRIFÁSICA

### 2.1 MODELO DE LA MÁQUINA DE INDUCCIÓN TRIFÁSICA EN VARIABLES $abc$ <sup>[2]</sup>

El modelo de la máquina de inducción en variables  $abc$  hace referencia al modelo de la máquina de inducción correspondiente a las variables de voltaje, corriente y flujo magnético de las tres fases: a, b, y c.

#### 2.1.1 CAMPO MAGNÉTICO GIRATORIO Y DESLIZAMIENTO

Un set balanceado de corrientes trifásicas fluyendo en un set de devanados trifásicos simétricamente distribuidos produce un campo magnético giratorio dado por<sup>[3]</sup>:

$$F(\theta_a^e, t) = \frac{3}{2} \frac{4}{\pi} \frac{N}{P} \text{Im} \cos(\theta_a^e - \omega_e t) \quad A.vuelta \quad (2.1)$$

Donde  $\theta_a^e$  es el ángulo eléctrico medido desde el eje de la fase a  
 $\omega_e$  ( $=2\pi f_e$ ) es la velocidad angular del campo magnético del estator en radianes eléctricos por segundo  
 $f_e$  es la frecuencia de las corrientes de excitación, y  
 $P$  número de polos.

En radianes mecánicos por segundo la velocidad sincrónica es:

$$\omega_{sm} = \frac{2}{P} \omega_e \quad rad/s \quad (2.2)$$

En revoluciones por minuto, la velocidad sincrónica es:

$$N_s = \frac{60\omega_{sm}}{2\pi} = \frac{120f_e}{P} \quad \text{rev/min} \quad (2.3)$$

Cuando el rotor gira a una velocidad estable  $\omega_{rm}$  (radianes mecánicos por segundo), la velocidad relativa o deslizamiento entre el rotor y el campo rotativo magnético sincrónico del estator,  $F$ , es:

$$\text{velocidad de deslizamiento} = \omega_{sm} - \omega_{rm} \quad (2.4)$$

El deslizamiento por unidad, también referido simplemente como deslizamiento, es definido como velocidad de deslizamiento normalizada, esto es:

$$s \equiv \frac{\omega_{sm} - \omega_{rm}}{\omega_{sm}} = \frac{\omega_e - \omega_r}{\omega_e} \quad (2.5)$$

El deslizamiento  $s$  es negativo en operación como generador cuando el rotor gira sobre la velocidad sincrónica.

La velocidad de deslizamiento puede ser expresada como  $s\omega_e$  o  $s\omega_{sm}$  y la frecuencia de deslizamiento como  $sf_e$ . Cuando  $\omega_{rm}$  es menor que la velocidad sincrónica,  $\omega_{sm}$ , los conductores del rotor están deslizándose hacia atrás a una velocidad de  $s\omega_{sm}$  relativa al flujo de entrehierro en movimiento hacia delante.

Como resultado, los voltajes inducidos en los devanados del rotor debidos al flujo de entrehierro rotando sincrónicamente tienen una frecuencia de deslizamiento de  $sf_e$ . Cuando el circuito del rotor está cerrado, debido a los voltajes inducidos circularán corrientes en el circuito del rotor. La magnitud de las corrientes que fluyen en el circuito del rotor es determinada por la magnitud de los voltajes inducidos en el rotor y la impedancia del circuito del rotor a la frecuencia de deslizamiento.

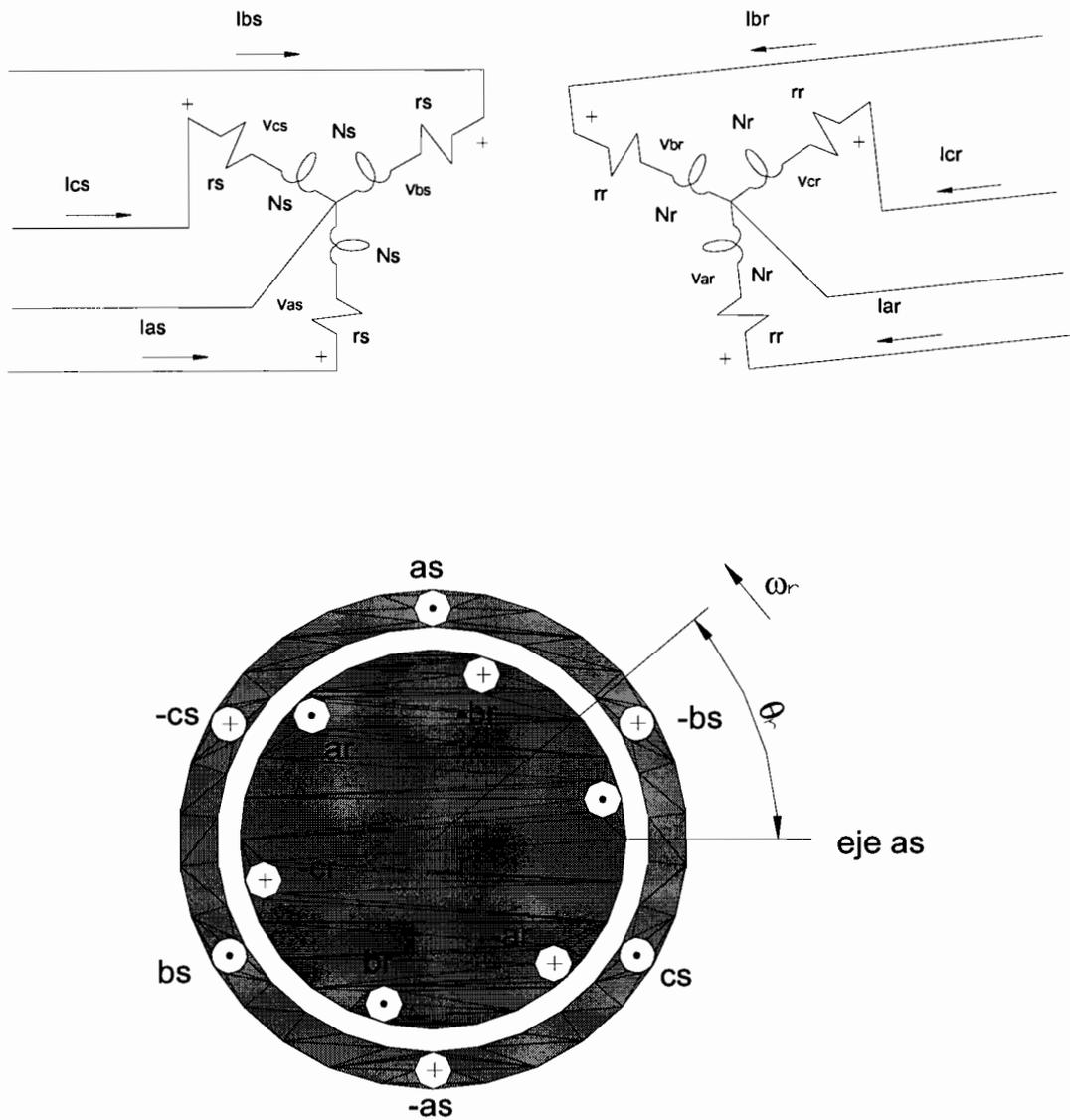
En el arranque,  $\omega_{rm}$  es cero o  $s=1$ , de este modo la frecuencia de deslizamiento en el arranque es  $f_e$ . A medida que el motor acelera hacia la velocidad sincrónica, la frecuencia de deslizamiento decrece. Así como con el flujo de corrientes en el estator, las corrientes en el rotor establecerán su propio campo magnético giratorio que rota a  $s\omega_{sm}$  rad/s relativo al rotor. A partir de que el rotor por sí mismo está rotando a  $\omega_{rm}$ , la velocidad absoluta del campo magnético giratorio del rotor puede demostrarse que es igual a la velocidad sincrónica, esto es

$$\omega_{rm} + s\omega_{sm} = \omega_{sm} \quad rad\ mec./s \quad (2.6)$$

Con los campos magnéticos giratorios del estator y el rotor rotando a la misma velocidad, en estado estable se produciría un torque estable cuando las diferencias de magnitud y fase de estos campos son constantes.

### 2.1.2 ECUACIONES DE VOLTAJE EN VARIABLES *abc*

El modelo del circuito idealizado de una máquina trifásica de inducción se muestra en la figura 2.1. Los bobinados del estator son idénticos, sinusoidales y distribuidos, desplazados  $120^\circ$  con un número de vueltas  $N_s$  y resistencia  $r_s$ . Para el propósito que se tiene, los devanados del rotor serán también considerados como tres devanados idénticos, sinusoidales, distribuidos, desplazados  $120^\circ$ , con un número de vueltas  $N_r$  y resistencia  $r_r$ .



**Fig. 2.1** Modelo idealizado del circuito de la máquina de inducción trifásica

Utilizando el circuito aproximado de la figura y la notación del motor, usándose el subíndice  $s$  para denotar variables y parámetros asociados con el estator, y el subíndice  $r$  para denotar variables y parámetros asociados con el rotor, las ecuaciones de voltaje de los circuitos magnéticos asociados al estator y al rotor pueden ser escritas de la siguiente manera:

### Ecuaciones de voltaje del estator

$$\begin{aligned}
 v_{as} &= i_{as}r_s + \frac{d\lambda_{as}}{dt} & V \\
 v_{bs} &= i_{bs}r_s + \frac{d\lambda_{bs}}{dt} & V \\
 v_{cs} &= i_{cs}r_s + \frac{d\lambda_{cs}}{dt} & V
 \end{aligned} \tag{2.7}$$

### Ecuaciones de voltaje del rotor

$$\begin{aligned}
 v_{ar} &= i_{ar}r_r + \frac{d\lambda_{ar}}{dt} & V \\
 v_{br} &= i_{br}r_r + \frac{d\lambda_{br}}{dt} & V \\
 v_{cr} &= i_{cr}r_r + \frac{d\lambda_{cr}}{dt} & V
 \end{aligned} \tag{2.8}$$

Donde  $\frac{d\lambda}{dt}$  representa las concatenaciones de flujo

#### 2.1.3 ECUACIONES DE CONCATENACIÓN DE FLUJO *abc*

En notación de matriz, las concatenaciones de flujo de los devanados del estator y del rotor, en términos de las inductancias y corrientes, pueden ser escritas compactamente como<sup>[4]</sup>:

$$\begin{bmatrix} \lambda_s^{abc} \\ \lambda_r^{abc} \end{bmatrix} = \begin{bmatrix} L_{ss}^{abc} & L_{sr}^{abc} \\ L_{rs}^{abc} & L_{rr}^{abc} \end{bmatrix} \begin{bmatrix} i_s^{abc} \\ i_r^{abc} \end{bmatrix} \quad \text{Wb.vuelta} \tag{2.9}$$

donde:

$$\begin{aligned}
 \lambda_s^{abc} &= (\lambda_{as}, \lambda_{bs}, \lambda_{cs})^t \\
 \lambda_r^{abc} &= (\lambda_{ar}, \lambda_{br}, \lambda_{cr})^t \\
 i_s^{abc} &= (i_{as}, i_{bs}, i_{cs})^t \\
 i_r^{abc} &= (i_{ar}, i_{br}, i_{cr})^t
 \end{aligned} \tag{2.10}$$

y el superíndice t denota el arreglo transpuesto.

Las submatrices de las inductancias de los devanados estator-a-estator y rotor-a-rotor son de la forma:

$$L_{ss}^{abc} = \begin{bmatrix} L_{ls} + L_{ss} & L_{sm} & L_{sm} \\ L_{sm} & L_{ls} + L_{ss} & L_{sm} \\ L_{sm} & L_{sm} & L_{ls} + L_{ss} \end{bmatrix} \quad H \tag{2.11}$$

$$L_{rr}^{abc} = \begin{bmatrix} L_{lr} + L_{rr} & L_{rm} & L_{rm} \\ L_{rm} & L_{lr} + L_{rr} & L_{rm} \\ L_{rm} & L_{rm} & L_{lr} + L_{rr} \end{bmatrix} \quad H \tag{2.12}$$

Las submatrices de las inductancias mutuas estator-a-rotor son dependientes del ángulo del rotor, esto es<sup>[4]</sup>:

$$L_{sr}^{abc} = [L_{rs}^{abc}]^t = L_{sr} \begin{bmatrix} \cos \theta_r & \cos \left( \theta_r + \frac{2\pi}{3} \right) & \cos \left( \theta_r - \frac{2\pi}{3} \right) \\ \cos \left( \theta_r - \frac{2\pi}{3} \right) & \cos \theta_r & \cos \left( \theta_r + \frac{2\pi}{3} \right) \\ \cos \left( \theta_r + \frac{2\pi}{3} \right) & \cos \left( \theta_r - \frac{2\pi}{3} \right) & \cos \theta_r \end{bmatrix} \quad H \tag{2.13}$$

donde  $L_{ls}$  es la inductancia de dispersión por fase del estator  
 $L_{lr}$  es la inductancia de dispersión por fase del rotor

- $L_{ss}$  es la inductancia propia del estator  
 $L_{rr}$  es la inductancia propia del rotor  
 $L_{sm}$  es la inductancia mutua entre los devanados del estator  
 $L_{rm}$  es la inductancia mutua entre los bobinados del rotor  
 $L_{sr}$  es el valor pico de la inductancia mutua entre el estator y el rotor.

Quando se expresan las ecuaciones de voltaje en variables  $abc$  es conveniente referir todas las variables del rotor a los devanados del estator<sup>[5]</sup>.

$$i'_{abc} = \frac{N_r}{N_s} i_{abc} \quad (2.14)$$

$$v'_{abc} = \frac{N_s}{N_r} v_{abc} \quad (2.15)$$

$$\lambda'_{abc} = \frac{N_s}{N_r} \lambda_{abc} \quad (2.16)$$

$$L_{ss} = \frac{N_s}{N_r} L_{sr} \quad (2.17)$$

Entonces, se define:

$$L'_{sr} = \frac{N_s}{N_r} L_{sr} \quad (2.18)$$

$$L'_{sr} = L_{ss} \begin{bmatrix} \cos \theta_r & \cos \left( \theta_r + \frac{2\pi}{3} \right) & \cos \left( \theta_r - \frac{2\pi}{3} \right) \\ \cos \left( \theta_r - \frac{2\pi}{3} \right) & \cos \theta_r & \cos \left( \theta_r + \frac{2\pi}{3} \right) \\ \cos \left( \theta_r + \frac{2\pi}{3} \right) & \cos \left( \theta_r - \frac{2\pi}{3} \right) & \cos \theta_r \end{bmatrix} \quad (2.19)$$

$L_{rr}$  puede expresarse como:

$$L_{rr} = \left( \frac{N_r}{N_s} \right)^2 L_{ss} \quad (2.20)$$

se sabe que:

$$L_{rr}^{abc} = \left( \frac{N_s}{N_r} \right)^2 L_{rr}^{abc} \quad (2.21)$$

entonces, de la ecuación (2.12):

$$L_{rr}^{abc} = \begin{bmatrix} L'_{lr} + L_{ss} & L_{sm} & L_{sm} \\ L_{sm} & L'_{lr} + L_{ss} & L_{sm} \\ L_{sm} & L_{sm} & L'_{lr} + L_{ss} \end{bmatrix} H \quad (2.22)$$

donde:

$$L'_{lr} = \left( \frac{N_s}{N_r} \right)^2 L_{lr} \quad (2.23)$$

Las concatenaciones de flujo ahora pueden expresarse como:

$$\begin{bmatrix} \lambda_s^{abc} \\ \lambda_r^{abc} \end{bmatrix} = \begin{bmatrix} L_{ss}^{abc} & L_{sr}^{abc} \\ (L_{sr}^{abc})^t & L_{rr}^{abc} \end{bmatrix} \begin{bmatrix} i_s^{abc} \\ i_r^{abc} \end{bmatrix} \quad (2.24)$$

Las ecuaciones de voltaje expresadas en términos de variables de máquina referidas a los bobinados del estator pueden ser escritas como:

$$\begin{bmatrix} v_s^{abc} \\ v_r^{abc} \end{bmatrix} = \begin{bmatrix} r_s^{abc} + pL_{ss}^{abc} & pL_{sr}^{abc} \\ p(L_{sr}^{abc})^t & r_r^{abc} + pL_{rr}^{abc} \end{bmatrix} \begin{bmatrix} i_s^{abc} \\ i_r^{abc} \end{bmatrix} \quad (2.25)$$

donde:

$$r_r^{abc} = \left( \frac{N_s}{N_r} \right)^2 r_r^{abc} \quad (2.26)$$

Se debe notar que la máquina idealizada es descrita por seis ecuaciones diferenciales de primer orden, una por cada devanado. Estas ecuaciones diferenciales están acopladas entre sí mediante las inductancias mutuas entre los distintos devanados. En particular, los términos acoplados entre estator y rotor son función de la posición del rotor; así, cuando el rotor rota, estos términos acoplados varían con el tiempo.

Transformaciones matemáticas como la  $dq$  ó  $\alpha\beta$  pueden facilitar el cálculo de la solución transitoria del modelo de la máquina de inducción presentado mediante la transformación de las ecuaciones diferenciales con inductancias variantes en el tiempo a ecuaciones diferenciales con inductancias constantes. Con este objetivo se desarrollará el modelo de la máquina de inducción trifásica en el marco de referencia  $qd0$  a continuación.

## **2.2 MODELO DE LA MÁQUINA DE INDUCCIÓN UTILIZANDO UN MARCO DE REFERENCIA**

### **2.2.1 MODELO DE LA MÁQUINA EN EL MARCO DE REFERENCIA ARBITRARIO $qd0$ <sup>[6]</sup>**

En las ecuaciones que describen el funcionamiento de las máquinas de inducción se encuentra que algunas de las inductancias de la máquina son funciones de la velocidad del rotor, con lo cual los coeficientes de las ecuaciones diferenciales de voltaje que describen el comportamiento de estas máquinas son variantes en el tiempo, excepto cuando el rotor está bloqueado. Un cambio en las variables, es

con frecuencia usado para reducir la complejidad de estas ecuaciones diferenciales.

Existen algunos cambios de variables que son usados, pero todos están contenidos en una transformación general que refiere las variables de máquina a un marco de referencia que rota a una velocidad angular arbitraria. Todas las transformaciones conocidas son obtenidas de ésta, simplemente asignando la velocidad de rotación del marco de referencia.

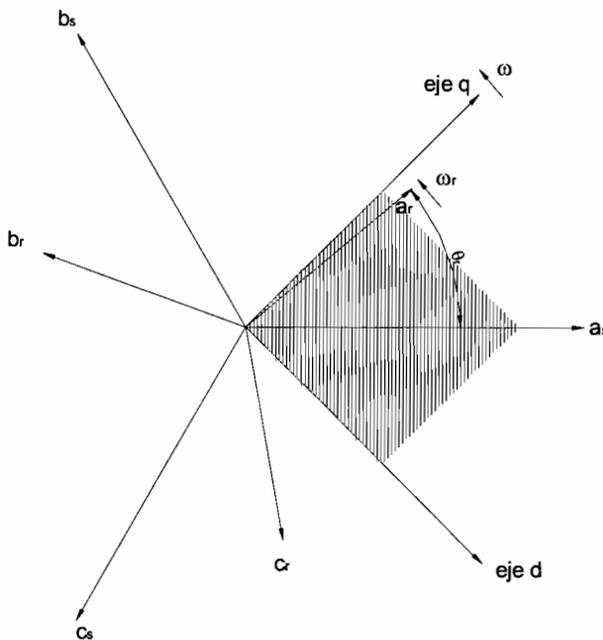
Estos cambios de variables son usados no solo en el análisis de máquinas de AC para eliminar las inductancias variables en el tiempo sino también para el análisis de algunos componentes estáticos o parámetros constantes en sistemas de potencia.

La máquina de inducción ideal se asume que tiene entrehierro simétrico. Los ejes de referencia  $qd0$  son usualmente seleccionados basándose en la conveniencia o compatibilidad con las representaciones de componentes de otras redes. Primero se deducirá las ecuaciones de la máquina de inducción en el marco de referencia arbitrario que rota a la velocidad  $\omega$  en la dirección de la rotación del rotor. La velocidad angular asociada con el cambio de variables no se especifica. El marco de referencia puede rotar a cualquier velocidad constante o variable o puede permanecer estacionario. La connotación de *arbitrario* parte del hecho de que la velocidad angular de la transformación no se especifica y puede ser seleccionada arbitrariamente para dar la solución del sistema de ecuaciones o satisfacer las restricciones del sistema.

Como en el modelo anterior en variables  $abc$ , se empezará con las ecuaciones de voltaje de la máquina por fase. Aplicando transformación a la referencia arbitraria  $qd0$  a estas ecuaciones, se obtienen las ecuaciones  $qd0$  correspondientes. La relación entre las cantidades  $abc$  y  $qd0$  de un marco de referencia rotando a una velocidad angular  $\omega$ , se muestran en la figura 2.2.

La ecuación de transformación de variables  $abc$  al marco de referencia  $qd0$  está dada por:

$$\begin{bmatrix} f_q \\ f_d \\ f_0 \end{bmatrix} = [T_{qd0}(\theta)] \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix} \quad (2.27)$$



**Fig. 2.2** Relación entre  $abc$  y  $qd0$  arbitrario

Donde la variable  $f$  puede representar voltajes de fase, corrientes, o concatenaciones de flujo de la máquina.

El ángulo de transformación,  $\theta(t)$ , entre el eje  $q$  del marco de referencia rotando a una velocidad  $\omega$  y el eje  $a$  puede ser expresado como:

$$\theta(t) = \int \omega(t) dt + \theta(0) \quad \text{rad elect.} \quad (2.28)$$

Del mismo modo, el ángulo del rotor,  $\theta_r(t)$ , entre los ejes de las fases a del estator y el rotor para un rotor rotando con velocidad  $\omega_r(t)$  puede ser expresado como:

$$\theta_r(t) = \int \omega_r(t) dt + \theta_r(0) \quad \text{rad elect.} \quad (2.29)$$

Los ángulos  $\theta(0)$  y  $\theta_r(0)$ , son los valores iniciales de estos ángulos.

La matriz de transformación  $qd0$ ,  $[T_{qd0}(\theta)]$ , es:

$$[T_{qd0}(\theta)] = \frac{2}{3} \begin{bmatrix} \cos \theta & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \text{sen } \theta & \text{sen}\left(\theta - \frac{2\pi}{3}\right) & \text{sen}\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2.30)$$

y su inversa es:

$$[T_{qd0}(\theta)]^{-1} = \begin{bmatrix} \cos \theta & \text{sen } \theta & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \text{sen}\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \text{sen}\left(\theta + \frac{2\pi}{3}\right) & 1 \end{bmatrix} \quad (2.31)$$

Aunque la transformación al marco arbitrario de referencia es un cambio de variables y no requiere connotación física, con frecuencia es conveniente visualizar las ecuaciones de transformación como relaciones trigonométricas entre variables como se muestra en la figura 2.2.

En particular, se tiene que las variables  $f_{qs}$  y  $f_{ds}$  son ortogonales entre sí y rotan a la velocidad angular  $\omega$ ; además, pueden ser consideradas como los ejes

magnéticos de los nuevos devanados creados por el cambio de variables; con lo cual  $f_{as}$ ,  $f_{bs}$  y  $f_{cs}$ , pueden ser consideradas como variables estacionarias que representan los ejes magnéticos de los devanados del estator, desplazados entre sí  $120^\circ$ . Es importante notar que las variables  $0$  no están asociadas con el marco arbitrario de referencia. Más bien, las variables  $0$  están relacionadas aritméticamente con las variables  $abc$ , con independencia de  $\theta$ . Es también importante no confundir  $f_{as}$ ,  $f_{bs}$  y  $f_{cs}$ . Ellas son cantidades instantáneas que son función del tiempo.

### 2.2.2 ECUACIONES DE VOLTAJE $qd0$

En notación matricial, las ecuaciones de voltaje de los devanados  $abc$  del estator pueden ser expresadas como:

$$v_s^{abc} = p\lambda_s^{abc} + r_s^{abc}i_s^{abc} \quad (2.32)$$

Aplicando la transformación,  $[T_{qd0}(\theta)]$ , al voltaje, concatenación de flujo y corriente, la ecuación 2.32 se convierte en:

$$v_s^{qd0} = [T_{qd0}(\theta)]p[T_{qd0}(\theta)]^{-1}[\lambda_s^{qd0}] + [T_{qd0}(\theta)]r_s^{abc}[T_{qd0}(\theta)]^{-1}[i_s^{qd0}] \quad (2.33)$$

El siguiente término derivable en el tiempo puede ser expresado como:

$$p[T_{qd0}(\theta)]^{-1}[\lambda_s^{qd0}] = \begin{bmatrix} -\operatorname{sen}\theta & \cos\theta & 0 \\ -\operatorname{sen}\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) & 0 \\ -\operatorname{sen}\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) & 0 \end{bmatrix} \frac{d\theta}{dt} [\lambda_s^{qd0}] + [T_{qd0}(\theta)]^{-1}[p\lambda_s^{qd0}] \quad (2.34)$$

Sustituyendo la ecuación 2.34 en la ecuación 2.33 y reordenando los términos, se obtiene:

$$v_s^{qd0} = \omega \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_s^{qd0} + p \lambda_s^{qd0} + r_s^{qd0} i_s^{qd0} \quad (2.35)$$

donde:

$$\omega = \frac{d\theta}{dt} \quad y \quad r_s^{qd0} = r_s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.36)$$

De igual modo, las cantidades del rotor deben ser transformadas sobre el mismo marco  $qd$ . Es claro, que las ecuaciones de transformación para los circuitos del rotor son las mismas que para circuitos estacionarios con  $(\theta - \theta_r)$  usado como el desplazamiento angular en vez de  $\theta$ , como se puede observar en la figura 2.2. Por consiguiente, todas las ecuaciones para los circuitos del estator son válidas para el rotor si  $\theta$  es reemplazado por  $(\theta - \theta_r)$  y  $\omega$  por  $(\omega - \omega_r)$ . Usando la transformación  $[T_{qd0}(\theta)]$  en las ecuaciones de voltaje del rotor, se obtiene las siguientes ecuaciones de voltaje  $qd0$  para los devanados del rotor:

$$v_r^{qd0} = (\omega - \omega_r) \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \lambda_r^{qd0} + p \lambda_r^{qd0} + r_r^{qd0} i_r^{qd0} \quad (2.37)$$

### 2.2.3 ECUACIONES DE CONCATENACIÓN DE FLUJO $qd0$

Las concatenaciones de flujo  $qd0$  del estator son obtenidas aplicando  $[T_{qd0}(\theta)]$  a las concatenaciones de flujo  $abc$  del estator en la ec. 2.9, esto es:

$$\lambda_s^{qd0} = [T_{qd0}(\theta)](L_{ss}^{abc} i_s^{abc} + L_{sr}^{abc} i_r^{abc}) \quad (2.38)$$

Usando la transformación inversa apropiada para reemplazar las corrientes  $abc$  del estator y el rotor por las correspondientes corrientes  $qd0$ , la ec. 2.38 se convierte en:

$$\begin{aligned} \lambda_s^{qd0} &= [T_{qd0}(\theta)]L_{ss}^{abc} [T_{qd0}(\theta)]^{-1} i_s^{qd0} + [T_{qd0}(\theta)]L_{sr}^{abc} [T_{qd0}(\theta - \theta_r)]^{-1} i_r^{qd0} \\ &= \begin{bmatrix} L_{ls} + \frac{3}{2}L_{ss} & 0 & 0 \\ 0 & L_{ls} + \frac{3}{2}L_{ss} & 0 \\ 0 & 0 & L_{ls} \end{bmatrix} i_s^{qd0} + \begin{bmatrix} \frac{3}{2}L_{sr} & 0 & 0 \\ 0 & \frac{3}{2}L_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} i_r^{qd0} \end{aligned} \quad (2.39)$$

En forma similar, las concatenaciones de flujo  $qd0$  del rotor están dadas por:

$$\begin{aligned} \lambda_r^{qd0} &= [T_{qd0}(\theta - \theta_r)]L_{rs}^{abc} [T_{qd0}(\theta)]^{-1} i_s^{qd0} + [T_{qd0}(\theta - \theta_r)]L_{rr}^{abc} [T_{qd0}(\theta - \theta_r)]^{-1} i_r^{qd0} \\ &= \begin{bmatrix} \frac{3}{2}L_{sr} & 0 & 0 \\ 0 & \frac{3}{2}L_{sr} & 0 \\ 0 & 0 & 0 \end{bmatrix} i_s^{qd0} + \begin{bmatrix} L_{lr} + \frac{3}{2}L_{rr} & 0 & 0 \\ 0 & L_{lr} + \frac{3}{2}L_{rr} & 0 \\ 0 & 0 & L_{lr} \end{bmatrix} i_r^{qd0} \end{aligned} \quad (2.40)$$

Las relaciones entre las concatenaciones de flujo del estator y rotor en las ec. 2.39 y 2.40 pueden ser expresadas compactamente como:

$$\begin{bmatrix} \lambda_{qs} \\ \lambda_{ds} \\ \lambda_{0s} \\ \lambda'_{qr} \\ \lambda'_{dr} \\ \lambda'_{0r} \end{bmatrix} = \begin{bmatrix} L_{ls} + L_m & 0 & 0 & L_m & 0 & 0 \\ 0 & L_{ls} + L_m & 0 & 0 & L_m & 0 \\ 0 & 0 & L_{ls} & 0 & 0 & 0 \\ L_m & 0 & 0 & L'_{lr} + L_m & 0 & 0 \\ 0 & L_m & 0 & 0 & L'_{lr} + L_m & 0 \\ 0 & 0 & 0 & 0 & 0 & L'_{lr} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i'_{qr} \\ i'_{dr} \\ i'_{0r} \end{bmatrix} \quad (2.41)$$

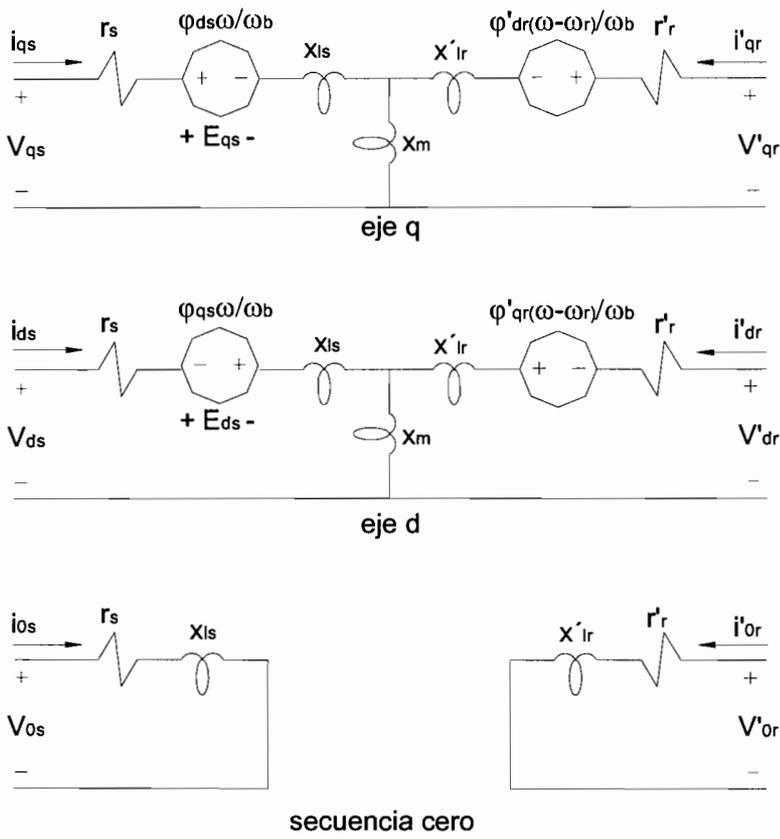
donde las cantidades primas denotan los valores referidos al lado del estator de las variables  $qd0$  análogas a las relaciones de las ec. 2.14, 2.16 y 2.23; y  $L_m$ , la inductancia de magnetización en el lado del estator, es:

$$L_m = \frac{3}{2} L_{ss} = \frac{3}{2} \frac{N_s}{N_r} L_{sr} = \frac{3}{2} \frac{N_s}{N_r} L_{rr} \quad (2.42)$$

Sustituyendo la eq. 2.41 en las ec. 2.35 y 2.37, y entonces agrupando los términos  $q$ ,  $d$ ,  $0$ , y  $\theta$  en las ecuaciones de voltaje resultantes, se obtendrá las ecuaciones de voltaje que sugieren el circuito equivalente de la figura 2.3 para la máquina de inducción en la referencia arbitraria de los ejes  $qd0$ .

*Ecuaciones de voltaje  $qd0$  del estator y el rotor*

$$\begin{aligned} v_{qs} &= p\lambda_{qs} + \omega\lambda_{ds} + r_s i_{qs} \\ v_{ds} &= p\lambda_{ds} + \omega\lambda_{qs} + r_s i_{ds} \\ v_{0s} &= p\lambda_{0s} + r_s i_{0s} \\ v'_{qr} &= p\lambda'_{qr} + (\omega - \omega_r)\lambda'_{dr} + r'_r i'_{qr} \\ v'_{dr} &= p\lambda'_{dr} - (\omega - \omega_r)\lambda'_{qr} + r'_r i'_{dr} \\ v'_{0r} &= p\lambda'_{0r} + r'_r i'_{0r} \end{aligned} \quad (2.43)$$



**Fig. 2.3** Representación del circuito equivalente de la máquina de inducción en un marco arbitrario de referencia.

Con frecuencia las ecuaciones de máquina son expresadas en términos de concatenaciones de flujo por segundo,  $\varphi$ 's, y reactancias,  $x$ 's, en lugar de concatenaciones de flujo,  $\lambda$ 's, e inductancias,  $L$ 's. Estas cantidades están interrelacionadas simplemente por el valor base de la frecuencia angular,  $\omega_b$ , esto es:

$$\varphi = \omega_b \lambda \quad V \text{ o por unidad} \quad (2.44)$$

y

$$x = \omega_b L \quad \Omega \text{ o por unidad} \quad (2.45)$$

donde  $\omega_b = 2\pi f_{nom}$  radianes eléctricos por segundo,  $f_{nom}$  constituye la frecuencia de la máquina en Hertz. Con formas de ondas complejas puede haber justificación para el uso del valor pico en lugar del valor rms del voltaje de fase de la máquina como el valor base. Las cantidades base con valor pico fase-neutro en vez del valor rms de una máquina de inducción trifásica de P polos con voltaje rms entre líneas,  $V_{nom}$ , y potencia en volt-amperios,  $S_{nom}$ , son las siguientes:

$$\begin{aligned}
 \text{voltaje base } V_b &= \sqrt{2/3} V_{nom} \\
 \text{volt - amperio base } S_b &= S_{nom} \\
 \text{corriente base pico } I_b &= 2S_b / 3V_b \\
 \text{impedancia base } Z_b &= V_b / I_b \\
 \text{torque base } T_b &= S_b / \omega_{bm}
 \end{aligned} \tag{2.46}$$

donde  $\omega_{bm} = 2\omega_b/P$ .

Las ecuaciones de la máquina de inducción simétrica en el marco de referencia arbitrario en término de las concatenaciones de flujo por segundo y las reactancias a la frecuencia base se resumen a continuación:

*Ecuaciones de voltaje qd0 del estator y el rotor*

$$\begin{aligned}
 v_{qs} &= \frac{p}{\omega_b} \dot{\varphi}_{qs} + \frac{\omega}{\omega_b} \varphi_{ds} + r_s i_{qs} \\
 v_{ds} &= \frac{p}{\omega_b} \dot{\varphi}_{ds} + \frac{\omega}{\omega_b} \varphi_{qs} + r_s i_{ds} \\
 v_{0s} &= \frac{p}{\omega_b} \dot{\varphi}_{0s} + r_s i_{0s} \\
 v'_{qr} &= \frac{p}{\omega_b} \dot{\varphi}'_{qr} + \left( \frac{\omega - \omega_r}{\omega_b} \right) \varphi'_{dr} + r'_r i'_{qr} \\
 v'_{dr} &= \frac{p}{\omega_b} \dot{\varphi}'_{dr} - \left( \frac{\omega - \omega_r}{\omega_b} \right) \varphi'_{qr} + r'_r i'_{dr} \\
 v'_{0r} &= \frac{p}{\omega_b} \dot{\varphi}'_{0r} + r'_r i'_{0r}
 \end{aligned} \tag{2.47}$$

donde:

$$\begin{bmatrix} \varphi_{qs} \\ \varphi_{ds} \\ \varphi_{0s} \\ \varphi'_{qr} \\ \varphi'_{dr} \\ \varphi'_{0r} \end{bmatrix} = \begin{bmatrix} x_{ls} + x_m & 0 & 0 & x_m & 0 & 0 \\ 0 & x_{ls} + x_m & 0 & 0 & x_m & 0 \\ 0 & 0 & x_{ls} & 0 & 0 & 0 \\ x_m & 0 & 0 & x'_{lr} + x_m & 0 & 0 \\ 0 & x_m & 0 & 0 & x'_{lr} + x_m & 0 \\ 0 & 0 & 0 & 0 & 0 & x'_{lr} \end{bmatrix} \begin{bmatrix} i_{qs} \\ i_{ds} \\ i_{0s} \\ i'_{qr} \\ i'_{dr} \\ i'_{0r} \end{bmatrix} \quad (2.48)$$

## 2.2.4 ECUACIÓN DE TORQUE $qd0$

La suma de la potencia instantánea de entrada a los seis devanados del estator y el rotor está dada por:

$$P_{in} = v_{as}i_{as} + v_{bs}i_{bs} + v_{cs}i_{cs} + v'_{ar}i'_{ar} + v'_{br}i'_{br} + v'_{cr}i'_{cr} \quad W \quad (2.49)$$

En términos de cantidades  $qd0$ , la potencia instantánea de entrada es:

$$P_{in} = \frac{3}{2} \left( v_{qs}i_{qs} + v_{ds}i_{ds} + 2v_{0s}i_{0s} + v'_{qr}i'_{qr} + v'_{dr}i'_{dr} + 2v'_{0r}i'_{0r} \right) \quad W \quad (2.50)$$

Usando las ec. 2.35 y 2.37 para sustituir los voltajes en la ec. 2.50, se obtiene tres tipos de términos:  $i^2r$ ,  $ip\lambda$ , y  $\omega\lambda i$ . Los términos  $i^2r$  representan las pérdidas en el cobre. Los términos  $ip\lambda$  son la proporción de intercambio de energía del campo magnético entre los devanados. Los términos  $\omega\lambda i$  representan la cantidad de energía convertida a trabajo mecánico. El torque electromagnético desarrollado por la máquina está dado por la suma de los términos  $\omega\lambda i$  divididos para la velocidad mecánica, esto es:

$$T_{em} = \frac{3}{2} \frac{P}{2\omega_r} \left[ \omega (\lambda_{ds}i_{qs} - \lambda_{qs}i_{ds}) + (\omega - \omega_r) (\lambda'_{dr}i'_{qr} - \lambda'_{qr}i'_{dr}) \right] \quad N.m. \quad (2.51)$$

Usando la relación de concatenación de flujo de la ec. 2.41, se puede demostrar que:

$$\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds} = -(\lambda'_{dr} i'_{qr} - \lambda'_{qr} i'_{dr}) = L_m (i'_{dr} i_{qs} - i'_{qr} i_{ds}) \quad (2.52)$$

La ecuación 2.51, puede así ser expresada en las siguientes formas:

$$\begin{aligned} T_{em} &= \frac{3}{2} \frac{P}{2} (\lambda'_{qr} i'_{dr} - \lambda'_{dr} i'_{qr}) && N.m. \\ &= \frac{3}{2} \frac{P}{2} (\lambda_{ds} i_{qs} - \lambda_{qs} i_{ds}) && N.m. \\ &= \frac{3}{2} \frac{P}{2} L_m (i'_{dr} i_{qs} - i'_{qr} i_{ds}) && N.m. \end{aligned} \quad (2.53)$$

Para propósitos de simulación, la selección de una forma sobre la otra es usualmente influenciada por cuáles variables están disponibles en otras partes de la simulación.

La ecuación de torque en términos de concatenaciones de flujo por segundo y reactancias está dada por:

$$\begin{aligned} T_{em} &= \frac{3}{2} \frac{P}{2\omega_r} \left[ \frac{\omega}{\omega_b} (\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds}) + \frac{\omega - \omega_r}{\omega_b} (\varphi'_{dr} i'_{qr} - \varphi'_{qr} i'_{dr}) \right] \\ &= \frac{3}{2} \frac{P}{2\omega_b} (\varphi'_{qr} i'_{dr} - \varphi'_{dr} i'_{qr}) && N.m. \\ &= \frac{3}{2} \frac{P}{2\omega_b} (\varphi_{ds} i_{qs} - \varphi_{qs} i_{ds}) && N.m. \\ &= \frac{3}{2} \frac{P}{2\omega_b} x_m (i'_{dr} i_{qs} - i'_{qr} i_{ds}) && N.m. \end{aligned} \quad (2.54)$$

### 2.2.5 MARCO DE REFERENCIA ESTACIONARIO

Los dos marcos de referencia comúnmente usados en el análisis de la máquina de inducción son los marcos de referencia estacionario y rotativo sincrónico. Generalmente, las condiciones de operación determinarán el marco de referencia más conveniente para propósitos de análisis y/o simulación. Cada uno tiene una ventaja para algún propósito como se especifica a continuación.

En el marco de referencia estacionario, las variables  $qd$  de la máquina están en el mismo marco que las variables normalmente usadas para la red de suministro de energía. Es una opción conveniente cuando la red de suministro es compleja.

Para estudios de sistemas de potencia, cargas de máquinas de inducción, junto con otros tipos de componentes de sistemas de potencia, las máquinas de inducción son frecuentemente simuladas en un sistema de marco de referencia rotativo sincrónico. Este marco de referencia se utiliza también para análisis de estabilidad dinámica de pequeñas señales sobre algún punto de operación específico, pues produce valores constantes de voltajes y corrientes de estado estable bajo condiciones balanceadas.

Sin embargo, para estudios transitorios de accionamientos de velocidad variable, usualmente es más conveniente simular la máquina de inducción y su convertidor en un marco de referencia estacionario, por lo que éste será el marco de referencia a utilizar en el desarrollo de este tema.

El análisis matemático simplificado de la máquina de inducción es aplicable a la solución de una variedad de problemas de estado estable y transitorio. Los voltajes de fase, corrientes, y concatenaciones de flujo están referidos a un set de ejes ortogonales que son estacionarios con respecto al estator.

Las ecuaciones diferenciales resultantes no sólo son lineales, sino que además tienen coeficientes constantes. Consecuentemente, el presente tratamiento no está limitado a problemas en los cuales la velocidad del rotor se mantiene

constante. En efecto, la mayor justificación para su uso se encuentra en el campo de problemas de velocidad variable. Las ecuaciones simplificadas deducidas son aplicables para cualquier máquina con entrehierro uniforme y devanados del estator y rotor distribuidos, sinusoidales y balanceados. Se asume que el rotor es liso y las inductancias propias de todos los devanados son independientes de la posición del rotor. Además los efectos de saturación, histéresis y corrientes de Eddy son despreciados.

#### **2.2.5.1 Análisis de una máquina de inducción en un marco de referencia estacionario<sup>[6]</sup>**

Una vez que se han deducido las ecuaciones de la máquina de inducción para el caso general, esto es en el marco de referencia arbitrario, las ecuaciones de la máquina en el marco de referencia estacionario pueden simplemente ser obtenidas haciendo la velocidad  $\omega = 0$ . Las ecuaciones de una máquina de inducción simétrica en términos de flujos y reactancias en el marco de referencia estacionario se resumen a continuación. Las representaciones del circuito equivalente correspondientes se tienen en la figura 2.4.

Para distinguir entre estos sistemas de referencia, las variables en el marco de referencia arbitrario se van a identificar con un superíndice s.

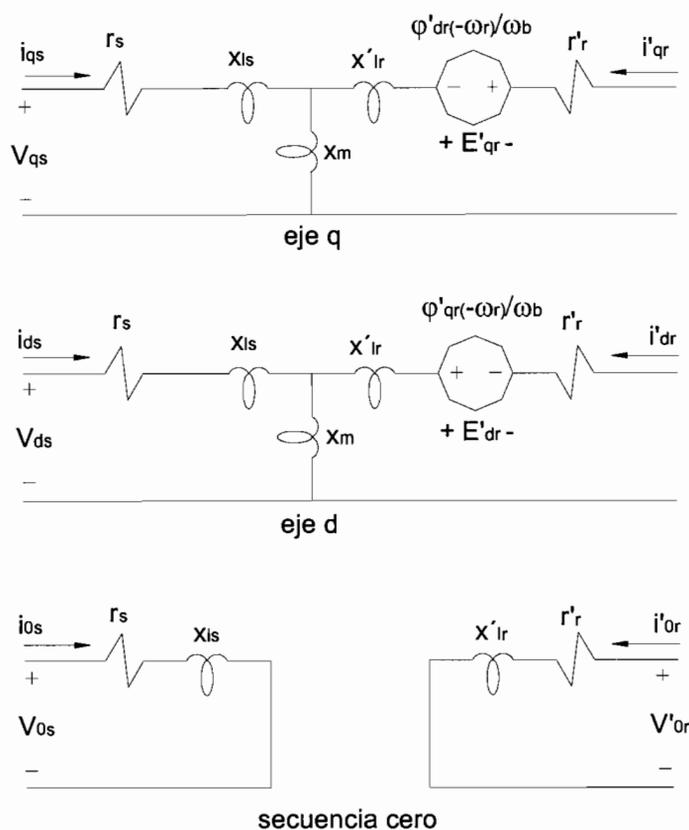
### Ecuaciones de voltaje del estator y el rotor

$$\begin{aligned}
 v_{qs}^s &= \frac{P}{\omega_b} \varphi_{qs}^s + r_s i_{qs}^s \\
 v_{ds}^s &= \frac{P}{\omega_b} \varphi_{ds}^s + r_s i_{ds}^s \\
 v_{0s}^s &= \frac{P}{\omega_b} \varphi_{0s}^s + r_s i_{0s}^s \\
 v_{qr}^{1s} &= \frac{P}{\omega_b} \varphi_{qr}^{1s} - \frac{\omega_r}{\omega_b} \varphi_{dr}^{1s} + r'_r i_{qr}^{1s} \\
 v_{dr}^{1s} &= \frac{P}{\omega_b} \varphi_{dr}^{1s} + \frac{\omega_r}{\omega_b} \varphi_{qr}^{1s} + r'_r i_{dr}^{1s} \\
 v_{0r}^{1s} &= \frac{P}{\omega_b} \varphi_{0r}^{1s} + r'_r i_{0r}^{1s}
 \end{aligned} \tag{2.55}$$

$$\begin{bmatrix} \varphi_{qs}^s \\ \varphi_{ds}^s \\ \varphi_{0s}^s \\ \varphi_{qr}^{1s} \\ \varphi_{dr}^{1s} \\ \varphi_{0r}^{1s} \end{bmatrix} = \begin{bmatrix} x_{ls} + x_m & 0 & 0 & x_m & 0 & 0 \\ 0 & x_{ls} + x_m & 0 & 0 & x_m & 0 \\ 0 & 0 & x_{ls} & 0 & 0 & 0 \\ x_m & 0 & 0 & x'_{lr} + x_m & 0 & 0 \\ 0 & x_m & 0 & 0 & x'_{lr} + x_m & 0 \\ 0 & 0 & 0 & 0 & 0 & x'_{lr} \end{bmatrix} \begin{bmatrix} i_{qs}^s \\ i_{ds}^s \\ i_{0s}^s \\ i_{qr}^{1s} \\ i_{dr}^{1s} \\ i_{0r}^{1s} \end{bmatrix} \tag{2.56}$$

### Ecuación de torque

$$\begin{aligned}
 T_{em} &= \frac{3}{2} \frac{P}{2\omega_b} (\varphi_{qr}^{1s} i_{dr}^{1s} - \varphi_{dr}^{1s} i_{qr}^{1s}) & N.m. \\
 &= \frac{3}{2} \frac{P}{2\omega_b} (\varphi_{ds}^s i_{qs}^s - \varphi_{qs}^s i_{ds}^s) & N.m. \\
 &= \frac{3}{2} \frac{P}{2\omega_b} x_m (i_{dr}^{1s} i_{qs}^s - i_{qr}^{1s} i_{ds}^s) & N.m.
 \end{aligned} \tag{2.57}$$



**Fig. 2.4** Representación del circuito equivalente de la máquina de inducción en el marco de referencia estacionario

Si:

$$x_{ss} = x_{ls} + x_m \quad y \quad x'_{rr} = x'_{lr} + x_m \quad (2.58)$$

y como las ecuaciones de voltaje están escritas en términos de corrientes y concatenaciones de flujo por segundo, sabiendo que las corrientes y concatenaciones de flujo están relacionadas entre sí, y ambas no pueden ser independientes, para la simulación se encontrará deseable expresar las ecuaciones de voltaje ya sea, en términos de corrientes o en términos de concatenaciones de flujo por segundo. Si las concatenaciones de flujo por

segundo son seleccionadas como variables independientes entonces la ec. 2.56 puede resolverse para las corrientes y ser escrita como:

$$\begin{bmatrix} i_{qs}^s \\ i_{ds}^s \\ i_{0s}^s \\ i_{qr}^{1s} \\ i_{dr}^{1s} \\ i_{0r}^{1s} \end{bmatrix} = \frac{1}{D} \begin{bmatrix} x'_{rr} & 0 & 0 & -x_m & 0 & 0 \\ 0 & x'_{rr} & 0 & 0 & -x_m & 0 \\ 0 & 0 & \frac{D}{x_{ls}} & 0 & 0 & 0 \\ -x_m & 0 & 0 & x_{ss} & 0 & 0 \\ 0 & -x_m & 0 & 0 & x_{ss} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{D}{x'_{lr}} \end{bmatrix} \begin{bmatrix} \varphi_{qs}^s \\ \varphi_{ds}^s \\ \varphi_{0s}^s \\ \varphi_{qr}^{1s} \\ \varphi_{dr}^{1s} \\ \varphi_{0r}^{1s} \end{bmatrix} \quad (2.59)$$

donde:

$$D = x_{ss} x'_{rr} - x_m^2 \quad (2.60)$$

Sustituyendo la ec. 2.59 para las corrientes en la ec. 2.55 se obtiene las ecuaciones de voltaje en términos de concatenaciones de flujo por segundo como se observa a continuación.

$$\begin{bmatrix} v_{qs}^s \\ v_{ds}^s \\ v_{0s}^s \\ v_{qr}^{1s} \\ v_{dr}^{1s} \\ v_{0r}^{1s} \end{bmatrix} = \begin{bmatrix} \frac{r_s x'_{rr}}{D} + \frac{p}{w_b} & 0 & 0 & -\frac{r_s x_m}{D} & 0 & 0 \\ 0 & \frac{r_s x'_{rr}}{D} + \frac{p}{w_b} & 0 & 0 & -\frac{r_s x_m}{D} & 0 \\ 0 & 0 & \frac{r_s}{x_{ls}} + \frac{p}{w_b} & 0 & 0 & 0 \\ -\frac{r'_r x_m}{D} & 0 & 0 & \frac{r'_r x_{ss}}{D} + \frac{p}{w_b} & -\frac{w_r}{w_b} & 0 \\ 0 & -\frac{r'_r x_m}{D} & 0 & \frac{w_r}{w_b} & \frac{r'_r x_{ss}}{D} + \frac{p}{w_b} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{r'_r}{x'_{lr}} + \frac{p}{w_b} \end{bmatrix} \begin{bmatrix} \varphi_{qs}^s \\ \varphi_{ds}^s \\ \varphi_{0s}^s \\ \varphi_{qr}^{1s} \\ \varphi_{dr}^{1s} \\ \varphi_{0r}^{1s} \end{bmatrix} \quad (2.61)$$

Es interesante notar que cada ecuación de voltaje  $q$  y  $d$  contiene dos derivadas de corriente cuando las corrientes son seleccionadas como variables independientes, cuando las concatenaciones de flujo son seleccionadas como variables independientes cada ecuación de voltaje tiene solo una derivada de concatenación de flujo. Esta propiedad hace que sea recomendable escoger las concatenaciones de flujo como variables de estado con el objetivo de simplificar la resolución del sistema de ecuaciones.

Las variables  $f_{0s}$  y  $f_{0r}$  de voltaje del estator y del rotor, han sido incorporadas a la transformación ya que en general tres variables independientes son necesarias. Éstas aparecen cuando existen condiciones desbalanceadas o asimétricas y no influyen en los demás ejes o en otros lugares. Sin embargo, en sistemas que tienen conexión de tres alambres, esto es, sin conexión al neutro, las componentes del eje cero desaparecen incluso para sistemas desbalanceados. Con esta consideración el problema queda reducido a la resolución de un sistema de cuatro ecuaciones diferenciales.

Ahora se verá como el modelo desarrollado puede ser usado para simular una máquina de inducción simétrica de  $P$  polos en el marco de referencia estacionario. Realizando la transformación de los voltajes de fase del estator a voltajes  $qd0$ . Con el eje  $q$  de la referencia estacionaria  $qd$  siempre alineada con el eje de la fase  $a$  del estator y una  $\omega$  igual a cero, se obtiene las siguientes relaciones haciendo  $\theta = 0$  en la matriz de transformación de la ec.2.27:

$$\begin{bmatrix} v_{qs}^s \\ v_{dr}^s \\ v_{0r}^s \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & -\sqrt{3}/2 & \sqrt{3}/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix}$$

$$\begin{aligned}
 v_{qs}^s &= \frac{2v_{as} - v_{bs} - v_{cs}}{3} \\
 v_{ds}^s &= \frac{-v_{bs} + v_{cs}}{\sqrt{3}} \\
 v_{0s}^s &= \frac{v_{as} + v_{bs} + v_{cs}}{3}
 \end{aligned} \tag{2.62}$$

La transformación de los voltajes  $abc$  del rotor en el marco de referencia estacionario  $qd0$  puede hacerse en un solo paso con la matriz de transformación o en dos pasos separados de la siguiente forma: primero se transforma los voltajes de fase del rotor al marco de referencia  $qdr$  fijado en el rotor con el eje  $q$  de dicho marco alineado al eje de la fase  $a$  del rotor. Las ecuaciones resultantes son las siguientes:

$$\begin{aligned}
 v_{qr}^{r'} &= \frac{2v'_{ar} - v'_{br} - v'_{cr}}{3} \\
 v_{dr}^{r'} &= \frac{-v'_{br} + v'_{cr}}{\sqrt{3}} \\
 v_{0r}^{r'} &= \frac{v'_{ar} + v'_{br} + v'_{cr}}{3}
 \end{aligned} \tag{2.63}$$

Transformando al marco de referencia estacionario, resulta:

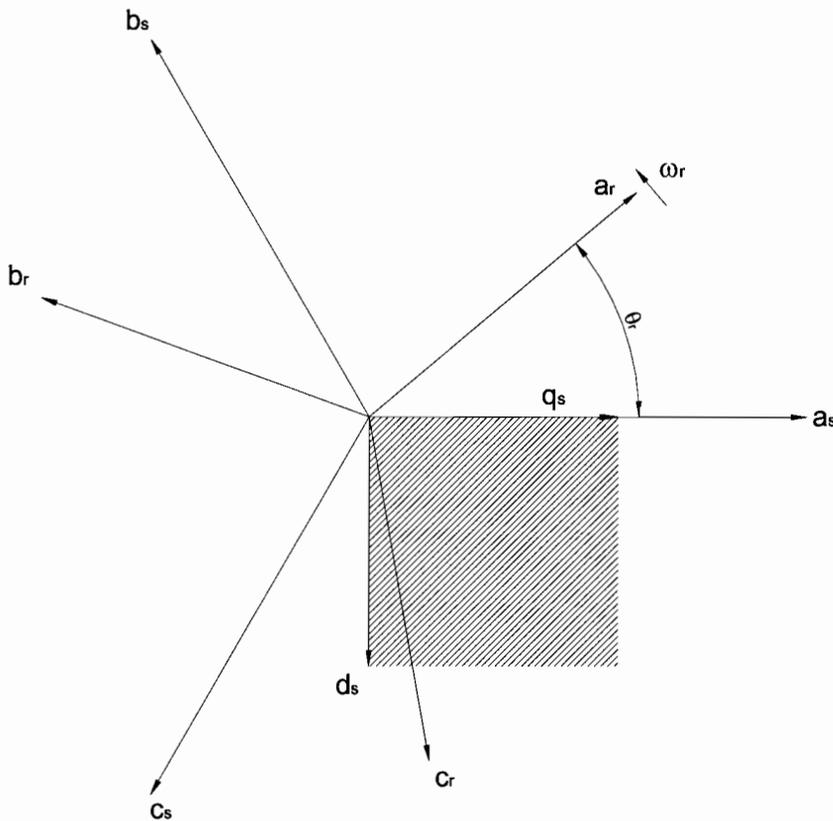
$$f_{qd0}^s = {}^r K^s f_{qdr}^{r'} \tag{2.64}$$

donde  ${}^r K^s$  es la matriz de transformación del marco de referencia rotativo al marco de referencia estacionario

$$\begin{bmatrix} v_{qr}^{s'} \\ v_{dr}^{s'} \\ v_{0r}^{s'} \end{bmatrix} = \begin{bmatrix} \cos(-\theta_r) & -\text{sen}(-\theta_r) & 0 \\ \text{sen}(-\theta_r) & \cos(-\theta_r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{qr}^{r'} \\ v_{dr}^{r'} \\ v_{0r}^{r'} \end{bmatrix}$$

$$\begin{aligned}
 v_{qr}^{rs} &= v_{qr}^{rr} \cos \theta_r(t) + v_{dr}^{rr} \sin \theta_r(t) \\
 v_{dr}^{rs} &= -v_{qr}^{rr} \sin \theta_r(t) + v_{dr}^{rr} \cos \theta_r(t) \\
 \vartheta_r(t) &= \int_0^t \omega_r(t) dt + \theta_r(0)
 \end{aligned}
 \tag{2.65}$$

donde, como se muestra en la figura 2.5,  $\theta_r(t)$  es el ángulo de la fase *a* del rotor con respecto a la del estator,  $\theta_r(0)$  es el valor inicial del mismo y  $\omega_r(t)$  es la velocidad angular instantánea del rotor.



**Fig. 2.5** Ejes de los devanados *abc* y de los devanados *qd* estacionarios

En el presente trabajo se va a utilizar un motor trifásico de inducción conectado en estrella con rotor bobinado, el cual se encuentra disponible en el Laboratorio de Máquinas Eléctricas para la realización de las pruebas, cuyas *características* se mencionará más adelante. Debe considerarse que dado que los terminales de

rotor se encuentran cortocircuitados, los voltajes terminales de éste son cero, así como los voltajes reflejados al estator tanto en variables  $abc$  como en el sistema de referencia  $qd0$ .

Los voltajes  $qd0$  de los terminales del estator y rotor pueden ser usados ahora como entradas para obtener los flujos y las corrientes  $qd0$  en el marco de referencia estacionario partiendo para ello de la ecuación 2.61, y a partir de éstos determinar el torque.

## **CAPÍTULO 3**

# **PROCESAMIENTO DIGITAL DE SEÑALES UTILIZANDO EL DSP TMS320C31**

## **3 PROCESAMIENTO DIGITAL DE SEÑALES UTILIZANDO EL DSP TMS320C31**

### **3.1 PROCESAMIENTO DIGITAL DE SEÑALES**

El rápido desarrollo del procesamiento digital de señales durante los últimos 30 años es el resultado de los avances tecnológicos tanto de los procesadores digitales como en la fabricación de circuitos integrados. Los ordenadores digitales y el hardware asociado hace tres décadas eran relativamente grandes y caros. Como consecuencia, su uso se limitaba a aplicaciones de propósito general en tiempo no real, tanto científicas como comerciales. El gran desarrollo de la tecnología de circuitos integrados, ha estimulado el desarrollo de ordenadores digitales más potentes, pequeños, rápidos y baratos y de hardware digital de propósito general que han hecho posible la construcción de sistemas digitales altamente sofisticados, capaces de realizar funciones y tareas del procesamiento de señal digital que normalmente eran muy difíciles y/o caras con circuitería o sistemas de procesamiento de señales analógicas.

En particular, el procesamiento digital de señal permite operaciones programables. En efecto, por medio de software se pueden modificar fácilmente las funciones de procesado de señal para que sean realizadas por el hardware. Por tanto, el hardware digital y el software asociado proporcionan un mayor grado de flexibilidad en el diseño de sistemas.

Dentro del procesamiento de la señal, la conversión de una señal analógica en digital, obtenida haciendo un muestreo de la señal y cuantificando las muestras, produce una distorsión que impide la reconstrucción de la señal analógica original a partir de las muestras cuantificadas. El control de esta distorsión se logra con la elección apropiada de la velocidad de muestreo y la precisión del proceso de cuantificación. Además, existen efectos debidos a la precisión finita que deben ser considerados en el procesamiento digital de las muestras cuantificadas.

En el presente proyecto se van a tomar las señales de voltaje y velocidad del motor las cuales van a ser procesadas digitalmente utilizando el DSP (Digital Signal Processor) TMS320C31 con el fin de obtener el torque de la máquina trifásica de inducción en tiempo real, para ello es necesario realizar la conversión de las señales análogas de voltaje y velocidad a señales digitales. Por lo tanto, se necesita una interfaz entre la señal analógica y el procesador digital. Esta interfaz se denomina conversor analógico-digital (A/D).

### 3.1.1 ELEMENTOS BÁSICOS DE UN SISTEMA DE PROCESADO DIGITAL DE SEÑALES

Con el propósito de comprender la función de un sistema de procesamiento digital de señales, es importante introducir algunos conceptos.

Una **señal** se define como una cantidad física que varía con el tiempo, el espacio o cualquier otra variable o variables independientes. Matemáticamente, se describe una señal como una función de una o más variables independientes.

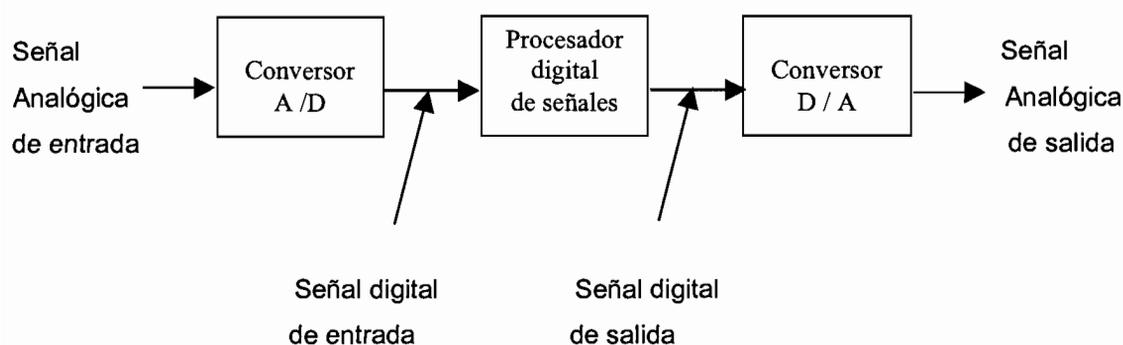
Un **sistema** se puede definir también como un dispositivo físico que realiza una operación sobre una señal. Por ejemplo, un filtro que se usa para reducir el ruido y las interferencias que corrompen la señal conteniendo la información deseada se denomina sistema. En este caso, el filtro realiza algunas operaciones sobre la señal, cuyo efecto es reducir (filtrar) el ruido y la interferencia presentes en la señal deseada.

Cuando una señal pasa a través de un sistema, como en el caso del filtrado, se dice que la señal ha sido procesada. En este caso, el procesamiento de la señal implica la separación de la señal deseada del ruido y la interferencia. En general, el sistema se caracteriza por el tipo de operación que realiza sobre la señal, tales operaciones se denominan habitualmente como **procesado de señal**.

Es conveniente ampliar la definición de sistema para incluir no sólo dispositivos físicos, sino también realizaciones de software de operaciones sobre una señal,

que se pueden especificar a menudo matemáticamente y se implementan mediante un algoritmo que debe ser eficiente, rápido y simple.

Las señales a ser procesadas como en este caso son generalmente analógicas, y son funciones del tiempo. El procesamiento digital de señales proporciona un método alternativo para procesar la señal analógica, como se ilustra en la figura 3.1. Para realizar el procesamiento digitalmente, como se mencionó anteriormente, se necesita una interfaz entre la señal analógica y el procesador digital. Esta interfaz se denomina convertor analógico–digital (A/D). La salida del convertor analógico-digital es una salida adecuada como entrada al procesador digital.



**Fig. 3.1** Diagrama de bloques de un sistema digital de procesamiento de señales

El procesador digital de señales puede ser un gran ordenador digital programable o un pequeño microprocesador programado para realizar las operaciones deseadas sobre la señal de entrada, como es el caso del procesador digital de señales TMS320C31. Los procesadores programables proporcionan la flexibilidad de cambiar las operaciones de procesamiento de señales mediante un cambio del software. En aplicaciones en las que la salida digital se ha de entregar en forma analógica, se debe proporcionar otra interfaz desde el dominio digital al analógico, que se denomina convertor digital-analógico (D/A).

### **3.1.2 PROCESADOR DIGITAL DE SEÑALES**

Los procesadores digitales de señales (DSP's) son básicamente "procesadores numéricos" muy rápidos. Los DSPs se diferencian de los microprocesadores en que los segundos son construidos típicamente para un rango de funciones de propósito general, y normalmente corren grandes bloques de software tales como los sistemas operativos. Los DSPs, en cambio, corren pequeños bloques de software pero procesan gran cantidad de datos. Los microprocesadores no son considerados a menudo para el cómputo en tiempo real pues están en la libertad para mezclar sus cargas de trabajo con su alrededor y para elegir su propia línea de conducta. A pesar de que los microprocesadores tienen algunas capacidades numéricas no se comparan con las de los DSPs que a menudo son usados como un tipo de procesador que acompañado por todo el software necesario, y ensamblado dentro de un equipo está dedicado a un solo grupo de tareas.

Los DSPs pueden clasificarse por su rango dinámico, es decir, la extensión de números desde el menor al mayor número representable, que debe ser procesado en el curso de una aplicación. El DSP toma un cierto rango de valores los cuales describen la forma de onda de una señal, el rango puede conseguir ser incluso más ancho con los cálculos que se realizan, generando números más grandes y más pequeños con la multiplicación y la división. El dispositivo DSP debe tener la capacidad de manejar los datos numéricos tan rápidamente como estos se generen, si no lo hace, los números pueden desbordarse, desviando los resultados del cómputo. La capacidad del procesador es función del número de bits que puede manipular y del tipo de aritmética que realice (punto flotante o punto fijo).

### **3.1.3 CONVERSIÓN ANÁLOGA - DIGITAL**

Antes de trabajar con sistemas DSP es necesario traducir las señales del mundo real dentro de una aproximación digital, esta traducción es lograda por un convertidor análogo-digital (ADC). Las muestras obtenidas por los ADCs son una serie de medidas de voltaje que rastrean la subida y la caída de la señal

analógica, si el ADC ha hecho bien su trabajo da una representación detallada y exacta de la señal. Después de filtrar la señal, el ADC pasa la información de la señal digitalizada a un DSP, el cual hace el proceso principal, eventualmente, cuando el DSP ha terminado sus tareas, el dato digital puede ser devuelto como señal analógica. Existen también aplicaciones que no requieren que la señal de entrada sea analógica pues se puede tener un dato digital desde el inicio.

El procesamiento digital de señales es altamente numérico y muy repetitivo. Los sistemas de DSP pueden trabajar en tiempo real, capturando y procesando la información mientras esta sucede.

#### **3.1.4 VENTAJAS DEL PROCESAMIENTO DIGITAL DE SEÑALES FRENTE AL ANALÓGICO**

Existen muchas razones por las que el procesado digital de una señal analógica puede ser preferible. Primero, un sistema digital programable permite flexibilidad a la hora de reconfigurar las operaciones de procesado digital de señales sin más que cambiar el programa. La reconfiguración de un sistema analógico implica habitualmente el rediseño del hardware, seguido de la comprobación y verificación para ver que opera correctamente.

También desempeña un papel importante al elegir el formato del procesador de señales, la consideración de la precisión. Las tolerancias en los componentes de los circuitos analógicos hacen que para el diseñador del sistema sea extremadamente difícil controlar la precisión de un sistema de procesado analógico de señales. En cambio, un sistema digital permite un mejor control de los requisitos de precisión a pesar de los errores de cuantización y aliasing (algunos autores traducen este término como solapamiento), introducidos por estos sistemas. Tales requisitos, a su vez, resultan en la especificación de requisitos en la precisión del conversor A/D y del procesador digital de señales, en términos de frecuencia de muestreo apropiada, resolución del conversor, longitud de palabra, aritmética de punto flotante frente a punto fijo y factores similares.

El método de procesado digital de señales también posibilita la implementación de algoritmos de procesado de señal más sofisticados. Generalmente es muy difícil realizar operaciones matemáticas precisas sobre señales en formato analógico. Algunas técnicas de control solo pueden ser implementadas a través de sistemas de procesamiento digital, pues en un sistema analógico sería muy difícil su implementación. Un ejemplo de ello es el control adaptivo, en el cual se requiere que la planta se adapte a cada evento o situación diferente, esto se puede realizar fácilmente mediante software, ya que se realizarían variaciones de ciertos parámetros de operaciones y constantes internas al programa sin necesidad de variar el hardware.

Otra ventaja consiste en la perdurabilidad de la información digital, ya que esta puede ser almacenada sin deterioro de la señal, y este almacenamiento enfrenta mejor al tiempo que una señal analógica almacenada.

En algunos casos, la implementación digital del sistema de procesado de señales es más barato que su equivalente analógico. El menor costo se debe a que el hardware digital es más barato, quizás, es resultado de la flexibilidad ante modificaciones que permite la implementación digital.

Como consecuencia de estas ventajas, el procesado digital de señales se ha aplicado a sistemas prácticos que cubren un amplio rango de disciplinas. Sin embargo, la implementación tiene sus limitaciones, una de ellas es la velocidad de operación de los conversores A/D y de los procesadores digitales de señales. En el caso de conversores A/D se debe tener un conversor lo suficientemente rápido *con respecto* a la señal de frecuencia máxima presente para evitar el error de aliasing, lo que implica que para señales con un amplio ancho de banda y por consiguiente un gran contenido de altas frecuencias, sea bastante difícil hacer un muestreo de la señal sin perder una parte de su contenido. Se debe tomar en cuenta que el costo de un conversor crece con la velocidad del mismo

En conclusión, las ventajas de un sistema de procesamiento digital superan a las desventajas por lo que a medida que la tecnología avanza el uso de sistemas digitales se incrementa.

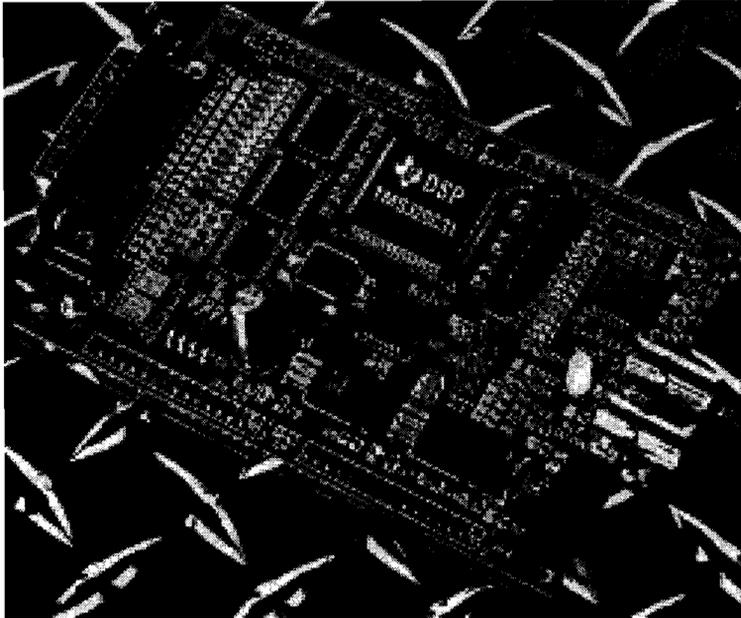
Para el desarrollo del presente proyecto se utiliza un módulo de evaluación de la Texas Instruments que contiene un circuito de interfase análogo, el mismo que posee un conversor análogo digital de 14 bits , utilizado para digitalizar las señales de voltaje y velocidad para que el DSP incluido en la tarjeta pueda realizar las operaciones matemáticas necesarias con dichas señales con el objetivo de obtener el torque de la máquina trifásica de inducción seleccionada.

A continuación se pretende dar una descripción general del módulo de evaluación que contiene el procesador digital de señales y del circuito de interfase análogo utilizados en el presente proyecto.

## **3.2 MÓDULO DE EVALUACIÓN DSK STARTER KIT**

### **3.2.1 VISIÓN GENERAL Y CARACTERÍSTICAS DEL DSK**

El DSK (DSP Starter Kit) es una plataforma de desarrollo de bajo costo, fácil de usar, expansible y de alto rendimiento que permite experimentar y desarrollar aplicaciones de procesamiento de señales con el DSP TMS320C31. El DSK contiene el DSP TMS320C31 en la tarjeta con el fin de permitir la verificación del código del mismo a plena velocidad. El DSK también proporciona libertad para desarrollar software propio en un PC host, descargar el software en el DSK, y correr software en la tarjeta. Además, el DSK permite la expansión del sistema mediante la interfaz con nuevas tarjetas hijas. El DSK se interconecta a un PC host a través del puerto paralelo. El depurador suministrado es orientado a Windows, simplificando el desarrollo del código y las aptitudes de depuración.



**Fig. 3.2** Plataforma de desarrollo DSP Starter Kit

### 3.2.1.1 Requerimientos adicionales

Los requerimientos adicionales para empezar a usar esta plataforma de desarrollo son los siguientes:

- IBM PC/AT o PC compatible con un mínimo de 640 Kb de memoria, un sistema de disco duro, un floppy-disk drive de 1.44 Mb, un puerto paralelo para comunicación
- Monitor a color
- MS-DOS, Windows u OS/2
- Una fuente de poder UL Clase II con conector jack 2.1 mm que provea al menos 500-1500 mA con 7-12 voltios DC o 6-9 voltios AC

### 3.2.1.2 Características principales del DSK

Esta sección detalla las características de la tarjeta de desarrollo DSK Starter Kit TMS320C31.

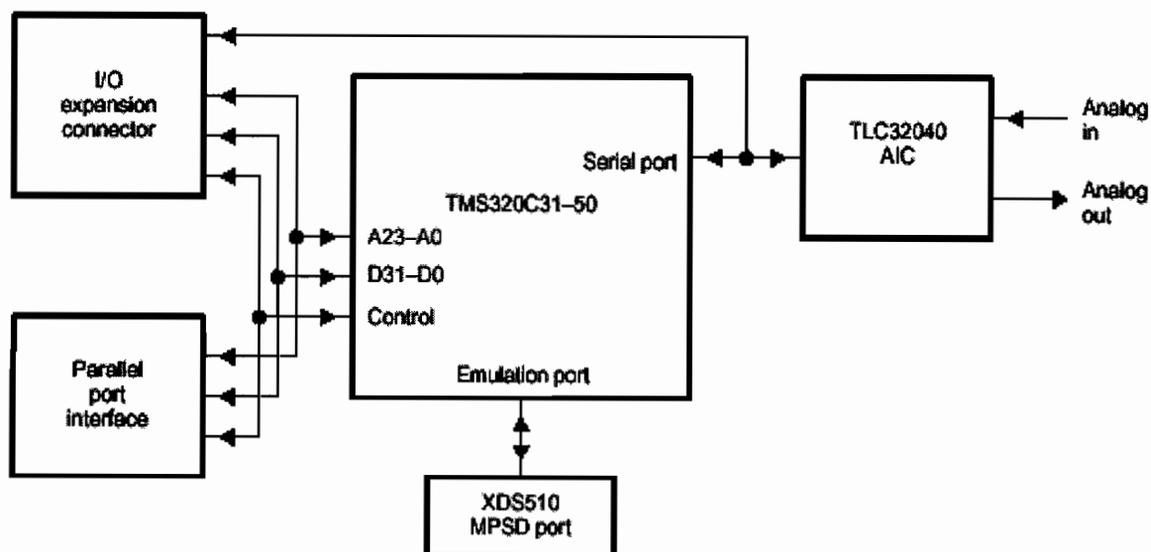
- DSP (Procesador digital de señales) TMS320C31 en punto flotante de fabricación estándar.
- Tiempo de ciclo de instrucción 40-ns.
- Circuito de interfase análogo TLC32040
- Interfase estándar de puerto paralelo de impresora con cable incluido, el cual se conecta al PC host y permite al TMS320C31 comunicarse con programas del computador.
- Depurador
- Ensamblador
- Adquisición de datos analógicos mediante el circuito de interfase análogo AIC (Analog Interface Circuit) TLC32040:
  - Conversor análogo-digital (ADC) y conversor digital-análogo (DAC) de velocidad variable con rango dinámico de 14-bits y 20.000 muestras por segundo.
  - Filtro de reconstrucción de salida pasa bajos y filtro de entrada anti-aliasing de capacitor conmutable.
- Acceso al puerto serial del DSP 'C31 a través de bloque de jumpers.
- Conectores plug RCA estándares para entrada y salida análoga que proveen una conexión directa a un micrófono y parlante.
- Soporte para el emulador XDS510.
- Bus de expansión I/O (entradas/salidas) para tarjetas-hijas del DSK.
- Soporta fuentes de poder de 6-9 V AC o 7-12 V DC
- Led tricolor indicador
- Aplicaciones
  - Audio
  - Control
  - Educación
  - Evaluación
  - Filtro
  - Instrumentación
  - Análisis de espectro
  - Telecomunicaciones
  - Voz

### 3.2.1.3 Hardware y software del DSK

La figura 3.3 representa el diagrama de bloques del hardware del DSK. Los componentes básicos son el DSP TMS320C31, el AIC TLC43040, los conectores de expansión, el reloj de sistema, interfase con el puerto paralelo de impresora, y un LED tricolor. El puerto paralelo de impresora conecta el DSK a un PC host y permite al TMS320C31 comunicarse con programas del PC.

Todas las señales para el 'C31 están encaminadas hacia los conectores de expansión. Los conectores de expansión incluyen 4 headers de 32 pines, un bloque jumper de 11 pines, y un header XDS510 de 12 pines.

El AIC TLC32040 se interconecta al puerto serial del TMS320C31. Un bloque jumper permite eliminar esta conexión para encaminar el puerto serial a una tarjeta-hija del DSK. Los dos conectores RCA proveen una entrada análoga y una salida análoga en la tarjeta.



**Fig. 3.3** Diagrama de bloques del DSK (DSP Starter Kit)

A continuación se describe las funciones básicas de los componentes del DSK:

**Conectores de expansión** Los cuatro headers de 32 pines permiten desarrollar tarjetas de ampliación que pueden interconectarse directamente con todas las señales del 'C31.

**Bloque jumper** Un bloque jumper de 11 pines conecta el puerto serial del 'C31 al AIC TLC32040. La eliminación de los jumpers desconecta el AIC del puerto serial del 'C31, a fin de que una tarjeta-hija pueda usar las señales del puerto serial.

**Interfaz lógica** La interfaz lógica del host consiste de un arreglo lógico programable (PAL) 22V10Z y dos transmisores-receptores de bus octal de alta velocidad con salidas de tres estados (74ACT245). Estos dispositivos interconectan el 'C31 con el puerto paralelo de impresora del host. Esta interfaz lógica soporta modos de datos de 8-bits bidireccionales o 4-bits unidireccionales del PC host.

**Oscilador** El oscilador de 50 Mhz maneja la entrada de reloj del 'C31. El valor del reloj interno del 'C31 es dividido por 1 (misma frecuencia).

**Conector de puerto paralelo** El conector DB25 de 25 pines se conecta directamente al puerto paralelo del host.

**Jacks RCA** Los jacks RCA proporcionan una entrada o salida análoga están encaminados a los pines I/O del AIC.

**Fusibles restaurables** Los fusibles restaurables interrumpen el flujo de corriente excesiva. Los fusibles se restablecen una vez que se han enfriado y la condición de falla es corregida. Los fusibles no requieren restablecimiento o reemplazo manual.

**AIC TLC32040** El circuito de interfase análogo provee el acceso del 'C31 al mundo análogo. El AIC hace un muestreo de datos análogos y los convierte en datos digitales para ser analizados por el 'C31. El 'C31 opera en estos datos digitales y retorna los datos digitales “transformados” al AIC para que los convierta en una señal análoga.

**TMS320C31** Es el procesador digital de señal de 32 bits y punto flotante. Se desarrolla el código de aplicación y se lo carga en la memoria del 'C31. Este código puede ser ejecutado, paso a paso, y observado en el depurador.

**Reguladores de voltaje** El DSK usa una fuente de poder de 7-12 Vdc o 6-9 Vac. La fuente de voltaje de 7-12 Vdc se regula a 5 voltios por el LM7805. Ésta también es convertida a -5 voltios por el circuito capacitivo conmutable LT1054, y luego es regulada por el LM7905. La fuente de 6-9 Vac es rectificadas (rectificación de onda completa) y regulada por el LM7805 y el LM7905 a +5 V y -5 V, respectivamente. Las fuentes de +5 V y -5 V son usadas para alimentar toda la circuitería del DSK. El AIC TLC32040 requiere una fuente de poder negativa de -5 V.

**Puerto Emulador XDS** Un header de 11 pines conecta el emulador XDS510 al 'C31. El emulador permite actualizarse al depurador XDS para depurar el código de aplicación mientras se usa el DSK como la tarjeta del XDS.

Ver el Anexo A, *Dimensiones de la tarjeta DSK y Diagramas Esquemáticos*, para una aclaración de los componentes del DSK.

El depurador y el ensamblador del DSK son interfases de software que ayudan a desarrollar, probar y refinar los programas en lenguaje ensamblador del DSK.

La figura 3.4 ilustra el desarrollo de flujo del código.

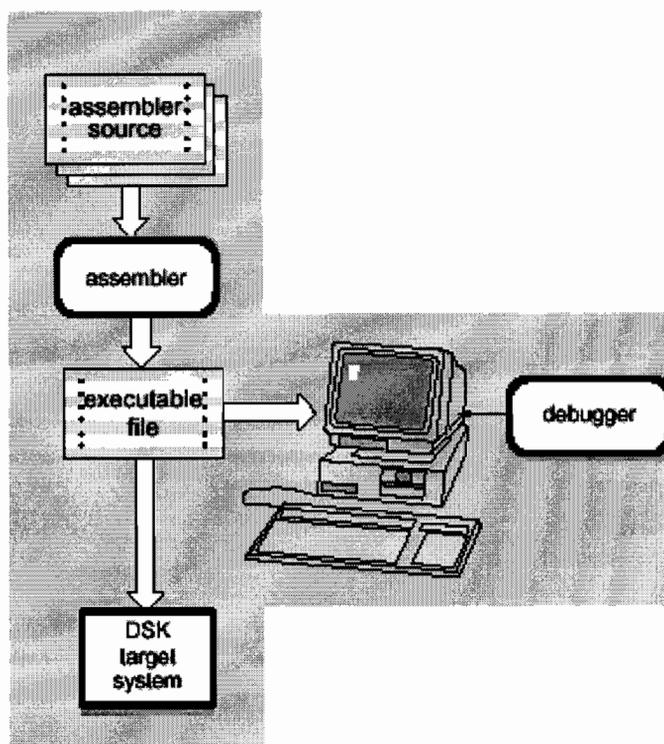


Fig. 3.4 Desarrollo de flujo del software del DSK

### Ensamblador

El ensamblador traduce los archivos fuente en lenguaje ensamblador a archivos objeto en lenguaje de máquina. Solo las características más esenciales del ensamblador son incorporadas. Los archivos objeto ejecutables creados por las herramientas del lenguaje ensamblador puede ser cargados y ejecutados en el DSK.

### Depurador

El propósito principal del proceso de desarrollo es producir un módulo que pueda ser ejecutado en el sistema de la tarjeta DSK. Se puede usar el depurador para refinar y corregir el código.

### 3.2.2 VISION FUNCIONAL DEL DSK

El hardware y el software del DSK TMS320C3x trabajan juntos para crear una plataforma de desarrollo de bajo costo que permite desarrollar aplicaciones de procesamiento de señal en tiempo real.

#### 3.2.2.1 Interfaz de Hardware del DSK

El 'C3x DSK arranca respondiendo a un comando de reset del PC host y cargando un programa inicial de comunicaciones denominado kernel. El kernel de comunicaciones provee las entradas y salidas necesarias para interconectar la tarjeta DSK y el sistema host. Las comunicaciones con el host ocurren a través del bus paralelo del 'C31, mientras las entradas y salidas análogas son manejadas por el circuito de interfase análogo (AIC) TLC32040 y son enviadas al puerto serial del 'C31.

La interfase host consiste de tres dispositivos:

- Un arreglo lógico programable (TICPAL22V110Z)
- Dos transmisores receptores de bus octal de alta velocidad con salidas de tres estados (74ACT245)

La figura 3.5 muestra un diagrama circuital de alto nivel del DSK del C3x.

El arreglo lógico programable (PAL) determina cuándo el 'C31 está accediendo a la interfase del host usando las señales  $\overline{STROBE}$  A23, A22, A21 y A20 para decodificar la dirección del 'C31.

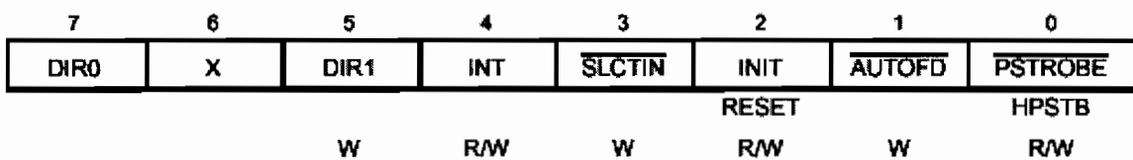
El PAL provee una entrada (TRI) que desconecta la interfase del host poniendo en el tercer estado a las señales  $\overline{INT2}$  y  $\overline{READY}$ . El PAL provee cinco salidas de direcciones decodificadas:  $\overline{USER\_IOR}$ ,  $\overline{USER\_IOW}$ ,  $\overline{USER\_IO}$ ,  $\overline{USER\_RAM}$ ,  $\overline{USER\_BOOT}$ ; y tres salidas:  $\overline{READY}$ ,  $\overline{INT2}$ , y  $\overline{EN}$ . Cuando la señal DEMO es



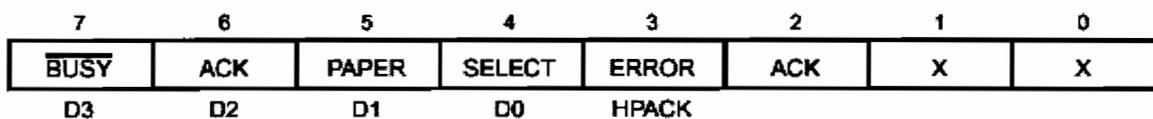
Los transmisores receptores almacenan temporalmente datos entre el puerto paralelo de impresora del PC y el bus paralelo del 'C31. La interfase del host soporta dos tipos de transferencias:

- El modo bidireccional de 8 bits permite rápidas transferencias en puertos paralelos que soportan transferencias bidireccionales.
- Los puertos unidireccionales soportan una transferencia de 8 bits del host al 'C31 mientras soporta transferencias de 4 bits del 'C31 hacia el host.

El PC host se comunica con el 'C31 a través del puerto paralelo, y manipula las señales del puerto paralelo escribiendo y leyendo los registros de estatus y de control. Las figuras 3.6 y 3.7 muestran los campos de bits usados por el software del DSK. (Las etiquetas bajo los nombres de las señales del puerto paralelo se refieren a los nombres como han sido usados por la tarjeta del DSK como se muestra en la figura 3.5)



**Fig. 3.6** Registro de control del puerto paralelo (0x37A)



**Fig. 3.7** Registro de estatus del puerto paralelo (0x379)

El host inicializa el 'C31 pulsando la señal  $\overline{INIT}$  (escribe un 0 seguido por un 1 en el campo del bit  $\overline{INIT}$  del registro de control del puerto paralelo). Esta señal

restablece el 'C31 y activa el modo de carga inicial (modo "bootload"). El ordenador entonces descarga el programa o el kernel de comunicaciones al 'C31. El puerto paralelo es ubicado en el mapa de memoria del 'C31 en el rango de direcciones 0xFFFF000 – 0xFFFFFFFF, como se muestra en la figura 3.6.

El ordenador envía datos al 'C31 de la siguiente manera:

1. El ordenador escribe el byte a ser transmitido al área I/O de las líneas de datos del puerto paralelo del ordenador (dirección I/O 0x378 para LPT1).
2. El ordenador pone la señal HPSTB en bajo y espera por un reconocimiento ("acknowledgement"). La señal HPSTB interrumpe al 'C31 pulsando la señal  $\overline{INT2}$ , indicando que el host está solicitando la transferencia de un paquete de información. La señal  $\overline{INT2}$  es necesitada solo para el requerimiento de transferencia de paquete inicial y es ignorada durante las peticiones de paquetes de información subsiguientes.
3. El 'C31 inicia un acceso de lectura en estado de espera a la localidad 0xFFFF000. El PAL decodifica esta dirección cuando la interfase del host activa la señal HPACK, pone la señal ERROR del host en bajo, y la señal  $\overline{READY}$  del 'C31 en alto. Este previene al 'C31 de completar su acceso de lectura. El host usa la señal ERROR (HPACK) para reconocer que el 'C31 está bloqueado esperando recibir datos.
4. El ordenador pone la señal HPSTB en alto, indicando al 'C31 que los datos están listos. El PAL detecta el estado alto de HPSTB y pone la señal  $\overline{READY}$  del 'C31 en bajo, liberando el bus de acceso, y concluyendo el ciclo de lectura del 'C31.
5. El proceso se repite hasta que los cuatro bytes son transferidos ( el menos significativo primero). En cada transferencia, el 'C31 pone los bytes juntos para formar una palabra de 32 bits.

El ordenador recibe datos del 'C31 de manera similar:

1. El ordenador espera por la señal HPACK indicando que el 'C31 entiende el requerimiento de una transferencia.
2. El 'C31 inicia un acceso de escritura en estado de espera a la localidad 0xFFFF000. El PAL decodifica esta dirección como la señal HPACK, pone la señal ERROR del host en bajo, y pone la señal  $\overline{READY}$  del 'C31 en alto. Esto evita que el 'C31 complete el acceso de escritura. El host usa la señal de ERROR para reconocer que el 'C31 ya está enviando datos.
3. Cuando el ordenador recibe la señal HPIA, éste pone la señal  $\overline{PSTROBE}$  en bajo y lee un byte o un nibble (4 bits), dependiendo si está presente un puerto paralelo bidireccional en el host.
4. El ordenador pone la señal HPSTB en alto, indicando al 'C31 que los datos fueron leídos. El PAL detecta que la señal HPSTB se ha puesto en alto y pone la señal  $\overline{READY}$  del 'C31 en bajo, concluyendo el ciclo de escritura del 'C31. Esto completa el ciclo de escritura del 'C31.
5. Este proceso se repite hasta que los cuatro bytes son transferidos ( el byte menos significativo primero). Durante cada transferencia, el ordenador ubica los bytes juntos para formar una palabra de 32 bits.

Debido a que las comunicaciones con el ordenador ocurren a través del bus paralelo del 'C31, el PAL decodifica la dirección del 'C31 para determinar cuándo éste está accediendo a la interfase del host de acuerdo al mapa de memoria mostrado en la figura 3.8.

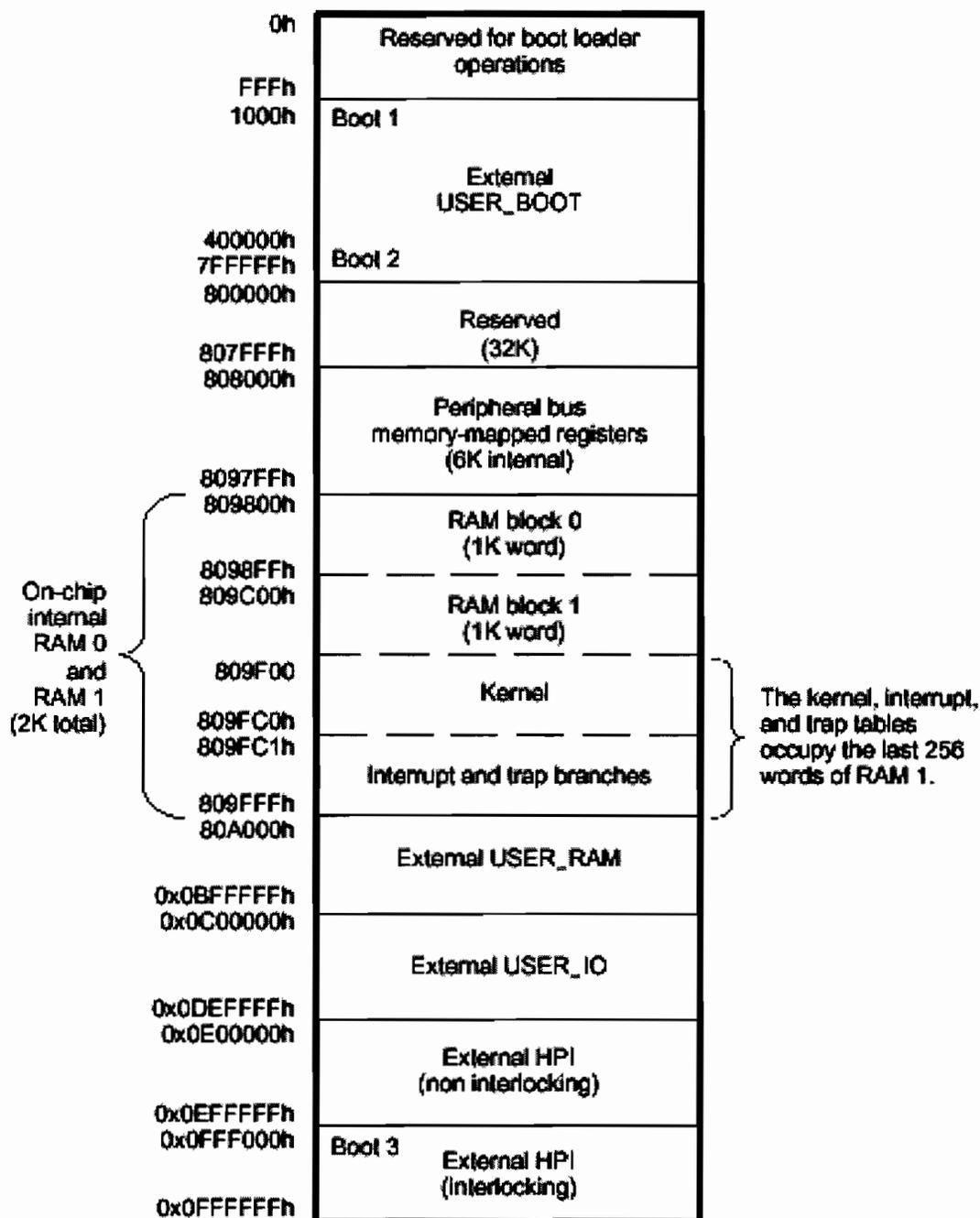


Fig. 3.8 Mapa de memoria del DSK

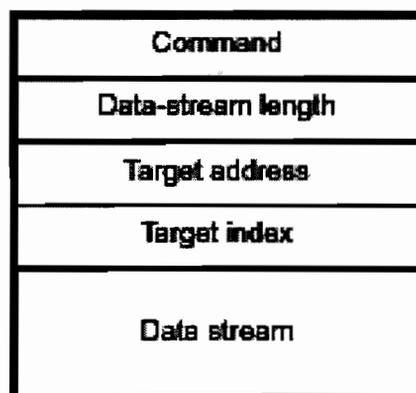
### 3.2.2.2 Kernel de Comunicaciones del DSK

Luego del reset, el ordenador descarga un kernel de comunicaciones al 'C31 usando el cargador inicial, "bootloader". Este kernel de comunicaciones

proporciona un set de rutinas de bajo nivel que permiten al host y al 'C31 intercambiar información y ejecutar las funciones de depuración.

El host y el 'C31 se comunican intercambiando paquetes de datos. La figura 3.9 presenta la estructura de los paquetes de datos. Los encabezamientos de los paquetes de datos ( sección sombreada) típicamente consisten de cuatro campos: comando, largo de la corriente de datos, dirección del objetivo, e índice del objetivo. Este encabezado está seguido por la corriente de datos. Los campos del encabezamiento son descritos como sigue:

- **Comando** dirige la manipulación de los paquetes.
- **Largo de la corriente de datos** indica el largo de los datos en la corriente de datos
- **Dirección del objetivo** apunta a la localidad de memoria donde los datos son leídos o escritos.
- **Índice del objetivo** pos-incrementa el valor de la dirección



**Fig. 3.9** Estructura del paquete de datos

Cuando el 'C31 recibe una interrupción del ordenador ( $\overline{INT2}$ ), el 'C31 guarda el estado actual del CPU y en ese momento recibe un paquete. Una vez que el 'C31 recibe el paquete, el kernel de comunicaciones analiza el comando en el

encabezamiento para controlar el manejo del paquete. El comando proporciona las rutinas de bajo nivel necesarias para comunicarse con el host y depurar el sistema.

### 3.2.2.3 Software del Host

El software del DSK incluye algunos archivos de código fuente que manipulan el puerto paralelo y realizan las funciones necesarias para inicializar y comunicarse con el 'C31. Los comandos en cada uno de los archivos de código fuente se resumen en los siguientes apartados. Los archivos fuente entre los que típicamente se establece una comunicación o son enlazados incluyen:

<b>driver.cpp</b>	incluye rutinas que controlan la interfaz del puerto paralelo del host. Éstas manipulan la circuitería de la interfaz de hardware del host para restablecer, enviar, y recibir datos a través de los puertos paralelos de impresora unidireccionales y bidireccionales.
<b>target.cpp</b>	incluye las rutinas de bajo nivel que manipulan las transmisiones de datos en paquetes que son reconocidos por el kernel de comunicaciones del 'C31.
<b>object.cpp</b>	usa las rutinas de los archivos fuente driver y target para inicializar y descargar programas al 'C31. Usando las rutinas de bajo nivel del archivo driver, el software del DSK proporciona algunas rutinas de alto nivel que permiten cargar programas o datos desde los archivos dsk3a o COFF (Common Object File Format), que mueven los datos binarios desde el host hacia el DSK, y que inicializan el sistema del DSK. Estas rutinas asumen un kernel de comunicaciones activo residente en el 'C31 para enviar y recibir paquetes de datos.
<b>dsk_coff.cpp</b>	incluye el cargador DSK y archivo COFF y utilitarios
<b>errmsg.cpp</b>	incluye caracteres de texto
<b>symbols.cpp</b>	incluye rutinas de soporte de tabla de símbolos

**helpmsg.cpp** incluye mensaje de ayuda en la línea de comando.

El software del DSK también incluye un ensamblador y un depurador. Estos serán descritos en el capítulo siguiente.

### **3.2.3 DESCRIPCIÓN GENERAL DEL DSP TMS320C31**

La generación de procesadores digitales de señal (DSPs) TMS320C3X está formada por dispositivos de punto flotante de alto rendimiento, de tecnología CMOS de 32-bits. El tiempo del ciclo de instrucción de 40-ns del TMS320C31 - 50 le permite ejecutar operaciones a una velocidad de ejecución mayor a 60 millones de instrucciones de punto flotante por segundo (MFLOPS) y 30 millones de instrucciones por segundo (MIPS). Anteriormente este desempeño estaba disponible solo en una supercomputadora. El rendimiento de esta generación es adicionalmente mejorado a través de sus extensas memorias, controlador de acceso directo a memoria concurrente (DMA), y un puerto externo de interfase.

El TMS320C31 puede realizar operaciones paralelas y aritméticas-lógicas (ALU) en datos enteros o de punto flotante en un solo ciclo. El procesador también posee un archivo de registros de propósito general, una aplicación caché, unidades de registros aritméticos auxiliares dedicados (ARAU), accesos de memoria duales internos, un canal DMA, y corto tiempo de ciclo de máquina. Un alto desempeño y facilidad de uso son producto de estas características.

Aplicaciones de propósito general son enormemente mejoradas por el amplio espacio de direccionamiento, interfase multiprocesador, estados de espera generados interna y externamente, un puerto externo para interfase, dos temporizadores, un puerto serial, una estructura de interrupción múltiple, soporta una amplia variedad de aplicaciones desde procesador host hasta coprocesador dedicado.

Un lenguaje de alto nivel es implementado más fácilmente a través de una arquitectura basada en registros, modos de direccionamiento poderosos, set de instrucciones flexible, y aritmética de punto flotante bien sustentada.

### 3.2.3.1 Características principales

- Procesador digital de señales en punto flotante de alto rendimiento.
- TMS320C31-50 (5V)  
Tiempo de ciclo de instrucción 40-ns  
275 MOPS, 50 MFLOPS, 25 MIPS
- CPU de alto desempeño de 32-bits.
- Operaciones de enteros de 16-/32-bits y de punto flotante de 32-/40-bits.
- Palabra de instrucción de 32-bits, direcciones de 24-bits.
- Dos bloques de RAM de 1K x 32-bits
- Periféricos en el mapa de memoria
  - Un puerto serial
  - Dos temporizadores de 32-bits
  - Coprocesador de acceso directo a memoria (DMA).
- Fabricado utilizando tecnología CMOS por Texas Instruments (TI™)
- Paquete cuadrangular, plano, liso de 132-pines
- 8 registros de precisión-extendida
- 2 generadores de direcciones con 8 registros auxiliares y dos unidades auxiliares aritméticas de registros (ARAUs)
- Dos modos de baja potencia
- Instrucciones de 2 y 3 operandos
- Unidad Aritmética / Lógica Paralela y Multiplicador con ejecución en 1 ciclo de máquina.
- Capacidad de Repetición de Bloque
- Llamadas y Retornos a subrutinas condicionales
- Instrucciones Ínter bloqueadas para multiprocesamiento

Ver el Anexo B, *Hojas de Datos del TMS320C31*, para una mejor explicación de las asignaciones y funciones de sus pines, las condiciones de operación, e información de los distintos parámetros del mismo.

### **3.2.4 ARQUITECTURA DEL TMS320C31**

La arquitectura del TMS320C31 responde a demandas de sistema que están basadas en algoritmos aritméticos sofisticados y que enfatizan tanto en soluciones de hardware como en soluciones de software. Un alto rendimiento es obtenido a través de la precisión y un amplio rango dinámico de las unidades en punto flotante, amplia memoria, un alto grado de paralelismo, y el controlador de acceso directo a memoria (DMA).

La figura 3.10 representa un diagrama de bloques de la arquitectura del TMS320C31.

#### **3.2.4.1 Unidad Central de Proceso (CPU)**

El TMS320C31 tiene una unidad central de proceso (CPU) cuya arquitectura está basada en registros, el CPU consiste de los siguientes componentes:

- Multiplicador de enteros y punto flotante
- Unidad Aritmética lógica (ALU) para realizar operaciones aritméticas / lógicas de enteros y punto flotante
- Un desplazador (shifter) de 32-bits que mueve o traslada los bits de una posición a otra.
- Buses internos (CPU1 / CPU2 y REG1 / REG2)
- Unidades aritméticas de registros auxiliares (ARAUs)
- Archivo de registro del CPU

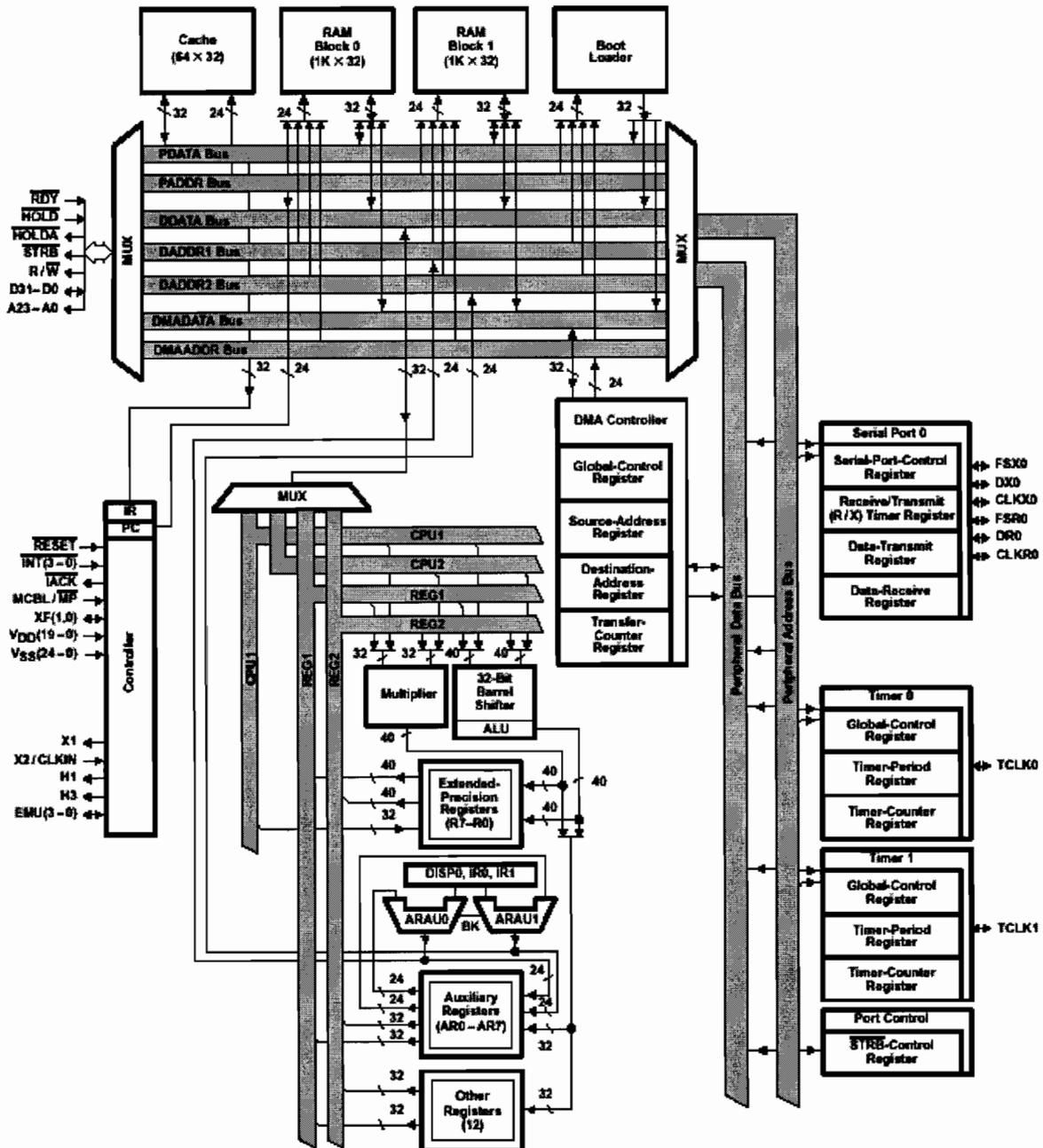


Fig. 3.10 Diagrama de Bloques del TMS320C31

La figura 3.11 muestra los distintos componentes del CPU que serán descritos a continuación.

#### 3.2.4.1.1 *Multiplicador*

El multiplicador realiza multiplicaciones de un solo ciclo en enteros de 24-bits y valores en punto flotante de 32-bits. La implementación de aritmética de punto flotante permite operaciones en punto flotante a velocidades de punto fijo mediante un ciclo de instrucción de 40-ns y un alto grado de paralelismo. Para obtener una mayor cantidad de datos transmitidos por segundo se puede usar las instrucciones paralelas para realizar una multiplicación y una operación ALU en un solo ciclo de máquina.

Cuando el multiplicador realiza una operación en punto flotante, las entradas son números en punto flotante de 32-bits, y el resultado es un número en punto flotante de 40-bits. Cuando el multiplicador realiza una operación de enteros, los datos de entrada son de 24-bits y produce un resultado de 32-bits.

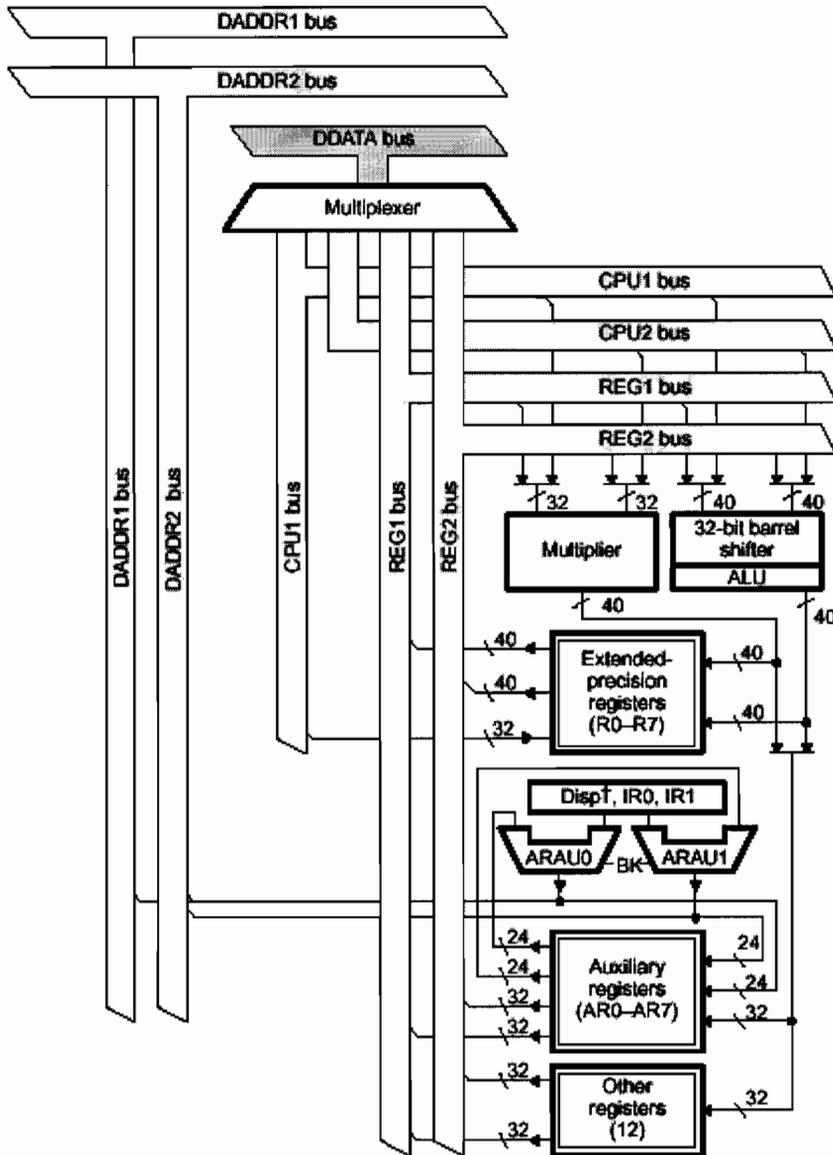
#### 3.2.4.1.2 *Unidad Aritmética Lógica (ALU)*

La ALU realiza operaciones de un ciclo de máquina de datos enteros de 32-bits, lógicas de 32-bits y en punto flotante de 40-bits. Los resultados de la ALU son mantenidos siempre en formatos de enteros de 32-bits y punto flotante de 40-bits. El desplazador (shifter) es usado para mover más de 32-bits a la izquierda o a la derecha en un solo ciclo de máquina.

Los buses internos, CPU1 / CPU2 y REG1 / REG2, transportan dos operandos de la memoria y dos operandos del archivo de registro, esto permite multiplicaciones paralelas y sumas y restas en 4 operandos enteros o en punto flotante en un solo ciclo de máquina.

#### 3.2.4.1.3 *Unidades Aritméticas Auxiliares de Registro*

Estas dos unidades (ARAU0 y ARAU1) pueden generar dos direcciones en un solo ciclo. Las ARAUs operan en paralelo con el multiplicador y la ALU. Ellas soportan direccionamiento con desplazamientos, registros índices (IR0 e IR1), y direccionamiento circular y de bit-invertido.



**Fig. 3.11** Unidad Central de Proceso (CPU)

#### 3.2.4.1.4 Archivo de Registro del CPU

El TMS320C31 provee de 28 registros en un archivo de registro que está estrechamente acoplado al CPU. Todos estos registros pueden ser operados por el multiplicador y la ALU y pueden ser usados como registros de propósito general. Sin embargo, los registros también tienen algunas funciones especiales. Por ejemplo, los ocho registros de precisión extendida son convenientes para mantener resultados en punto flotante. Los ocho registros auxiliares soportan una

variedad de modos de direccionamiento indirecto y pueden ser usados como registros de propósito general lógicos y de enteros de 32-bits. Los registros restantes proporcionan funciones de sistema como direccionamiento, manejo de la pila (stack), estado del procesador, interrupciones, y repetición de bloque.

Los nombres de los registros y sus funciones se encuentran listados a continuación en la Tabla 3.1.

<b>Nombre del Registro</b>	<b>Función Asignada</b>
R0	Registro de precisión extendida 0
R1	Registro de precisión extendida 1
R2	Registro de precisión extendida 2
R3	Registro de precisión extendida 3
R4	Registro de precisión extendida 4
R5	Registro de precisión extendida 5
R6	Registro de precisión extendida 6
R7	Registro de precisión extendida 7
AR0	Registro auxiliar 0
AR1	Registro auxiliar 1
AR2	Registro auxiliar 2
AR3	Registro auxiliar 3
AR4	Registro auxiliar 4
AR5	Registro auxiliar 5
AR6	Registro auxiliar 6
AR7	Registro auxiliar 7
DP	Puntero de la página de datos
IR0	Registro índice 0
IR1	Registro índice 1
BK	Tamaño de bloque
SP	Puntero del stack
ST	Registro de estatus
IE	Habilitación de interrupciones CPU/DMA
IF	Banderas de interrupción del CPU
IOF	Banderas I/O (entrada / salida)
RS	Dirección de inicio de repetición
RE	Dirección de fin de repetición
RC	Contador de repetición

**Tabla 3.1** Registros del CPU

### 3.2.5 ORGANIZACIÓN DE MEMORIA DEL TMS320C31

El espacio total de memoria del TMS320C31 es 16M (millones) de palabras de 32-bits. Espacio de programa, datos e I / O (entrada / salida) están contenidos en estos 16M de palabras, esto permite que tablas, coeficientes, código de programa, o datos sean almacenados sea en RAM o en ROM. De este modo, el uso de la memoria es maximizado y el espacio de memoria asignado como se desea.

### **3.2.5.1 RAM, ROM y Caché**

La figura 3.12 muestra como está organizada la memoria en el TMS320C31. Los bloques de RAM 0 y 1 son cada uno de 1K x 32 bits. El bloque ROM está disponible solo en el TMS320C30. Cada bloque de RAM es capaz de soportar dos accesos del CPU en un solo ciclo. Los buses separados de programa, datos y DMA dan cabida a búsquedas de programa paralelas, lecturas y escrituras de datos, y operaciones DMA. Por ejemplo: El CPU puede acceder a dos valores en un bloque de RAM y realizar una búsqueda externa de programa en paralelo con el DMA cargando otro bloque de RAM, todo dentro de un ciclo de máquina.

Una caché de 64 x 32-bits es proporcionada para almacenar secciones de código que se repiten con frecuencia, de esta manera se reduce el número de accesos necesarios. Esto permite al código ser almacenado en memorias más lentas y de menor costo. Los buses externos también son liberados para ser usados por el DMA, búsquedas de memoria externas u otros dispositivos en el sistema.

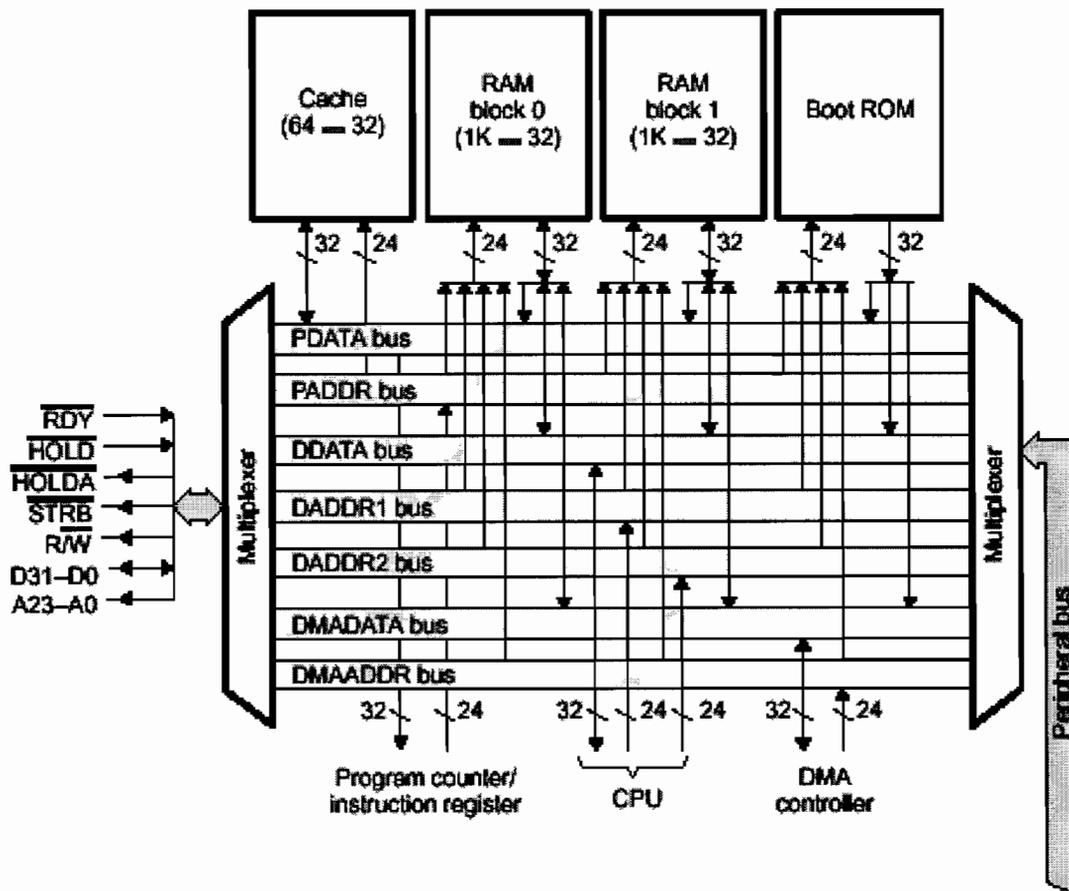


Fig. 3.12 Organización de memoria

### 3.2.5.2 Mapas de memoria

El mapa de memoria depende de si el procesador está corriendo en modo de microprocesador ( $MCBL/\overline{MP} = 0$ ) o en modo de microcomputador ( $MCBL/\overline{MP} = 1$ ). Los mapas de memoria para estos modos son similares. Las localidades 800000h–807FFFh están reservadas. Todos los registros de buses periféricos en el mapa de memoria están en las localidades 808000h–8097FFFh. En ambos modos, el bloque 0 de RAM está ubicado en las direcciones 809800h–809BFFFh, y el bloque 1 de RAM está localizado en las direcciones 809C00h–809FFFh. Las localidades 80A000h–0FFFFFFFh son accedidas por el puerto de memoria externa ( $\overline{STRB}$  activo).

En modo de microprocesador, el cargador de inicio (boot loader) no está en el mapa de memoria. Las localidades 0h–03Fh consisten en el vector de interrupción, y localidades reservadas, todas ellas son accedidas por el puerto de memoria externa ( $\overline{STRB}$  activo). Las localidades 040h–7FFFFFFh también son accedidas por el puerto de memoria externa.

En modo de microcomputadora, el cargador de inicio (boot loader) está en las localidades 0h–0FFFh. Las últimas 63 palabras (809Fc1h a 809FFFh) de bloque interno de la RAM1 son usadas para saltos de interrupción. Las localidades 1000h–7FFFFFFh son accedidas por el puerto de memoria externa ( $\overline{STRB}$  activo).

La figura 3.13 muestra los mapas de memoria del TMS320C31-50

### 3.2.5.3 Modos de direccionamiento de la memoria

El TMS320C31 soporta un set básico de instrucciones de propósito general así como instrucciones aritméticas intensivas que son apropiadas particularmente para procesamiento digital de señales y otras aplicaciones numéricas intensivas.

Se provee de cinco modos de direccionamiento en el TMS320C31. Seis tipos de direccionamiento pueden ser usados dentro de cada grupo.

- Modos de direccionamiento generales
- Modos de direccionamiento de tres operandos
- Modos de direccionamiento paralelos
- Modo de direccionamiento inmediato
- Modos de direccionamiento de saltos condicionales

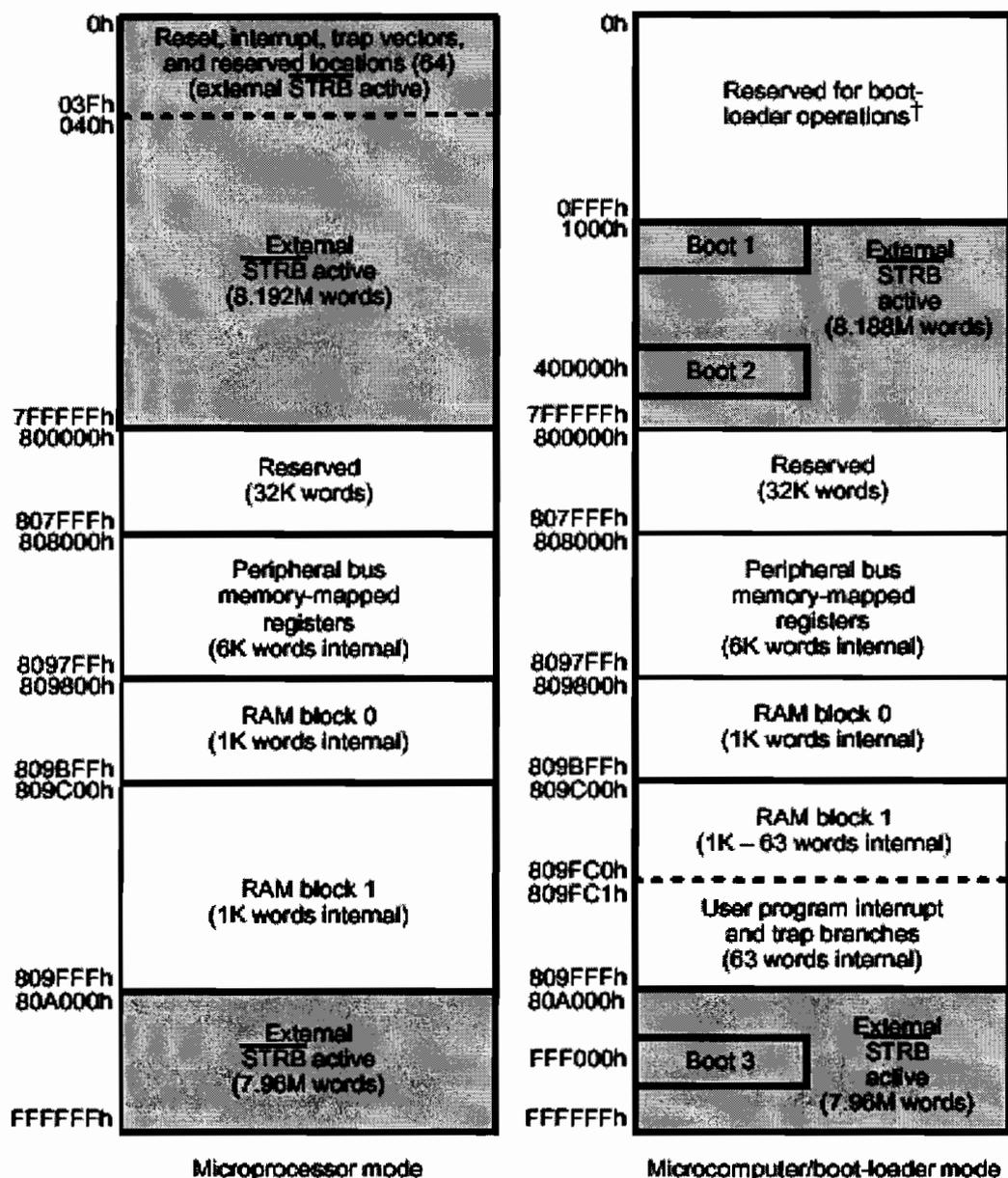


Fig. 3.13 Mapas de memoria del TMS320C31

### 3.2.6 SET DE INSTRUCCIONES

El set de instrucciones está organizado en varios grupos que consisten en instrucciones de cargar/almacenar, aritméticas/lógicas de dos y tres operandos, paralelas, control del programa, y operaciones íter bloqueadas.

El set de instrucciones del TMS320C31, puede también usar uno de los 20 códigos de condición con cualquiera de las 10 instrucciones condicionales, como *LDFcond* (ver Anexo C, *Resumen del Set de Instrucciones del TMS320C31 y Códigos Condicionales*).

El ensamblador permite formas de sintaxis opcionales para simplificar el lenguaje ensamblador para instrucciones especiales.

Todas las instrucciones del set de instrucciones en el TMS320C31 son de una palabra de máquina. La mayoría requieren un ciclo para ejecutarse. Además para multiplicar y acumular instrucciones, posee un complemento de instrucciones de propósito general. El set de instrucciones contiene 113 instrucciones organizadas en los siguientes grupos funcionales:

- Cargar y almacenar (Load-and-Store)
- Aritmética / lógica de 2 operandos
- Aritmética / lógica de 3 operandos
- Control del programa
- Operaciones interconectadas
- Operaciones paralelas

### **3.2.6.1 Cargar y guardar (Load-and-Store)**

Soporta 12 instrucciones, estas instrucciones pueden:

- Cargar una palabra de memoria en un registro
- Almacenar una palabra de un registro en la memoria, o
- Manipular datos en el sistema del stack.

Dos de estas instrucciones permite cargar datos condicionalmente. Esto es útil para localizar el máximo o mínimo valor en un set de datos.

Instrucción	Descripción	Instrucción	Descripción
LDE	Carga exponente de punto flotante	POP	Recupera entero del stack
LDF	Carga valor punto flotante	POPF	Recupera valor p. f. del stack
LDF <i>cond</i>	Carga valor p.f. condicionalmente	PUSH	Pone entero en el stack
LDI	Carga entero	PUSHF	Pone valor p. f. en el stack
LDI <i>cond</i>	Carga entero condicionalmente	STF	Almacena valor punto flotante
LDM	Carga mantisa de punto flotante	STI	Almacena entero
LDP	Carga puntero página de datos		

**Tabla 3.2** Instrucciones para cargar y almacenar

### 3.2.6.2 Instrucciones de dos operandos

Soporta 35 instrucciones aritméticas y lógicas de dos operandos. Los dos operandos son la fuente y el destino. La fuente puede ser una palabra de memoria, un registro, o una parte de la palabra de instrucción. El operando destino es siempre un registro.

Instrucción	Descripción	Instrucción	Descripción
ABSF	Valor absoluto de un # p. f.	NORM	Normaliza un valor p. f.
ABSI	Valor absoluto de un entero	NOT	Complemento lógico
ADDC*	Suma enteros con carry	OR*	OR lógico
ADDF*	Suma valores de punto flotante	RND	Redondea un valor p. f.
ADDI*	Suma enteros	ROL	Rota hacia la izquierda
AND*	AND lógico modo de bit	ROLC	Rota hacia la izquierda con carry
ANDN*	AND lógico con complemento	ROR	Rota hacia la derecha
ASH*	Traslado aritmético	RORC	Rota hacia la derecha con carry
CMPF*	Compara valores de p. f.	SUBB*	Resta entero con borrow
CMPI*	Compara enteros	SUBC	Resta enteros condicionalmente
FIX	Convierte un valor p. f. a entero	SUBF	Resta valores p. f. condicional
FLOTA	Convierte un entero a valor p. f.	SUBI	Resta entero
LSH*	Traslado lógico	SUBRB	Resta entero invertido con borrow
MPYF*	Multiplifica valores p. f.	SUBRF	Resta valor p. f. invertido

MPYI*	Multiplica enteros	SUBRI	Resta entero invertido
NEGB	Niega entero con borrow	TSTB*	Prueba los campos de bits
NEGF	Niega un valor en punto flotante	XOR	OR exclusivo
NEGI	Niega un entero		

\*Versiones de 2 y 3 operandos

**Tabla 3.3** Instrucciones de dos operandos

### 3.2.6.3 Instrucciones de tres operandos

Las instrucciones de tres operandos permiten al TMS320C31 leer dos operandos de la memoria o el archivo registro del CPU en un único ciclo y almacenar los resultados en un registro. Las instrucciones de tres operandos tienen 2 operandos fuente y un operando destino. Un operando fuente puede ser una palabra de memoria o un registro. El destino es siempre un registro. Se puede omitir el 3.

Instrucción	Descripción	Instrucción	Descripción
ADDC3	Suma con carry	MPYF3	Multiplica valores p. f.
ADDF3	Suma valores de punto flotante	MPYI3	Multiplica enteros
ADDI3	Suma enteros	OR3	OR lógico
AND3	AND lógico	SUBB3	Resta enteros con borrow
ANDN3	AND lógico con complemento	SUBF3	Resta valores de punto flotante
ASH3	Traslado aritmético	SUBI3	Resta enteros
CMPF3	Compara valores punto flotante	TSTB3	Prueba campos de bits
CMPI3	Compara enteros	XOR3	OR exclusivo lógico
LSH3	Traslado lógico		

**Tabla 3.4** Instrucciones de tres operandos

### 3.2.6.4 Instrucciones de control de programa

El grupo de instrucciones de control de programa consiste en todas aquellas instrucciones (17) que afectan el flujo del programa. El modo repetitivo permite la repetición de un bloque de código (RPTB) o de una línea de código (RPTS). Algunas de estas instrucciones son capaces de operaciones condicionales.

Instrucción	Descripción	Instrucción	Descripción
<i>Bcond</i>	Salto condicional estándar	IDLE	Inactivo hasta interrupción
<i>BcondD</i>	Salto condicional con retardo	NOP	Sin operación
BR	Salto incondicional estándar	<i>RETIcond</i>	Retorno de interrupción cond.
BRD	Salto incondicional con retardo	<i>RETScond</i>	Retorno de subrutina condic.
CALL	Llamada a subrutina	RPTB	Repite bloque de instrucciones
<i>CALLcond</i>	Llamada a subrutina condicional	RPTS	Repite una sola instrucción
<i>Dbcond</i>	Decremento y <i>Bcond</i>	SWI	Interrupción de software
<i>DbcondD</i>	Decremento y <i>BcondD</i>	<i>TRAPcond</i>	Trampa condicional
IACK	Reconocimiento de interrupción		

**Tabla 3.5** Instrucciones de control de programa

### 3.2.6.5 Instrucciones de control de baja potencia

Consisten en tres instrucciones que afectan los modos de baja potencia. La instrucción IDLE2 permite el modo extremo de baja potencia. La instrucción que divide el reloj por 16 (LOPOWER) reduce la velocidad de la entrada del reloj de frecuencia. La instrucción que recupera la velocidad del reloj normal (MAXSPEED) causa la reanudación de operación a velocidad máxima.

Instrucción	Descripción	Instrucción	Descripción
IDLE2	Inactivo en baja potencia	MAXSPEED	Restaura el reloj a la veloc.normal
LOPOWER	Divide el reloj por 16		

**Tabla 3.6** Instrucciones de control de baja potencia

### 3.2.6.6 Instrucciones de operaciones ínter bloqueadas

Éstas soportan comunicación multiprocesador y el uso de señales externas para permitir sincronización potente de mecanismos. Las instrucciones también garantizan la integridad de la comunicación y resulta en una alta velocidad de operación.

Instrucción	Descripción	Instrucción	Descripción
LDFI	Carga un valor en punto flotante	STFI	Almacena un valor en p. f.
LID	Carga un entero	STII	Almacena un entero
SIGI	Señal ínter bloqueada		

**Tabla 3.7** Instrucciones de operaciones ínter bloqueadas

### 3.2.6.7 Instrucciones de operaciones paralelas

El grupo de instrucciones de operaciones paralelas hace posible un alto grado de paralelismo. Algunas de las instrucciones del TMS320C31 pueden ocurrir en pares que serán ejecutados en paralelo. Estas instrucciones ofrecen las siguientes características:

- Carga paralela de los registros
- Operaciones aritméticas paralelas

- Instrucciones aritméticas / lógicas usadas en paralelo con una instrucción de almacenar.

Cada instrucción en un par es ingresada como una expresión fuente separada. La segunda instrucción en el par debe estar precedida por dos barras verticales (||). La siguiente tabla muestra los pares de instrucciones válidos.

Instrucción	Descripción
<b>Instrucciones aritméticas con almacenamiento</b>	
ABSF    STF	Valor absoluto de un número en punto flotante y almacenamiento de un valor en punto flotante
ABSI    STI	Valor absoluto de un entero y almacena un entero
ADDF3    STF	Suma valores en punto flotante y almacena valor en punto flotante
ADDI    STI	Suma enteros y almacena un entero
AND3    STI	AND lógico y almacena un entero
ASH3    STI	Traslado aritmético y almacena un entero
FIX    STI	Convierte punto flotante en entero y almacena un entero
FLOAT    STF	Convierte entero en punto flotante y almacena un valor en punto flotante
LDF    STF	Carga un valor en punto flotante y almacena un valor en punto flotante
LDI    STI	Carga un entero y almacena un entero
LSH3    STI	Traslado lógico y almacena un entero
MPYF3    STF	Multiplica valores en punto flotante y almacena un valor en punto flotante
MPYI3    STI	Multiplica y entero y almacena un entero

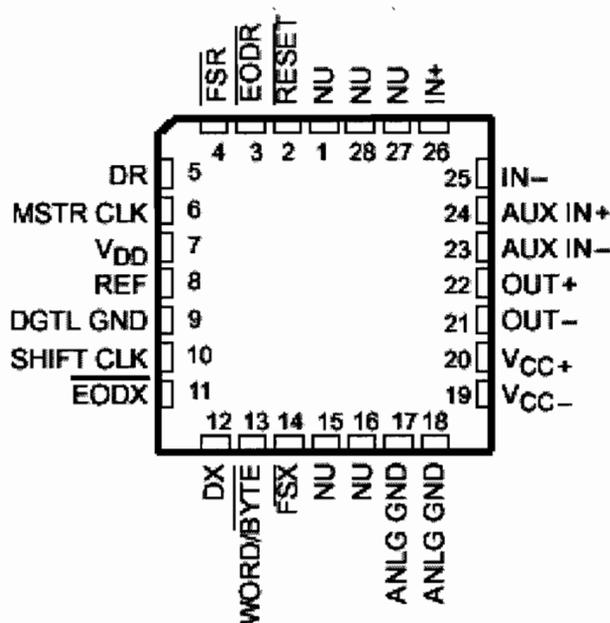
NEGF    STF	Niega un valor en punto flotante y almacena un valor en punto flotante
NEGI    STI	Niega un entero y almacena un entero
NOT    STI	Complementa un valor y almacena un entero
OR3    STI	OR lógico y almacena un entero
STF    STF	Almacena valores en punto flotante
STI    STI	Almacena enteros
SUBF3    STF	Resta un valor en punto flotante y almacena un valor en punto flotante
SUBI3    STI	Resta un entero y almacena un entero
XOR3    STI	OR exclusivo y almacena un entero
<b>Instrucciones de cargar</b>	
LDF    LDF	Carga valores en punto flotante
LDI    LDI	Carga números enteros
<b>Instrucciones para multiplicar, sumar y restar</b>	
MPYF3    ADDF3	Multiplica y suma valores en punto flotante
MPYF3    SUBF3	Multiplica y resta valores en punto flotante
MPYI3    ADDI3	Multiplica y suma enteros
MPYI3    SUBI3	Multiplica y resta enteros

**Tabla 3.8** Instrucciones de operaciones paralelas

Ver Anexo C, *Resumen del Set de Instrucciones del TMS320C31 y Códigos Condicionales*, para aclarar la operación que realiza cada mnemónico y cuáles son los operandos fuente y destino.

### 3.2.7 DESCRIPCIÓN DEL CIRCUITO DE INTERFAZ ANÁLOGO

El circuito de interfaz análogo está diseñado para proveer un alto nivel de integración del sistema y de desempeño. Éste combina una alta resolución de los conversores A/D y D/A, filtros programables, circuitos de control digital así como amplificadores de entrada programables y multiplexores.



**Fig. 3.14** Distribución de pines del TLC32040

El circuito de interfaz TLC32040 es un completo sistema análogo-digital y digital-análogo de entrada/salida en un único chip CMOS. Este dispositivo integra un filtro de entrada "antialiasing" pasa banda con capacitor conmutable, una conversor A/D de 14 bits, cuatro modos de puerto serial microprocesador-compatibles, una conversor D/A de 14 bits, y un filtro de reconstrucción de salida

pasa bajos con capacitor conmutable. El dispositivo ofrece numerosas combinaciones de frecuencias de entrada del reloj master y velocidades de conversión/muestreo, las cuales pueden ser cambiadas vía control digital.

Aplicaciones típicas para este circuito integrado incluye módems, interfaz análogo para DSPs, sistemas de reconocimiento y almacenamiento de voz, control de procesamiento industrial, instrumentación biomédica, procesamiento de señal acústica, análisis espectral, y adquisición de datos. Se tiene un esquema de control flexible que permite que las funciones del circuito integrado puedan ser seleccionadas y ajustadas de acuerdo al procesamiento de la señal vía control por software.

El filtro de entrada antialiasing comprende filtros tipo Chebyshev de séptimo y cuarto orden, pasa bajos y pasa altos, respectivamente. El filtro de entrada es implementado en tecnología de capacitor conmutable y está precedido por un filtro de tiempo continuo para eliminar cualquier posibilidad de aliasing causada por el filtrado de los datos muestreados. Cuando no se desea filtrado, todo el filtro compuesto puede ser desconectado del trayecto de la señal. Una entrada diferencial análoga adicional es proporcionada para aplicaciones en las cuales se requiere más de una entrada análoga, como es el caso de la presente aplicación.

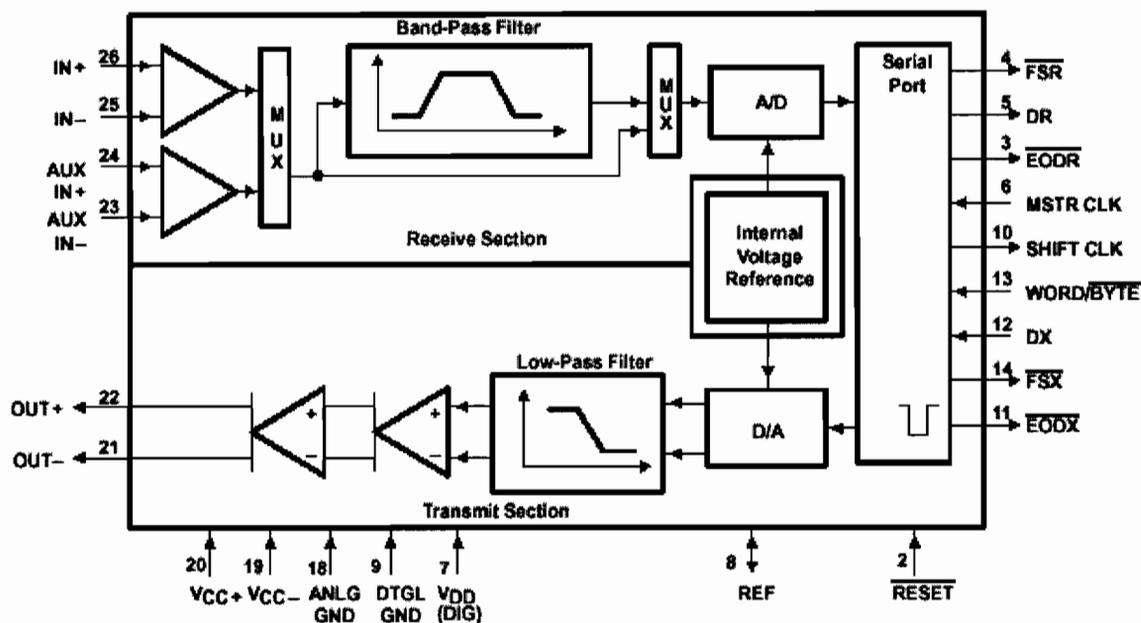
Los conversores A/D y D/A tienen 14 bits de resolución cada uno. Se provee de fuentes de voltaje y tierras para minimizar el ruido y asegurar un amplio rango dinámico. Además, el trayecto del circuito análogo contiene solo circuitería diferencial para mantener el ruido en un mínimo absoluto. La única excepción es el muestreo del DAC, el cual utiliza circuitería pseudo-diferencial.

El filtro de reconstrucción de salida es un tipo Chebyshev pasa bajos de séptimo orden y está implementado en tecnología de capacitor conmutable seguido por un filtro de tiempo continuo para eliminar imágenes de la señal codificada digitalmente.

El circuito de interfaz análogo AIC TLC32040 está caracterizado para una operación entre  $-55^{\circ}\text{C}$  a  $125^{\circ}\text{C}$ .

### 3.2.7.1 Funciones de los Terminales

En la figura 3.15 se presenta un diagrama de bloques funcional del TLC32040



**Fig. 3.15** Diagrama de bloques funcional del AIC TLC32040

**ANLG GND** Tierra análoga para todos los circuitos análogos internos. No está conectada internamente a DGTL GND.

**AUX IN+** Entrada análoga auxiliar no invertida. Esta entrada puede ser escogida vía software. Si el bit apropiado en el registro de control es un 1, las entradas auxiliares reemplazan a las entradas IN+ e IN-. Si el bit es un 0, las entradas IN+ e IN- son usadas.

**DGTL GND** Tierra digital para todos los circuitos lógicos internos. No está conectada internamente a ANLG GND.

<i>DR</i>	DR es usada para transmitir los bits de salida del ADC desde el AIC al puerto serial del TMS320. Esta transmisión de bits está sincronizada con la señal SHIFT CLK.
<i>DX</i>	DX es usada para recibir los bits de entrada del DAC y la información de temporización y control del TMS320. Esta transmisión serial está sincronizada con la señal SHIFT CLK.
$\overline{EODR}$	Final de recepción de datos. Durante el modo de palabra esta señal es un pulso negativo que ocurre inmediatamente después que 16 bits de información del A/D han sido transmitidos del AIC al puerto serial del TMS320. Puede ser usada para interrumpir al microprocesador cuando se han completado las comunicaciones seriales. Durante el modo de byte, se pone en cero luego de que el primer byte se ha transmitido y se mantiene en bajo hasta que el segundo byte ha sido transmitido.
$\overline{EODX}$	Final de transmisión de datos. Durante el modo de palabra, es un pulso bajo que ocurre inmediatamente después de que 16 bits del conversor D/A y la información de control o registro ha sido transmitida del puerto serial del TMS320 al AIC. Puede usarse para interrumpir un microprocesador cuando se han completado las comunicaciones seriales. Durante el modo de byte se pone en bajo después de que el primer byte ha sido transmitido del puerto serial del TMS320 al AIC y se mantiene en bajo hasta que el segundo byte ha sido transmitido.
$\overline{FSR}$	En los modos de transmisión seriales, es mantenida en bajo durante la transmisión. Cuando $\overline{FSR}$ se pone en bajo, el puerto serial TMS320 empieza a recibir bits del AIC vía DR. El bit más significativo de DR está presente en DR antes de que $\overline{FSR}$ se ponga en bajo. $\overline{FSR}$ no ocurre después de la comunicación secundaria.
$\overline{FSX}$	Cuando $\overline{FSX}$ se pone en bajo, el puerto serial del TMS320 inicia transmitiendo bits al AIC vía DX del AIC. En todos los modos de transmisión serial, $\overline{FSX}$ es mantenida en bajo durante la transmisión.

<i>IN+</i>	Entrada no invertida a la entrada análoga de la etapa de amplificación
<i>IN-</i>	Entrada invertida a la entrada análoga de la etapa de amplificación
<i>MSTR CLK</i>	Es usado para derivar todas las señales lógicas significantes del AIC, tales como la señal del shift clock, los relojes de los filtros de capacitor conmutable, y las señales de tiempo del A/D y D/A. Las frecuencias de estas señales son sincrónicas submúltiplos de la frecuencia del reloj master para eliminar el aliasing cuando las señales análogas muestreadas son transferidas entre los filtros de capacitor conmutable y los conversores A/D y D/A.
<i>OUT+</i>	Salida no invertida de la salida análoga del amplificador de poder. Puede manejar cargas de alta impedancia directamente ya sea en configuración diferencial o no diferencia.
<i>OUT-</i>	Salida invertida de la salida análoga del amplificador de poder. Es funcionalmente idéntica y complementaria a <i>OUT+</i> .
<i>REF</i>	Referencia interna de voltaje para el TLC32040. Para el TLC32040 y el TLC32041 una referencia de voltaje externo puede ser aplicada a este terminal.
$\overline{RESET}$	<p>Una función de reset es proporcionada para inicializar TA, TA', TB, RA, RA', RB, y los registros de control. Esta función de reset inicia las comunicaciones seriales entre el AIC y el DSP. La función de reset inicializa todos los registros del AIC incluyendo el registro de control. Después de un pulso negativo en <math>\overline{RESET}</math>, los registros del AIC son inicializados para proporcionar una velocidad de conversión de datos de 8 kHz para una señal de entrada del reloj maestro de 5.184 MHz. Los bits del registro de control son reseteados como sigue:</p> <p>d7=1, d6=1, d5=1, d4=0, d3=0, d2=1</p> <p>La inicialización permite la comunicación por el puerto serial entre el AIC y el DSP</p>
<i>SHIFT CLK</i>	Es obtenido dividiendo la señal de frecuencia del reloj maestro por cuatro. Es usado para cronometrar las transferencias de datos seriales del AIC.

$V_{DD}$	Fuente de voltaje digital, $5\text{ V} \pm 5\%$
$V_{CC+}$	Fuente de voltaje análoga positiva, $5\text{ V} \pm 5\%$
$V_{CC-}$	Fuente de voltaje análoga negativa, $5\text{ V} \pm 5\%$

$\overline{WORD/BYTE}$  En conjunción con un bit en el registro de control, es usado para establecer uno de cuatro modos seriales. Dos de estos cuatro modos seriales utilizados con el DSP TMS320C31 son descritos abajo:

*Las secciones de transmisión y recepción operan asincrónicamente.-*

La siguiente descripción se aplica cuando el AIC es programado para tener secciones de transmisión y recepción asíncronas. Si valor en el bit apropiado del registro de control es un 0 las secciones de transmisión y recepción son asincrónicas.

*Las secciones de transmisión y recepción son operadas sincrónicamente.-*

Si el bit apropiados en el registro de control es un 1, las secciones de transmisión y recepción son configuradas para ser sincrónicas. En este caso las señales  $\overline{FSX}$  y  $\overline{FSR}$  son idénticas durante la comunicación de datos primaria; sin embargo,  $\overline{FSR}$  no es sostenida durante la comunicación secundaria debido a que no hay un nuevo resultado de conversión A/D.

Ver Anexo D, *Hojas de datos del AIC TLC32040*, para mayor información sobre este dispositivo.

## **CAPÍTULO 4**

### **DESARROLLO DEL ALGORITMO PARA LA OBTENCIÓN DEL TORQUE**

## 4 DESARROLLO DEL ALGORITMO PARA LA OBTENCIÓN DEL TORQUE

En este capítulo se describe el software utilizado, así como los pasos seguidos para la adquisición de las señales, su procesamiento digital, la obtención de la señal de torque resultante y su visualización gráfica.

### 4.1 DESCRIPCIÓN DEL SOFTWARE DEL DSK

El ensamblador y depurador del DSK son interfaces de software que ayudan a desarrollar, probar, y depurar los programas en lenguaje ensamblador del DSK. A continuación se presenta una visión general del ensamblador y depurador y se describe sus principales características y funciones.

#### 4.1.1 DESCRIPCIÓN DEL ENSAMBLADOR DEL DSK

El ensamblador del DSK es una herramienta fácil y simple de usar.

El ensamblador del DSK difiere de otros ensambladores porque éste no pasa por una fase enlazadora (linker) para crear un archivo de salida. En su lugar, el DSK usa directivas especiales para ensamblar código en una dirección absoluta durante la fase de ensamble. Como resultado, es posible crear pequeños programas de una manera rápida y fácil. Si se desea crear programas más grandes, se lo puede hacer encadenando archivos con la directiva *.include*.

##### 4.1.1.1 Creación de archivos fuente

Para crear un archivo fuente, no se puede usar cualquier programa editor. Se debe tener cuidado al usar procesadores de palabras ya que estos archivos contienen varios códigos formateados y caracteres especiales.

Los programas fuente en lenguaje ensamblador (assembler) consisten en expresiones fuente que pueden contener directivas assembler, instrucciones en lenguaje ensamblador, y comentarios. Las líneas de la expresión fuente pueden tener más de 80 caracteres por línea. Las siguientes líneas muestran ejemplos de expresiones fuente:

```

        .start "CODE",0x809802 ; Empezar ensamblando código
        .sect  "CODE"         ;
        .entry INICIO         ; Punto de entrada del depurador
        ;-----
INICIO  ldp    @stack         ; Carga una página de datos
        ldi    @stack,SP     ; Carga el puntero del stack
h      .float  0.002
        ldf    @h,R4
        stf    R4,@9e2ah     ; Almacena h
        ldf    0.00,R4
        stf    R4,@9E0Dh     ; Condiciones iniciales x0 = 0
        stf    R4,@9e0eh     ; y0 = 0
        stf    R4,@9e0fh     ; z0 = 0
        stf    R4,@9e10h     ; u0 = 0

```

La expresión fuente puede contener cuatro campos. La sintaxis general para las expresiones fuente es como sigue:

```
[etiqueta][:]  mnemónico  [operandos]  [:comentario]
```

- Todas las expresiones deben comenzar con una etiqueta, un espacio en blanco, un asterisco o un punto y coma.
- Las etiquetas son opcionales; si se las usa, deben escribirse en la columna 1
- Uno o más espacios en blanco deben separar cada campo.
- Los comentarios son opcionales. Los comentarios que comienzan en la columna 1 pueden empezar con un asterisco o un punto y coma, pero los comentarios que comienzan en cualquier otra columna deben empezar con un punto y coma.

### 4.1.1.2 Tipos de Constantes

El ensamblador soporta cinco tipos de constantes:

- Enteros binarios
- Enteros decimales
- Enteros hexadecimales
- Números en punto flotante
- Caracteres

El ensamblador mantiene cada constante internamente como una cantidad de 32 bits.

#### 4.1.1.2.1 Enteros binarios

Una constante entera binaria es una cadena de 0s y 1s seguida del sufijo B ó b. Por ejemplo:

0101b	constante igual a 5
10101B	constante igual a 21
-0101b	constante igual a -5

#### 4.1.1.2.2 Enteros decimales

Una constante entera decimal es una cadena de dígitos decimales, en el rango de -2 147 483 647 a 4 294 967 295. Por ejemplo:

1000	constante igual a $3E8_{16}$
-32768	constante igual a $8000_{16}$
25	constante igual a $19_{16}$

#### 4.1.1.2.3 Enteros hexadecimales

Una constante hexadecimal entera es una cadena de más de 8 dígitos hexadecimales seguidos por el sufijo H ó h o precedidos por el prefijo 0x. Los

dígitos hexadecimales incluyen los valores decimales 0-9 y las letras A-F o a-f. Una constante hexadecimal debe comenzar con un valor decimal (0-9). Ejemplos de constantes hexadecimales válidas incluyen:

78H	Constante igual a 120 <sub>10</sub>
0x0f	Constante igual a 15 <sub>10</sub>
37ACh	Constante igual a 14 2252 <sub>10</sub>

#### 4.1.1.2.4 *Números de punto flotante*

Una constante de punto flotante es una cadena de dígitos decimales, seguidos por un punto decimal opcional, porción fraccional, y un exponente. Ejemplos de números en punto flotante incluyen:

1.75e-10	representado internamente como 2202 629A <sub>16</sub>
4	representado internamente como 0200 0000 <sub>16</sub>
-3.5	representado internamente como 01A0 0000 <sub>16</sub>
3.2e5	representado internamente como 12E3 C000 <sub>16</sub>

Una constante de punto flotante puede ser precedida con un signo + o -.

#### 4.1.1.2.5 *Caracteres*

Una constante caracter es un único caracter encerrado en comillas simples. Los caracteres son representados como caracteres ASCII de 8 bits. Ejemplos de constantes caracter válidas incluyen:

'a'	representada internamente como 0000 0061 <sub>16</sub>
'C'	representada internamente como 0000 0043 <sub>16</sub>

Es importante notar la diferencia entre constantes caracter y cadenas caracter. Una constante representa un valor entero simple y está encerrada en comillas simples; una cadena es una lista de caracteres y está encerrada en comillas dobles.

#### 4.1.1.3 Cadenas de caracteres

Es una cadena encerrada en comillas dobles. La longitud máxima de la cadena varía y es definida por cada directiva que requiera una cadena. Ejemplos de cadenas de caracteres válidas incluyen:

“programa”

“temp.asm”

Las cadenas de caracteres son usadas para nombres de archivos, nombres de secciones, o como operando de la directiva `.string`

#### 4.1.1.4 Forma de ensamblar el programa

Antes de intentar depurar un programa, es necesario ensamblarlo. A continuación se presenta el comando para invocar el ensamblador cuando se prepara un programa para ser depurado

```
dsk3a nombre del archivo [opciones]
```

donde

<b>dsk3a</b>	es el comando que invoca al ensamblador
nombre del archivo	es el archivo fuente en lenguaje ensamblador.
opciones	afectan la forma en que el ensamblador procesa los archivos de entrada.

Al ingresar este comando, el ensamblador crea un archivo ejecutable con extensión `.dsk`. Este archivo usado para cargar código ejecutable directamente en el DSK. El archivo ejecutable incluye una lista de todos los errores y advertencias que pueden haber ocurrido durante el ensamble del programa. Esta lista es útil porque contiene una lista de todos los símbolos y código por resolver. En la figura 4.1 se representa la ventana del ensamblador una vez que el ensamblador ha sido invocado y se ha ensamblado el programa.

```

MS-DOS PROMPT
(c) Copyright 1994-2000 Texas Instruments Incorporated
Fri Nov 02 10:57:22 2001
File TURQUE.ASM
PASS
PASS
PASS PASS 1 Complete
PASS Errors: 0 warnings: 0
PASS
PASS TURQUE.ASM line 309: ERROR 0 WARNING 1
PASS WARNING: value converted to zero
PASS mpyf wr,R7
PASS
PASS TURQUE.ASM line 320: ERROR 0 WARNING 2
PASS WARNING: value converted to zero
PASS mpyf wr,R1
PASS
PASS
PASS PASS 2 Complete
PASS Errors: 0 warnings: 2
PASS
PASS ENTRY: 0x00009802
PASS
PASS END DSK
C:\ADSP\DSK1.B>

```

**Fig. 4.1** Pantalla de resultados del ensamblador del DSK

#### 4.1.2 DESCRIPCIÓN DEL DEPURADOR DEL DSK

El depurador es fácil de aprender y de usar. Su interfaz orientada a ventanas es amigable y reduce el tiempo de aprendizaje así como elimina la necesidad de memorizar comandos complejos. El depurador puede cargar y ejecutar código paso a paso y con puntos de parada y detención. El depurador corre y depura código en un DSP real, a diferencia de un simulador que utiliza un computador para simular un DSP.

Para invocar al depurador se utiliza el siguiente comando:

```
dsk3d [opciones]
```

**dsk3d** es el comando que invoca al depurador  
**opciones** proporciona información adicional al depurador

La figura 4.2 identifica algunas características de la pantalla del depurador. Cuando se invoca el depurador tecleando **dsk3d**, se verá una pantalla similar a la siguiente:

The screenshot shows the DSK3D debugger interface. At the top, it says 'MS-DOS PROMPT - DSK3D'. Below the title bar, there are window control buttons and a menu bar. The main area is divided into several sections:

- Registers:** A list of registers (D, C, etc.) with their current values in hexadecimal.
- Code:** A list of assembly instructions with their addresses and values.
- COMMAND:** A text input field with the prompt '>Enter a command or F1 for help' and a 'CMD>' prompt below it.
- MEMORY:** A section showing memory addresses and their corresponding values.
- Function Key Legend:** A row of function keys (F1-F10) with their corresponding actions: F1 to F9 are 'F1 to F9' and F10 is 'F10'.

Fig. 4.2 Pantalla básica del depurador

#### 4.1.2.1 Características del depurador

- El depurador del DSK separa código, datos, y comandos en porciones manejables.
- A diferencia de otros sistemas depuradores, este no obliga a aprender un largo y complicado conjunto de comandos. El depurador soporta un pequeño, pero poderoso, conjunto de comandos.
- Existen dos formas principales de ingresar comandos. Se lo puede hacer en la línea de comando o usar las teclas de función.

## 4.1.2.2 Ventanas del depurador

### 4.1.2.2.1 Ventana DISASSEMBLY

La ventana DISASSEMBLY muestra el assembler inverso del contenido de la memoria. Como se muestra en la figura 4.3, esta ventana despliega algunas líneas de código. Cada línea muestra la dirección de la instrucción, código de la instrucción, etiqueta, y mnemónico de la instrucción. Cuando una línea está acentuada representa la siguiente instrucción a ser ejecutada.

```

DISASSEMBLY
00000000 62809857 CALL FLOG2
00000001 07601400 LDF 3.000000e+00,R0
00000002 62809833 CALL LOG2
00000003 6a00ffff B OUTER
00000004 04603000 SQRT CMPF 0.000000e+00,R0
00000005 43e0f000 LDPLI -1.000000e+00,R0
00000006 42e00000 LDPEQ 0.000000e+00,R0
00000007 70000000 RETSLE
00000008 07010000 LDP R0,R1
00000009 09e10001 LSH 1,R1
0000000a 0fa10000 PUSHF R1
0000000b 0e210000 POP R1
0000000c 0a010001 NOT R1,R1
0000000d 03e1ffff ASH -1,R1
0000000e 0f210000 PUSH R1
0000000f 0ea10000 POPF R1
00000010 09e1ffff LSH -1,R1
00000011 0a601000 MPYF 5.000000e-01,R0
00000012 00760004 LD1 4,RC
00000013 64809820 RPTB NB_SQRT
00000014 24020101 MPYF3 R1,R1,R2
00000015 0a020000 MPYF R0,R2
00000016 19620400 SUBRF 1.500000e+00,R2
00000017 0a010002 NR_SQRT MPYF R2,R1
00000018 11010001 RND R1,R1
00000019 0a000001 MPYF R1,R0
0000001a 0a601000 MPYF 2.000000e+00,R0
0000001b 70000000 RETS
0000001c fff7ffff MASK .word 0ff7ffffh
0000001d 07010000 INUF LDF R0,R1
0000001e 0fa10000 PUSHF R1
0000001f 0e210000 POP R1
00000020 1aa19825 XOR @MASK,R1
00000021 0f210000 PUSH R1
00000022 0ea10000 POPF R1
00000023 00760004 LD1 4,RC
00000024 64809830 RPTB NR_INU

```

Fig. 4.3 Ventana DISASSEMBLY

#### 4.1.2.2.2 Ventana CPU REGISTERS

La ventana CPU REGISTER despliega el contenido de todos los registros del CPU como se muestra en la figura. 4.4. Los contenidos de los registros son normalmente representados en formato hexadecimal. Se puede presionar la tecla F3 para mostrar los registros de precisión extendida en formato de punto flotante. Se puede presionar F2 para presentar los registros de precisión extendida en formato hexadecimal de 40 bits.

```

CPU REGISTERS
00 0080782d
01 00809c01
02 0140000000
03 fe3fffff00
04 0000000040
05 0000000000
06 0000000000
07 0000000000
08 0000000000
09 0000000000
0a 0000000000
0b 0000000000
0c 0000000000
0d 0000000000
0e 0000000000
0f 0000000000
10 00007f04
11 00007f05
12 00007f01
13 0000000e
14 00000000
15 00000000
16 00000000
17 00000000
18 00000002
19 00000000
1a 00000000
1b 00000004
1c 0080981d
1d 00809820
1e 00000000
1f 00000000
20 00000000
21 000000c4
22 00000304
23 00003941
  
```

**Fig. 4.4** Ventana CPU REGISTERS

#### 4.1.2.2.3 Ventana MEMORY

La ventana MEMORY muestra los contenidos de un rango de memoria como se muestra en la figura 4.5. La ventana MEMORY tiene dos partes:

- **Direcciones** La primera columna de números identifica las direcciones de la primera columna de datos desplegados. No importa cuántas columnas de datos se despliegue, solo una columna de direcciones es desplegada. Cada dirección en esta columna identifica la dirección de los datos que están inmediatamente a su derecha.
- **Datos** Las columnas restantes representan los valores de las direcciones listadas.

Por ejemplo, la ventana MEMORY abajo tiene cuatro columnas de datos, así cada nueva dirección es incrementada en 4. A pesar de que la ventana muestra cuatro columnas de datos, todavía hay solo una columna de direcciones.

```

MEMORY
69029140 00000063 50700000 00349861
07601400 6280980d 07601400 62809826
07601400 62809857 07601400 62809833
6a00ffff 04608000 43e0f800 42e00000
78880000 07010000 09e10001 0fa10000
0e210000 0d010001 03e1ffff 0f210000
0ea10000 09e1ffff 0a60f000 0b7h0004
64809820 24820101 0a020000 19620400
0a010002 11010001 0a000001 0a601000

```

**Fig. 4.5** Ventana MEMORY

#### 4.1.2.2.4 Ventana COMMAND

La ventana COMMAND proporciona un área para ingresar comandos, repetir comandos, y desplegar errores de salida y mensajes. La ventana COMMAND tiene dos partes:

- **Línea de comando** Esta es un área donde se ingresan comandos. Cuando se requiere ingresar un comando, solo se teclea, no importa cual ventana está activa.

- *Área de visualización* Esta área repite los comandos ingresados, muestra cualquier salida de estos comandos, y presenta mensajes de error del depurador.

La figura 4.6 muestra la línea de comando y el área de visualización de esta ventana.



```
COMMAND
load math
>load math
load math
```

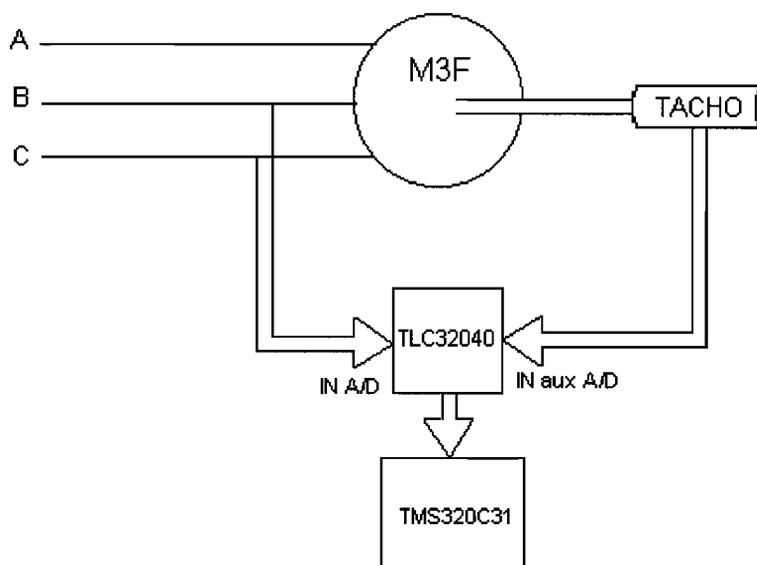
**Fig. 4.6** Ventana COMMAND

Ver *Manual de Usuario del TMS320C3x DSP Starter Kit*, para mayor información sobre este software.

## 4.2 ACONDICIONAMIENTO DE SEÑALES ADQUIRIDAS

Para la implementación del programa que realiza el cálculo del torque es necesario, en primer lugar, censar las variables utilizadas en el desarrollo matemático de la máquina de inducción con el fin de que el procesador digital de señales las pueda procesar realizando sobre éstas todas las operaciones pertinentes a fin de obtener el resultado requerido. Estas señales son: la corriente, el voltaje y la velocidad. Sin embargo, dichas señales requieren ser acondicionadas antes de ingresar en la tarjeta de evaluación. Para realizar el análisis es necesario tomar al menos dos de estas tres señales, en el caso del presente proyecto se toman las señales de voltaje y velocidad.

Por lo tanto, se realiza un muestreo de las señales que provienen de las líneas de alimentación de la máquina y de un tacho generador que es el que proporciona una señal de voltaje que refleja la velocidad a la que está operando el motor. Las señales análogas que son tomadas deben ser convertidas a señales digitales por un conversor A/D previo a ser evaluadas en el algoritmo del DSP, así como se muestra en la figura 4.7.



**Fig. 4.7** Adquisición de señales de voltaje y velocidad

En el siguiente desarrollo se muestra la forma en que estas dos señales fueron tratadas a fin obtener los niveles deseados para que en el módulo de evaluación se pueda implementar la rutina para la posterior obtención del Torque.

#### 4.2.1 SEÑAL DE VOLTAJE

El motor de inducción a ser utilizado es un motor con rotor bobinado, conectado en estrella, y tiene las siguientes características:

- Potencia  $P = 2.25 \text{ HP}$
- Voltaje nominal de línea  $V = 380 \text{ V}$

- Corriente nominal  $I = 4.04 \text{ A}$
- Velocidad nominal  $\omega_m = 1800 \text{ rpm}$
- Número de polos  $P = 4$

En el motor, al ser alimentado con un voltaje entre líneas de 380V, se tiene que el pico de esta tensión va a ser de 537.4 V. Considerando que el conversor análogo–digital que forma parte del AIC soporta un voltaje de entrada máximo igual a  $\pm 6\text{V}$ , para poder introducir el voltaje al módulo de evaluación se hace imprescindible que éste pase antes por una etapa de acondicionamiento.

El acondicionamiento de la señal de voltaje se logra mediante un transformador con una relación de transformación igual a 95. Dicho transformador, a más de reducir el voltaje hasta un valor de 4 Vrms, aísla las dos señales de voltaje de una eventual falla que pueda ocurrir en el voltaje de línea que sería el que alimenta al primario del transformador, es decir desacopla la señal.

Cabe notar en este punto que al ser reducido el voltaje a un valor de 4 Vrms, el máximo valor que estaría ingresando al AIC de la tarjeta es el valor pico de éste, es decir 5.66 V, que resulta un nivel adecuado para que el módulo funcione correctamente.

#### 4.2.2 SEÑAL DE VELOCIDAD

La otra señal que se va a introducir al conversor análogo–digital es la señal de la velocidad, la misma que va a ser representada por valores de voltaje obtenidos mediante la utilización de un taco generador que refleja en valores de voltaje la velocidad a la que está girando el motor. Este dispositivo no es más que un pequeño generador DC de excitación independiente que genera tensión continua a sus terminales al girar conjuntamente acoplado al motor en estudio.

En el caso de esta aplicación, el taco generador (sensor de velocidad) tiene los siguientes datos:

Cuando se alimenta el campo con un valor de tensión de 6.36 V y el motor gira a 1760 rpm, el taco generador genera un voltaje de 5.5 V y una componente de alterna de frecuencia 2.5 kHz y voltaje 3 mV pico.

Para filtrar el ruido se ha implementando un filtro pasa bajos RC con una frecuencia de corte muy baja para que la señal sea continua. Para el filtro, se ha propuesto una frecuencia de corte de alrededor de 5 Hz, cabe mencionar que el valor de la reactancia propia del capacitor debe ser tal que no disminuya el valor del voltaje generado, es decir, junto con la resistencia del filtro deben tener un valor de impedancia tal que se obtengan 5 V a la salida del filtro cuando el rotor gira a 1760 rpm.

Filtro:

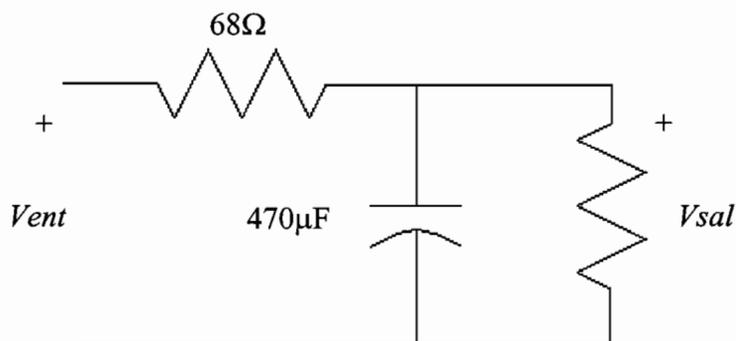
$$f_c = \frac{1}{2\pi RC}$$

$f_c = 5 \text{ Hz}$ .

Sea  $R = 68\Omega$ ,

El valor del capacitor es de  $470 \mu\text{F}$ .

El filtro implementado es el siguiente:



**Fig. 4.8** Filtro implementado para la señal proveniente del taco generador

Es importante mencionar que si se utiliza un taco generador en donde no se pueda controlar el voltaje generado con el campo, se tendrá que acondicionar la señal para que a la velocidad nominal se genere 5.5 V

De esta manera se ha obtenido valores adecuados de las señales de voltaje y velocidad para que éstas puedan ingresar a la tarjeta de evaluación DSK.

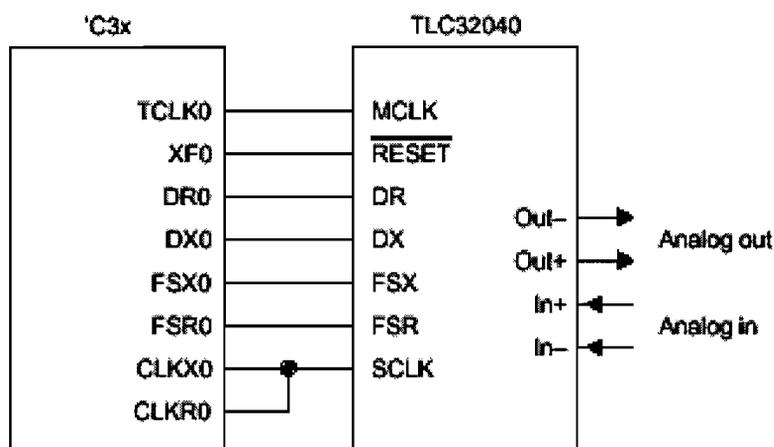
### **4.3 CONVERSIÓN ANÁLOGA-DIGITAL DE LAS SEÑALES DE VOLTAJE Y VELOCIDAD**

Una vez acondicionadas las señales que reflejan tanto el voltaje entre líneas de la máquina, así como la velocidad a la que gira el rotor, dichas señales se introducen al circuito de interfase análogo AIC TLC32040, descrito en el capítulo anterior, proporcionado por la tarjeta DSK.

Como se había mencionado, éste es un circuito completo de interfaz tanto análogo-digital como digital-análogo el cual se programa vía software. Para usar el AIC, se debe seguir una secuencia de pasos para inicializar el temporizador del 'C31 y el puerto serial, y para arrancar y programar el AIC para que éste tome muestras de la señal de voltaje y la señal de velocidad alternadamente. Como se puede observar en la figura 3.15 se tienen dos grupos de entradas análogas. Normalmente es usado el grupo de entradas IN+ e IN-; sin embargo, el set de entradas auxiliares, AUX IN+ y AUX IN-, puede usarse si una segunda entrada es requerida como en el caso del presente proyecto. Cada entrada puede ser o no operada en modo diferencial, en el presente caso no son operadas en modo diferencial. Cualquier circuito de entrada puede ser seleccionado vía control por software. Es importante notar que un amplio rango dinámico es asegurado por la arquitectura análoga diferencial interna y por las fuentes de voltaje análoga y digital separadas así como por las tierras análoga y digital separadas.

La figura 4.9 muestra como se interconecta el 'C31 con el circuito de interfaz análogo AIC. A continuación se describirá los pasos necesarios para inicializar y

configurar el temporizador del 'C31 y el puerto serial del mismo, así como la forma de arrancar y programar el TLC32040 para cumplir con el propósito del presente proyecto.



**Fig. 4.9** Interconexión del 'C31 con el TLC32040

#### 4.3.1 INICIALIZACIÓN DEL AIC TLC32040

La inicialización del sistema puede ser dividida en dos distintas partes:

- Inicialización del temporizador y el puerto serial (frecuencia del Reloj Master, registros de transmisión y recepción, etc.)
- Inicialización del AIC (frecuencia de muestreo, frecuencia de corte del filtro pasa banda, y otros parámetros)

Como se muestra en la figura 4.9, la señal XF0 del 'C31 está conectada a la señal de  $\overline{RESET}$  del AIC. Manejando la señal de  $\overline{RESET}$ , el 'C31 puede arrancar el AIC mediante la ejecución de ciertas instrucciones.

#### 4.3.1.1 Inicialización del temporizador del 'C31

Como se muestra en la figura 4.9, la señal del temporizador del 'C31 (TCLK0) está conectada a la señal del reloj master (MCLK) del AIC. La señal MCLK maneja todas las señales lógicas del AIC, tales como el shift clock, los relojes de los filtros de capacitor conmutable, y las señales de tiempo del A/D y el D/A. El temporizador pulsa la señal TCLK0 cuandoquiera que el registro contador del temporizador del 'C31 (ubicado en la dirección 0x0080 8024h de memoria) cuente hasta el valor del registro del período del temporizador (ubicado en la dirección 0x0080 8028h). Entonces, los registros contadores del temporizador se ponen en cero y repiten. Debido a las diferencias entre la frecuencia máxima del temporizador del 'C31 y la frecuencia máxima y mínima del AIC es importante observar ciertas restricciones:

- *Valor mínimo del registro de período del temporizador.* El 'C31 50 MHz puede generar una frecuencia máxima de reloj de 12.5 MHz (CLKIN/4), la cual está sobre la frecuencia máxima probada del reloj master del AIC de 10 MHz. Si se usan frecuencias más allá de aquellas listadas en las hojas de datos el desempeño resultante no puede ser predecible. Si el temporizador es puesto en marcha en modo de pulso (valor de control 0x2C1) el período mínimo de 1 resulta en una velocidad master de 12.5 MHz y 2 resulta en 6.25 MHz.
- *Valor máximo del registro de período del temporizador.* La frecuencia mínima del reloj master del AIC es 75 kHz. Tomando en cuenta la frecuencia máxima del temporizador del 'C31 de 12.5 MHz y la frecuencia mínima del reloj master del AIC, el valor máximo del registro contador del temporizador debería ser 165 ( $12.5 \text{ MHz} / 75 \text{ kHz} = 166.7$ ). El temporizador del 'C31 cuenta hasta 0, por lo tanto, es necesario sustraer 1 de este número ( $166 - 1 = 165$ ). Notar que la especificación del TLC32040 describe una frecuencia mínima de reloj debido a que las señales internas del AIC son almacenadas en capacitores que deben ser actualizados periódicamente.

- *Inicialización del temporizador.* Un período de cero no está permitido en el modo de pulso. Si el temporizador es puesto en marca en modo de reloj, la salida resultante es una onda cuadrada con una frecuencia mitad del modo de pulso. Un período de cero es permitido en este modo resultado en un reloj de 12.5 MHz.

#### **4.3.1.2 Inicialización del puerto serial del 'C31**

Para inicializar el puerto serial es necesario inicializar el registro de control del puerto serial ubicado en memoria en la dirección 0x0080 8040h, el registro de control de transmisión FSX/DX/CLKX ubicado en memoria en la dirección 0x0080 8042h, y el registro de control de recepción FSR/DR/CLKR ubicado en memoria en la dirección 0x0080 8043h.

La inicialización del registro de control global del puerto serial se realiza de la siguiente manera:

- Emitiendo resets de transmisión y recepción
- Habilitando interrupciones de transmisión y recepción
- Estableciendo transferencias de transmisión y recepción de 16 bits.
- Estableciendo modo continuo

#### **4.3.1.3 Inicialización del AIC**

La inicialización del AIC está basada en tres parámetros:

- El registro de control del AIC (para establecer ganancia, sincronización, etc.)
- El registro TX del contador A (y el registro RX del contador A si el modo sincrónico es deshabilitado)
- El registro TX del contador B (y el registro RX del contador B si el modo sincrónico es deshabilitado)

Una vez que el 'C31 proporciona la señal MCLK, inicializa su puerto serial, y arranca el AIC, se puede inicializar el AIC a una velocidad de muestreo especificada. La velocidad de muestreo del AIC es determinada por los valores de dos registros llamados A y B en las secciones de transmisión y recepción del AIC. Estos valores son cargados en el respectivo contador cuandoquiera que el contador llega a 0. TA y TB determinan el tiempo de conversión del D/A, RA y RB determinan el tiempo de conversión del A/D. La fórmula para la frecuencia de conversión está dada en la ecuación 4.1.

$$\text{Frecuencia de conversión} = \frac{MCLK}{2xAxB} \quad (4.1)$$

Para que los filtros pasa bajos y pasa banda encuentren las especificaciones de su función de transferencia, la frecuencia de las entradas de reloj de los filtros debe ser 288 kHz; de otro modo, las frecuencia superior e inferior son escaladas como corresponde. La siguiente ecuación muestra la frecuencia del filtro de capacitor conmutable (switched-capacitor filter).

$$\text{Frecuencia de reloj SCF} = \frac{MCLK}{2XA} \quad (4.2)$$

Los registros TX del contador A y del contador B son recargados cada período de conversión del D/A, mientras que los registro RX del contador A y del contador B son recargados cada período de conversión del A/D.

Una vez que la frecuencia MCLK es fijada, la frecuencia de muestre  $F_s$  depende del valor TA x TB. De esta manera TA y TB son definidos con las siguientes restricciones:

- TA es un entero en el rango de 4 a 31
- TB es un entero en el rango de 2 a 63
- La frecuencia de reloj SCF cercana y menor que  $F_s/2$

Para señales que tienen contenido en frecuencia limitado (ancho de banda finito), el teorema del muestreo especifica la forma óptima de interpolación. En principio, la señal analógica puede reconstruirse a partir de sus muestras, siempre que la tasa de muestreo sea lo suficientemente alta como para evitar el problema comúnmente denominado *aliasing* (algunos autores traducen este término por solapamiento). La máxima frecuencia de una señal en tiempo continuo que puede determinarse unívocamente cuando dicha señal se muestra a una velocidad  $F_s$  es  $F_{\max} = F_s/2$ .

Dada una señal analógica cualquiera, para elegir la velocidad de muestreo  $F_s$  es necesaria cierta información sobre las características de la señal que va a ser muestreada. En particular, se debe tener información general sobre el contenido frecuencial de la señal. Generalmente, dicha información se encuentra disponible. Por ejemplo, la señal de voltaje a ser muestreada es una señal sinusoidal que tiene una frecuencia de 60 Hz. El conocimiento de  $F_{\max}$  permite seleccionar la velocidad de muestreo apropiada. Se sabe que la frecuencia más alta de la señal analógica que puede reconstruirse sin ambigüedad cuando la señal se muestra a una velocidad  $F_s = 1/T$  es  $F_s/2$ . Cualquier frecuencia por encima de  $F_s/2$  o por debajo de  $-F_s/2$  produce muestras que son idénticas a las correspondientes frecuencias dentro del intervalo  $-F_s/2 \leq F \leq F_s/2$ . Para evitar las ambigüedades que resultan del aliasing, se debe seleccionar una velocidad de muestreo lo suficientemente alta, esto es, se debe escoger  $F_s/2$  mayor que  $F_{\max}$ . Por tanto, para evitar el problema del aliasing, se selecciona  $F_s$  según:

$$F_s > 2F_{\max} \quad (4.3)$$

donde  $F_{\max}$  es la frecuencia más alta de la señal analógica. Con la velocidad de muestreo seleccionada de esta manera se garantiza que todas las componentes sinusoidales de la señal analógica se correspondan con componentes en frecuencia de tiempo discreto en el intervalo fundamental. Por lo tanto, todas las componentes en frecuencia de la señal analógica están representadas sin ambigüedad en la forma muestreada de la señal, y así la señal analógica puede

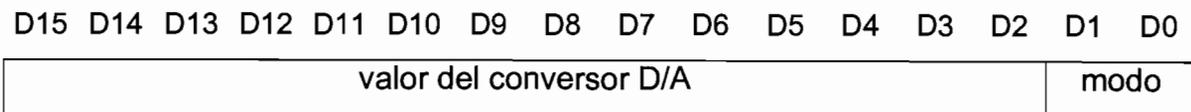
ser reconstruida sin distorsión a partir de las muestras. Consecuentemente, la frecuencia de conversión a ser establecida en el AIC debe ser mayor a 120 Hz.

Se debe tomar en cuenta que al tomar muestras de los dos canales del AIC en forma alternada la frecuencia de muestreo establecida en el AIC se reduce a la mitad; por lo tanto, para aprovechar la velocidad de muestreo del AIC se procurará trabajar con una frecuencia de muestreo alta.

Para programar el AIC se requiere realizar una comunicación primaria y una comunicación secundaria.

#### 4.3.1.3.1 Comunicación Primaria

Las comunicaciones primarias tienen un valor en los 14 bits más significativos (D15-D2) de datos y una selección de modo en los 2 bits menos significativos (D1-D0). Este formato se muestra en la figura 4.10.



**Fig. 4.10 Formato de Datos de la Comunicación Primaria**

El AIC envía un valor al conversor D/A y habilita uno de los modos mostrados en la Tabla 4.1 dependiendo del valor de los bits D1 y D0.

LSBs	Mode
00	Tx Counter A $\underline{\underline{=}}$ TA, Rx Counter A $\underline{\underline{=}}$ RA Tx Counter B $\underline{\underline{=}}$ TB, Rx Counter B $\underline{\underline{=}}$ RB
01	Tx Counter A $\underline{\underline{=}}$ TA + TA', Rx Counter A $\underline{\underline{=}}$ RA + RA' Tx Counter B $\underline{\underline{=}}$ TB, Rx Counter B $\underline{\underline{=}}$ RB
10	Tx Counter A $\underline{\underline{=}}$ TA - TA', Rx Counter A $\underline{\underline{=}}$ RA + RA' Tx Counter B $\underline{\underline{=}}$ TB, Rx Counter B $\underline{\underline{=}}$ RB
11	Tx Counter A $\underline{\underline{=}}$ TA, Rx Counter A $\underline{\underline{=}}$ RA Tx Counter B $\underline{\underline{=}}$ TB, Rx Counter B $\underline{\underline{=}}$ RB

**Tabla 4.1** Selección del modo de la comunicación primaria

El segundo y tercer modo usan los registros TA' y RA' para adelantar o retrasar la frecuencia de muestreo acortando o alargando el período de muestreo. Esto es particularmente útil en aplicaciones de módem. Estableciendo los 2 bits menos significativos en 1 en la comunicación primaria se da paso a las comunicaciones secundarias una vez completadas las comunicaciones primarias.

#### 4.3.1.3.2 Comunicación Secundaria

Las comunicaciones secundarias siguen a la comunicación primaria que tiene los dos bits menos significativos como 11. Esta comunicación secundaria programa el AIC cargando A, A', B, o los registros de control. La figura 4.11 muestra el formato de datos de la comunicación secundaria. El registro de control habilita y deshabilita las entradas auxiliares, filtros pasa banda, entre otros. La figura 4.12 describe los campos de bits del Registro de Control.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
X	X	TA register value (unsigned)					X	X	RA register value (unsigned)					0	0
X	TA' register value (signed 2s complement)					X	RA' register value (signed 2s complement)					0	1		
X	TB register value (unsigned)					X	RB register value (unsigned)					1	0		
X	X	X	X	X	X	X	X	Control register					1	1	

**Fig. 4.11** Formato de datos de la comunicación secundaria

D7	D6	D5	D4	D3	D2
<b>Input Gain</b>		<b>Transmit/Receive</b>	<b>AUX IN Pins</b>	<b>Loopback Function</b>	<b>Bandpass Filter</b>
0	0 = 1X for — 6V analog Input	0 = asynchronous	0 = disables	0 = disables	0 = deletes
0	1 = 2X for — 3V analog Input	1 = enables	1 = enables	1 = enables	1 = inserts
1	0 = 4X for — 1.5V analog input				
1	1 = 1X for — 6V analog Input				

**Fig. 4.12** Campos de bits del Registro de Control

## 4.4 DESCRIPCIÓN DEL ALGORITMO IMPLEMENTADO EN EL DSP TMS320C31

A continuación se va a describir el algoritmo a ser implementado para la obtención del torque de la máquina de inducción trifásica, el cual se desarrolla siguiendo los requerimientos que se van a necesitar para que el circuito de interfaz análogo digital que posee la tarjeta DSK (AIC), y el análisis matemático tengan una correcta sincronización.

En el procesador, el programa desarrollado se puede separar en varias etapas, y se basa en la utilización de subrutinas, las mismas que le dan versatilidad. Las distintas etapas se muestran en el diagrama de bloques de la figura 4.13.

### 4.4.1 INICIALIZACIÓN

En la sección de inicialización, en primer lugar se definen las etiquetas de direcciones de registros a utilizarse, se definen también las secciones de código ejecutable y de vectorización de interrupciones, así como los valores más usados en el programa, se ubican valores en palabras consecutivas de la sección presente con sus respectivas etiquetas, y se especifica la dirección de inicio del programa.

#### 4.4.2 ESTABLECIMIENTO DE PARÁMETROS DEL AIC

En esta sección se establecen los parámetros del AIC como son el período del temporizador, los valores de los registros TA y TB para seleccionar la velocidad de muestreo, el valor de la comunicación primaria para especificar como van a ser cargados los valores en los registros TA y TB, valor de los registros para la comunicación secundaria incluyendo el registro de control que habilita o deshabilita el filtro y la entrada auxiliar AUXIN (ver sección 4.3.1.3 página 111).

TIMERPER = 0

TA = 8

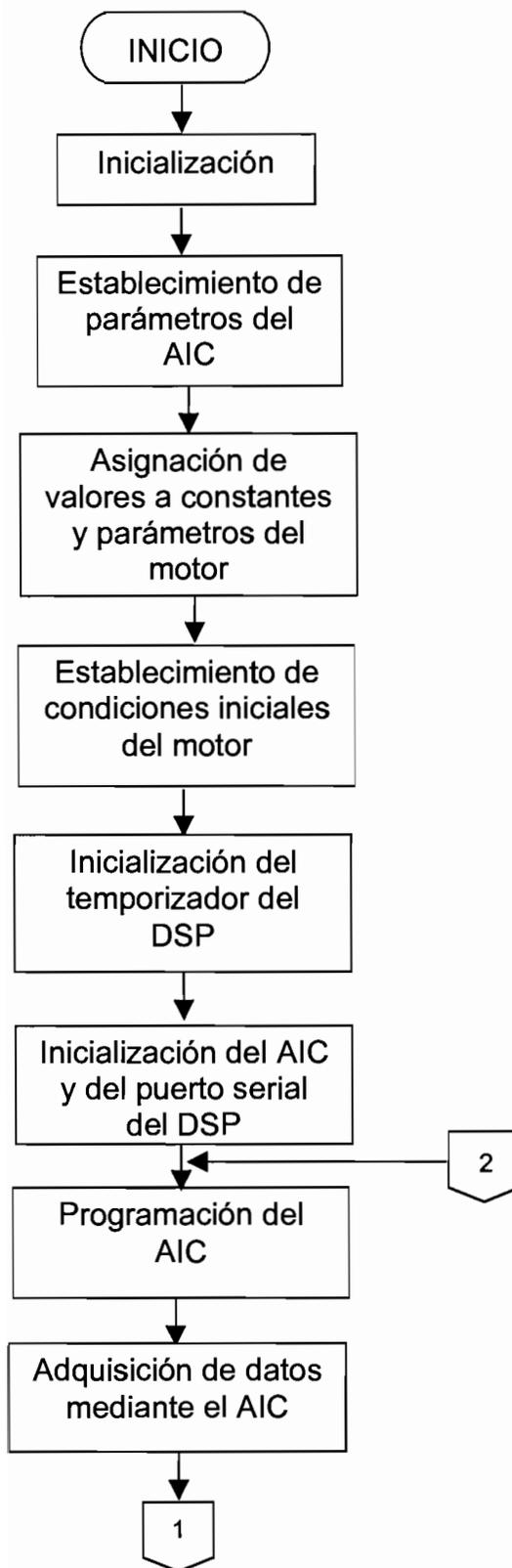
TB = 14

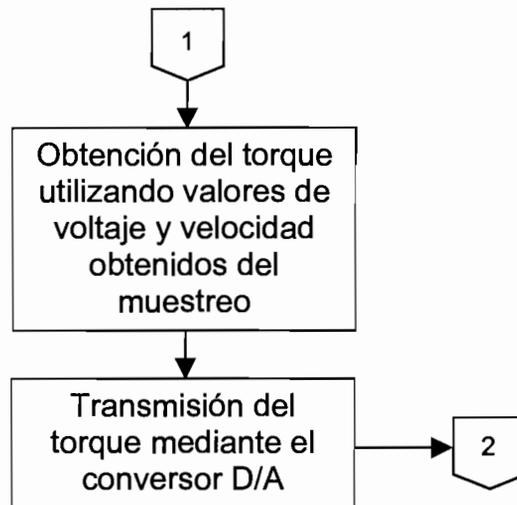
RA = 8

RB = 14

Con los parámetros así establecidos se obtiene la máxima frecuencia de muestreo del AIC. En realidad esta frecuencia se reduce a la mitad debido a que se obtienen muestras de las señales de los dos canales alternadamente, una cada vez. De esta manera se aprovecha la velocidad de muestreo del AIC y se logra que las señales adquiridas no sufran modificaciones y/o atenuaciones.

Además, es necesario en la comunicación primaria establecer los 2 bits menos significativos en 1 con el fin de que se pueda dar paso a las comunicaciones secundarias, y en éstas se debe remover el filtro de entrada del conversor A/D, pues cuando el filtro está habilitado se pierde la habilidad de convertir los dos canales debido a que el filtro es común a las dos entradas. El filtro, efectivamente, suma las dos entradas antes de filtrarlas como una señal.





**Fig. 4.13** Diagrama de bloques del programa en el DSP

#### **4.4.3 ASIGNACIÓN DE VALORES A CONSTANTES Y PARÁMETROS DEL MOTOR**

Aquí se establecen los valores de las constantes utilizadas en el desarrollo del programa, así como los valores de los parámetros del motor con sus respectivas etiquetas, para que luego puedan ser utilizados inmediatamente sin necesidad de ser guardados en localidades de memoria específicas.

Para calcular los parámetros del motor bobinado en por unidad se tomó como referencia valores de los parámetros del motor previamente calculados en trabajos anteriores<sup>[7]</sup>, así como las siguientes cantidades base de acuerdo a las características del motor mencionadas anteriormente.

$$V_b = \frac{380}{\sqrt{3}} = 219.3931 \text{ V}$$

$$I_b = 4.04 \text{ A}$$

$$Z_b = \frac{V_b}{I_b} = 54.3052 \text{ } \Omega$$

$$S_b = V_b \times I_b = 886.3481 \text{ VA}$$

$$\omega_b = 376.992 \text{ rad / s}$$

$$\omega_{bm} = \frac{2\omega_b}{P} = 188.496$$

$$T_b = \frac{S_b}{\omega_{bm}} = 4.7022$$

Los parámetros del motor en estudio son los siguientes:

Resistencia del estator	$r_s = 1.4988 \text{ } \Omega$
Resistencia del rotor	$r'_r = 4.9255 \text{ } \Omega$
Reactancia del estator	$x_{ls} = 5.7835 \text{ } \Omega$
Reactancia del rotor	$x'_{lr} = 5.7835 \text{ } \Omega$
Reactancia del magnetización	$x_m = 104.8525$

Partiendo de estos valores se obtienen los siguientes valores en por unidad de los parámetros del motor:

$$r_{s \text{ p.u.}} = 0.0276$$

$$r'_{r \text{ p.u.}} = 0.0907$$

$$x_{ls \text{ p.u.}} = 0.1065$$

$$x'_{lr \text{ p.u.}} = 0.1065$$

$$x_{m \text{ p.u.}} = 1.9308$$

Adicionalmente se requiere ingresar las siguientes constantes:

Paso de iteración	$h = 0.002$
-------------------	-------------

Valor pico del voltaje del línea	$D = 537.4012$
----------------------------------	----------------

Valor pico del voltaje de fase	$C = 310.2687$
--------------------------------	----------------

Número de polos	$p = 4$
Valor inverso del voltaje base	$\text{invVb} = 0.004558$
Valor de la velocidad base	$w_b = 377$
Valor inverso de la velocidad base	$\text{invwb} = 2.6525e-3$

#### 4.4.4 ESTABLECIMIENTO CONDICIONES INICIALES DEL MOTOR

En esta sección se establecen las condiciones iniciales del motor en el momento del arranque, esto es, se establecen los valores de las concatenaciones de flujo por segundo. Las concatenaciones de flujo por segundo son las variables del sistema de ecuaciones diferenciales que representan el modelo matemático de la máquina de inducción en base a las cuales se realiza el presente programa. Esto se realiza en esta sección del programa debido a que una vez que corre el programa se debe inicializar una sola vez, pues estos valores son función del tiempo y por lo tanto varían con el mismo. Por lo tanto, las condiciones iniciales se establecen iguales a cero.

#### 4.4.5 INICIALIZACIÓN DEL TEMPORIZADOR DEL DSP

El temporizador tiene tres registros que deben ser configurados para inicializarlo (ver 4.3.1.1 página 110):

- El registro de control global, que determina el modo de operación del temporizador, monitorea el estado del temporizador, y controla la función del pin de entrada / salida del temporizador
- Registro de período, que especifica la frecuencia de la señal del temporizador
- Registro contador, que contiene el valor actual del contador. Es posible incrementar el temporizador en el lado ascendente o en el lado descendente de la señal de entrada de reloj. El contador se pone en cero y puede originar una interrupción cuando su valor igual al valor del registro de período. El generador de pulsos genera dos tipos de señales externas de reloj: pulso o reloj.

En este caso el registro de control global del temporizador ha sido configurado para que trabaje en modo de reloj.

El mapa de memoria del temporizador se muestra a continuación.

808020h	Timer0 global control†
808024h	Timer0 counter†
808028h	Timer0 period†
808030h	Timer1 global control†
808034h	Timer1 counter†
808038h	Timer1 period†

**Fig. 4.14** Localidades de los temporizadores 0 y 1 en el mapa de memoria

#### 4.4.6 INICIALIZACIÓN Y PROGRAMACIÓN DEL AIC

Esta sección llama a una subrutina llamada AICINIT que se encarga de inicializar el AIC, inicializar el DMA, inicializar el puerto serial y realizar la comunicación primaria y la comunicación secundaria, con el fin de que este dispositivo se encuentre listo para realizar la primera adquisición de datos.

##### 4.4.6.1 Inicialización del puerto serial del DSP

Como se mencionó en la sección 4.3.1.2 (página 111), es necesario inicializar el puerto serial del DSP. El puerto serial del DSP puede configurarse para transferir 8, 16, 24, o 32 bits de datos por palabra simultáneamente en ambas direcciones. El reloj para el puerto serial puede originarse ya sea internamente, vía el temporizador del puerto serial y los registros de período, o externamente. Un

modo de transferencia continua está disponible, el cual permite al puerto serial transmitir y recibir cualquier número de palabras sin nuevos pulsos de sincronización.

Los registros proporcionados para el puerto serial son ocho:

- Registro de control global
- Dos registros de control para los seis pines seriales de E/S
- Tres registros temporizadores de transmisión / recepción
- Registro de transmisión de datos
- Registro de recepción de datos

El registro de control global controla las funciones globales del puerto serial y determina el modo de operación del puerto serial. En este caso debe configurarse para permitir transferencias de 16 bits. Los dos registros de control son configurados como pines de puerto serial. El buffer de transmisión contiene la siguiente palabra completa a ser transmitida. El buffer de recepción contiene la última palabra completa recibida.

El mapa de memoria del puerto serial se representa en la figura 4.15 a continuación.

808040h	Serial-port 0 global control†
808042h	Serial port 0 FSX/DX/CLKX control†
808043h	Serial port 0 FSR/DR/CLKR control‡
808044h	Serial port 0 R/X timer control¶
808045h	Serial port 0 R/X timer counter#
808046h	Serial port 0 R/X timer period
808048h	Serial port 0 data transmit*
80804Ch	Serial port 0 data receive□

**Fig. 4.15** Localidades del puerto serial en el mapa de memoria

#### 4.4.6.2 Inicialización del DMA del DSP

El 'C31 tiene un chip controlador de acceso directo a memoria (DMA) que reduce la necesidad de que el CPU realice funciones de entrada/salida. El controlador DMA puede realizar operaciones de entrada/salida sin interferir con la operación del CPU. Por lo tanto, es posible interconectar el 'C31 con memorias externas y componentes periféricos (A/Ds, puertos seriales, etc.) lentos, sin reducir la cantidad de datos transmitidos del CPU. El resultado es un rendimiento del sistema mejorado.

Una transferencia del DMA consiste de dos operaciones: una lectura de una localidad de memoria y una escritura en una localidad de memoria. El DMA puede leer desde y escribir en cualquier localidad en el mapa de memoria del 'C31. Esto incluye todas los componentes periféricos que localizados en el mapa de memoria. La operación del DMA es controlada con el siguiente grupo de registros:

- Registro de control global ubicado en la dirección 808000h
- Registro de dirección-fuente ubicado en la dirección 808004h
- Registro de dirección-destino ubicado en la dirección 808006h
- Registro contador de transferencia ubicado en la dirección 808008h

Para inicializar o reconfigurar el DMA se deben seguir los siguientes pasos:

1. Parar el DMA escribiendo un cero en el registro de control global del DMA.
2. Configurar el DMA mediante el registro de control global así como los registros fuente, destino, y contador de transferencias.
3. Arrancar el DMA estableciendo los bits START del registro de control global como sea necesario.

En el presente programa el DMA se utiliza para actualizar el AIC.

#### **4.4.7 ADQUISICIÓN DE DATOS MEDIANTE EL AIC**

En primer lugar se establece qué canal se va a muestrear. Una vez realizado esto se llama a una subrutina llamada XMIT, la misma que se encarga de configurar el AIC para que adquiera la señal escogida.

A continuación se llama a una nueva subrutina llamada RECV. Esta subrutina determina el canal muestreado, adquiere el dato del conversor A/D, y almacena dicho dato en una localidad de memoria para luego poder utilizarla.

Cuando ya se tiene almacenado el primer dato se procede a muestrear la señal ingresada por el canal auxiliar de la misma forma, y se llama nuevamente a la subrutina RECV para determinar el canal, adquirir la señal y almacenarla en una nueva dirección de memoria.

Como se puede observar el AIC está siendo reprogramado continuamente, por lo cual el Acceso Directo a Memoria (DMA) es utilizado para actualizar el AIC constantemente. El conversor A/D alterna canales en cada muestra. Esto, efectivamente reduce la velocidad de muestreo a la mitad.

Cuando el programa ha adquirido las dos señales, estas están digitalizadas y listas para ser utilizadas en el algoritmo para calcular el torque, a continuación se describe las partes del algoritmo del torque, qué se obtiene en cada una de ellas y como se implementan estos resultados para el objetivo planteado.

#### **4.4.8 OBTENCIÓN DEL TORQUE UTILIZANDO DATOS DE VOLTAJE Y VELOCIDAD MUESTREADOS**

Una vez establecidos los parámetros del motor, así como las condiciones iniciales de las ecuaciones para la obtención de los flujos electromagnéticos, éstos se utilizan para la posterior obtención de la corriente, la cual es utilizada en las fórmulas finales que deducen el torque de la máquina, según el marco de referencia estacionario.

Los parámetros del motor son utilizados para obtener las constantes que se van a necesitar en las ecuaciones diferenciales que se obtienen finalmente para el cálculo de los flujos electromagnéticos (según ecuación 2.61 página 40).

Las constantes calculadas en el programa son:

$$x_{ss} = x' r r = x l s + x m$$

$$D = x_{ss}(x' r r) - x m^2$$

$$a1 = \frac{r s(x_{ss})}{D}$$

$$a2 = \frac{r s(x m)}{D}$$

$$b1 = \frac{r' r(x m)}{D}$$

$$b2 = \frac{r' r(x_{ss})}{D}$$

#### 4.4.8.1 Acondicionamiento de la señal de voltaje

A continuación se toma la primera señal adquirida, es decir el voltaje, que ya se encuentra almacenada en una localidad de memoria del DSP. Sin embargo, este valor debe acondicionarse pues se utilizó un transformador de 380/4 V por lo cual el valor almacenado no es el valor real con el cual se debe trabajar para poder realizar los cálculos sobre la base de los niveles de tensión reales de la máquina.

De acuerdo al desarrollo del marco de referencia estacionario, se necesita las tres señales de voltajes de fase además de la señal de la velocidad para desarrollar las ecuaciones que permiten la obtención del torque de la máquina de inducción trifásica. Sin embargo, el conversor A/D disponible en la tarjeta permite únicamente el muestreo de dos señales. Consecuentemente, solo se va a muestrear la señal de voltaje de una de las fases y se va a obtener las otras dos a partir de la primera en forma analítica.

Como se sabe, la señal de voltaje disponible se obtiene de la red, que a su vez alimenta al motor. Las tres señales de voltaje están desfasados  $120^\circ$  eléctricos entre sí y tienen la misma amplitud.

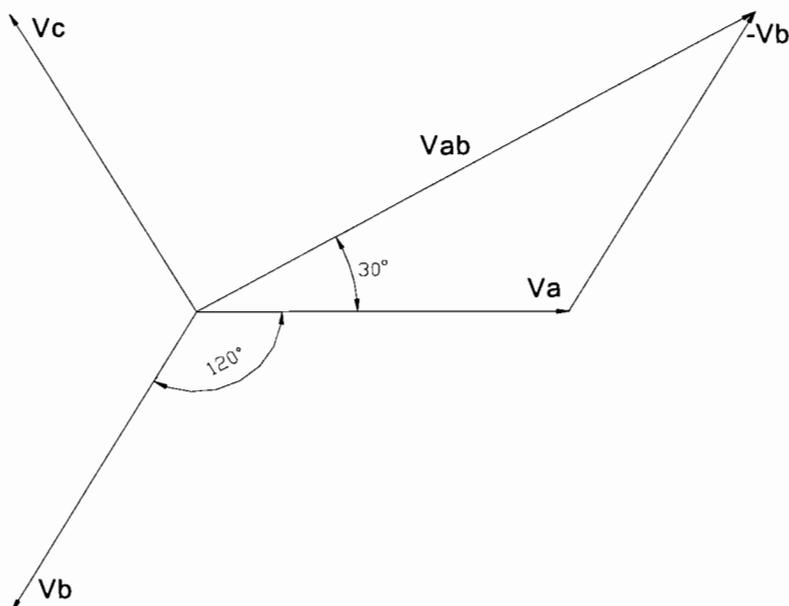
El voltaje que se introduce para ser convertido en una señal digital, corresponde al voltaje entre líneas. No obstante, se necesita el voltaje por fase, ya que el algoritmo necesita voltajes  $a$ ,  $b$  y  $c$  para su evaluación. Esto no ocurriría si los devanados del motor estuviesen conectados en triángulo; sin embargo, los devanados del motor que se va a utilizar están conectados en estrella.

Los voltajes que se muestran son sinusoidales, por lo que éstos tienen la forma:

$$\begin{aligned}V_{AB} &= A \sin \theta \\V_{BC} &= A \sin(\theta - 120^\circ) \\V_{CA} &= A \sin(\theta + 120^\circ)\end{aligned}\tag{4.4}$$

donde  $A$  es la amplitud de la señal, es decir el voltaje pico o voltaje máximo que puede obtenerse a un tiempo o instante determinado.

En el análisis fasorial de los voltajes, los voltajes de fase respecto a los de línea están desfasados  $30^\circ$  respectivamente, así como muestra la figura 4.16, con esto el análisis de los voltajes se realiza partiendo de la ecuación 4.4.



**Figura 4.16** Desfasamiento de los voltajes de fase respecto a los de línea

Se parte de que el voltaje entre líneas es la suma vectorial del voltaje entre fases y se determina el ángulo de desfase entre el voltaje  $V_{ab}$  y el voltaje  $V_a$ , así como el módulo del voltaje de fase relacionado con el de línea. A partir de esto se deducen los voltajes de fase. A continuación se presenta el desarrollo matemático de lo expuesto:

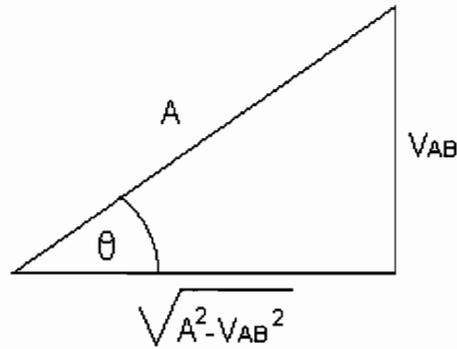
De esto se concluye primero:

$$V_a = \frac{A}{\sqrt{3}} \operatorname{sen}(\theta - 30^\circ) \quad (4.5)$$

desarrollando la expresión del seno de la resta de dos números a partir de la ecuación anterior, se tiene:

$$V_a = \frac{A}{\sqrt{3}} (\operatorname{sen}\theta \cos 30^\circ - \operatorname{sen} 30^\circ \cos\theta) \quad (4.6)$$

Según la ecuación 4.4, lo obtenido en la práctica es la señal del voltaje entre líneas, y debido a que la señal es sinusoidal, ésta se la puede representar en el diagrama siguiente:



**Fig. 4.17** Diagrama triangular del voltaje entre líneas

Observando la figura 4.17, la ecuación 4.6 se puede escribir como sigue:

$$V_a = \frac{A}{\sqrt{3}} \left( \frac{\sqrt{3} V_{AB}}{2 A} - \frac{1}{2} \frac{\sqrt{A^2 - V_{AB}^2}}{A} \right) \quad (4.7)$$

Para la obtención de los otros voltajes de fase, se parte de una definición del voltaje  $V_a$  y se hace una deducción similar pero teniendo en cuenta que el ángulo de desfase entre los voltajes de fase es de  $120^\circ$ . A continuación se presenta el desarrollo matemático de la deducción de los voltajes de fase  $V_b$  y  $V_c$ .

$$V_a = C \operatorname{sen} \theta$$

$$V_b = C \operatorname{sen}(\theta - 120^\circ)$$

$$V_c = C \operatorname{sen}(\theta + 120^\circ)$$

$$V_b = C(\operatorname{sen} \theta \cos 120 - \operatorname{sen} 120 \cos \theta)$$

$$V_b = C \left( \frac{V_a}{C} (-0.5) - \frac{\sqrt{3}}{2} \frac{\sqrt{C^2 - V_a^2}}{C} \right)$$

Entonces :

$$V_b = -\frac{1}{2} V_a - \frac{\sqrt{3}}{2} \sqrt{C^2 - V_a^2}$$

$$V_c = C(\operatorname{sen} \theta \cos 120 + \operatorname{sen} 120 \cos \theta)$$

$$V_c = C \left( \frac{V_a}{C} (-0.5) + \frac{\sqrt{3}}{2} \frac{\sqrt{C^2 - V_a^2}}{C} \right)$$

Finalmente :

$$V_c = -\frac{1}{2} V_a + \frac{\sqrt{3}}{2} \sqrt{C^2 - V_a^2}$$

donde C es la amplitud del voltaje de fase, es decir:

$$C = \frac{A}{\sqrt{3}}$$

En el presente trabajo, se ha implementado la deducción de los tres voltajes de fase en el algoritmo que utiliza el DSP para el cálculo del torque de la máquina de inducción, de esta manera, simplemente se necesita adquirir la señal de un solo voltaje entre líneas y los demás quedan deducidos a partir de éste. Cabe destacar también, que cualquier perturbación en las líneas de alimentación, se verá reflejado en todos los voltajes que alimentan al motor.

Una vez que se han obtenido los voltajes de fase, el siguiente paso es calcular los voltajes  $v_q$  y  $v_d$  del marco de referencia estacionario utilizando las ecuación 2.62

para que puedan ser reemplazados en el sistema de ecuaciones diferenciales de la ecuación 2.61.

#### **4.4.8.2 Acondicionamiento de la señal de velocidad**

Tal como se acondicionó la señal de voltaje, se procede ahora a acondicionar vía software la señal de la velocidad adquirida que está ubicada en una localidad de memoria. De la misma forma, tomando en cuenta que el acondicionamiento de la señal de velocidad por hardware permitía obtener 5 V a la entrada del canal auxiliar cuando el motor gira a máxima velocidad, el valor almacenado debe ser transformado proporcionalmente para obtener la velocidad en radianes por segundo.

#### **4.4.8.3 Algoritmo de Runge-Kutta**

A continuación se llama a una subrutina llamada R\_K la misma que se encarga de resolver el sistema de ecuaciones diferenciales de primer orden propuesto en la ecuación 2.61., mediante la utilización de métodos numéricos.

Los métodos numéricos, son en la actualidad la herramienta más difundida en el lenguaje computacional para la resolución de múltiples problemas matemáticos que para su solución se necesitan muchas iteraciones, que a la final no es el objetivo de la ingeniería, pero que sin embargo es de utilidad resolverlos. Son capaces de manejar sistemas de ecuaciones grandes, no linealidades y geometrías complicadas que son comunes en la práctica de la ingeniería y que algunas son imposibles de resolverlas analíticamente.

Con esta herramienta matemática, no se pierde tiempo en la solución de un problema, más bien el tiempo que se emplearía en ello, se lo podría emplear para la formulación del problema y para la interpretación de los resultados del mismo.

Según lo mostrado en el desarrollo en un marco de referencia estacionario de la máquina de inducción, los valores del flujo electromagnético se obtienen a partir

de un sistema de cuatro ecuaciones diferenciales ordinarias lineales y de primer orden.

En la ecuación, a la cantidad que se le va a diferenciar, se le llama variable dependiente. A la cantidad respecto a la cual se le va a derivar, se le conoce con el nombre de variable independiente. Cuando la función incluye una variable independiente, se le llama *ecuación diferencial ordinaria*, que está en contraste con las ecuaciones diferenciales parciales que comprenden dos o más variables independientes.

A las ecuaciones diferenciales también se las clasifica por su orden, se le llama de primer orden cuando la derivada mayor es la primera derivada, que es el caso del presente proyecto.

Para comprender de una mejor manera el método numérico para la solución de una ecuación diferencial, recuerde que una ecuación diferencial introduce la derivada de la ecuación original o la razón de cambio, que es la pendiente en un punto determinado de la ecuación, entonces para averiguar el siguiente punto en la ecuación, se debe tener el punto actual, al cual se le suma la pendiente (derivada) multiplicada por el tamaño del paso. Mientras el tamaño del paso sea más pequeño, más exacta será la solución o curva de aproximación a la función original.

Los métodos de Runge-Kutta, son los más utilizados en la solución de ecuaciones diferenciales, ya que estos métodos reducen el error que se da por la derivada, ya que se supone que la derivada se da al principio del intervalo, estos métodos, calculan un promedio de la derivada, tanto en el primer como en el siguiente punto del intervalo.

Entonces, para este caso, se necesita conocer las condiciones iniciales de flujo. Estos valores se dan cuando la máquina está parada, es decir las condiciones iniciales son cero, ya que no existe inducción de flujo electromagnético cuando no existe rotación debido a que el campo no está alimentado.

A partir de las condiciones iniciales, y utilizando el método de Runge-Kutta de cuarto orden, se trazan los siguientes puntos de los flujos, que se dan tanto en el eje  $q$  como en el eje  $d$ , en el rotor y en el estator.

Una vez obtenidos los siguientes puntos del flujo, éstos son utilizados para despejar las corrientes de los ejes de referencia a partir de la ecuación 2.59 en la página 40. En otras palabras, cada valor de flujo, modifica el anterior valor de las corrientes en un determinado eje.

Finalmente, todos los valores obtenidos, tanto de flujos como de corrientes, son utilizados para el cálculo del torque de acuerdo a lo expuesto en el desarrollo matemático de la máquina que fue mostrado en el capítulo 2 (ver ecuación 2.57 página 38).

Cabe señalar que mientras exista una variación de la velocidad o del voltaje de alimentación, el valor del torque va a estar siendo modificado, pues, el algoritmo va a tomar diferentes condiciones para el cálculo de todas las variables involucradas.

Para la implementación de la adecuación por software de la señal de voltaje, así como de las diferentes constantes introducidas por las reactancias de magnetización, éste algoritmo, presenta además subrutinas para cálculos de raíces cuadradas y de valores inversos de las cantidades manejadas, debido a que estas funciones no están disponibles en el lenguaje assembler.

#### **4.4.9 TRANSMISIÓN DEL TORQUE MEDIANTE EL CONVERTOR D/A**

Una vez que se retorna de la subrutina para el cálculo del torque, y que el resultado obtenido ha sido almacenado en una localidad de memoria, éste es transmitido para ser convertido por el conversor D/A del circuito de interfaz análogo AIC, y su posterior visualización gráfica. El lazo se vuelve a repetir para tener los nuevos datos de voltaje y velocidad. Es decir, el torque se calcula para cada muestra de las señales tomadas y es transmitido cada vez.

## **4.5 INTERFAZ GRÁFICA PARA LA VISUALIZACIÓN DEL RESULTADO**

Una vez que se ha obtenido el torque y éste ha sido almacenado en la dirección 809e91h y ha sido convertido mediante la utilización del D/A, la señal análoga es ingresada en la tarjeta de adquisición de datos DAQ LAB – PC1200 (100 muestras por segundo) disponible en el Laboratorio de Instrumentación, con el fin de visualizar gráficamente el resultado utilizando LabView.

## **CAPÍTULO 5**

### **PRUEBAS Y CONCLUSIONES**

## 5 PRUEBAS Y CONCLUSIONES

### 5.1 PRUEBAS EXPERIMENTALES

Para realizar las pruebas y comparar los resultados con el presente proyecto se tomó como referencia el medidor de torque análogo del que se dispone en el Laboratorio de Máquinas Eléctricas.

El objetivo de las presentes pruebas es determinar el torque del motor trifásico de inducción para distintos pasos de carga, observar su variación y comparar los resultados.

Se realizaron las siguientes pruebas:

- Arranque sin carga
- Añadir pasos de carga
- Arranque con carga
- Disminución de la carga
- Aumento de la carga
- Cambiando pasos de carga

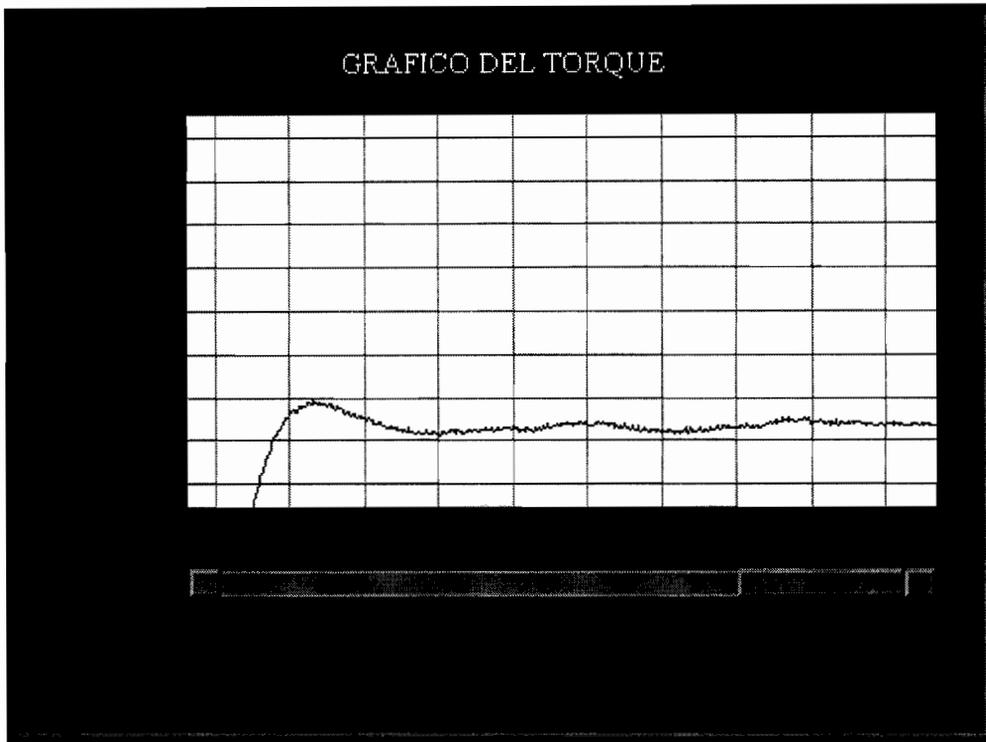
#### 5.1.1 ARRANQUE SIN CARGA

Se alimentó el motor con el voltaje nominal y sin conectar carga. Se obtuvo los siguientes datos:

Lectura	Torque
0.28 kgf	0.55 Nm

Siendo el radio  $R = 0.2$  m, se tiene que el torque en Nm es igual a:  $0.28$  [kgf] x  $0.2\text{m} \times 9.8 \text{ m/s}^2 = 0.49$  Nm

El resultado obtenido con el DSK se presenta en la figura 5.1.



**Fig. 5.1** Arranque sin carga

### 5.1.2 AÑADIR PASOS DE CARGA

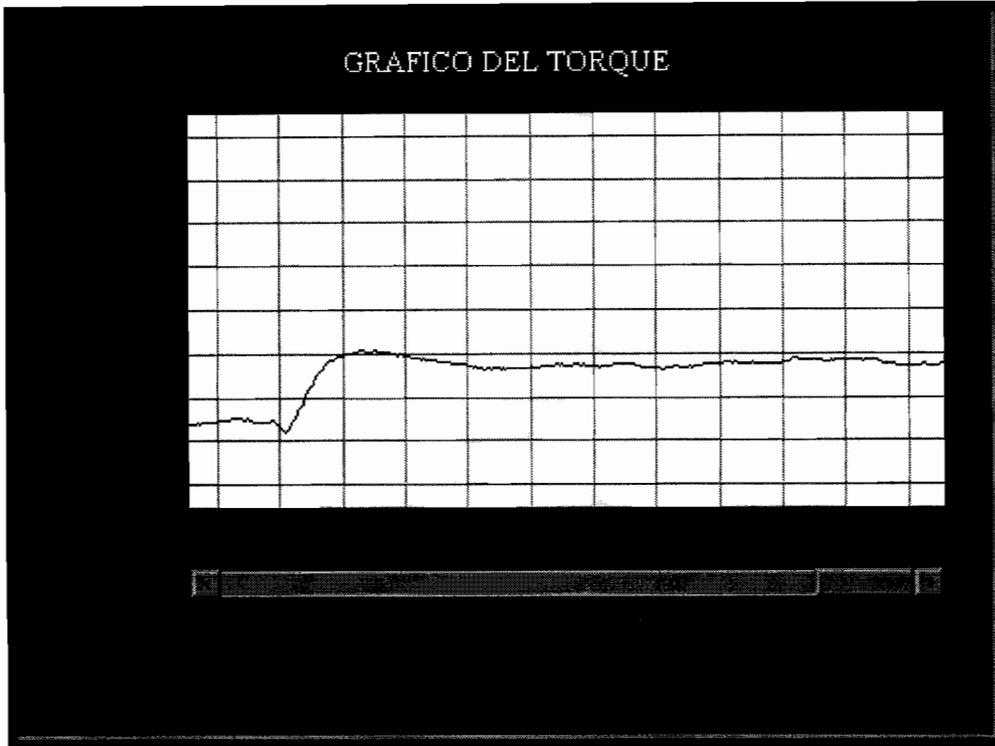
Una vez que el motor alcanzó su estado estable luego del arranque sin carga, se conectó la carga al generador acoplado al motor y se puso dos pasos de carga, obteniendo los siguientes datos:

<b>Lectura</b>	<b>Torque</b>
1.92 kgf	3.76 Nm

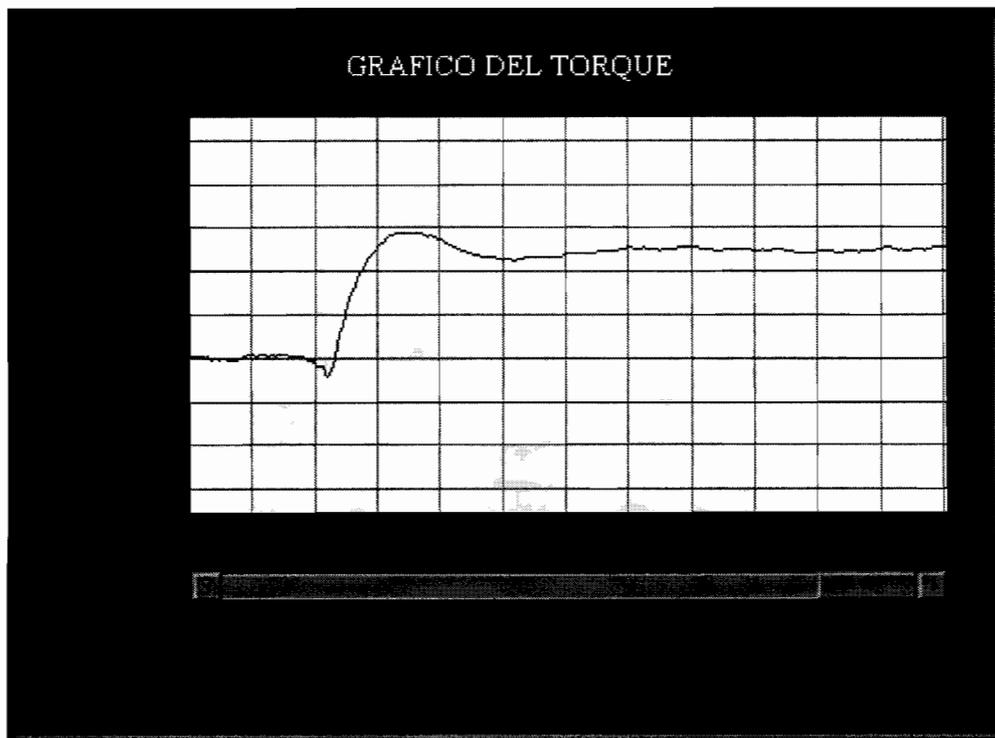
Para dos pasos adicionales de carga la lectura fue:

<b>Lectura</b>	<b>Torque</b>
4.22 kgf	8.27 Nm

El resultado obtenido con el DSK para distintos pasos de carga se presenta en las figuras 5.2 y 5.3.



**Fig. 5.2** Dos pasos de carga



**Fig. 5.3** Cuatro pasos de carga

### 5.1.3 ARRANQUE CON CARGA

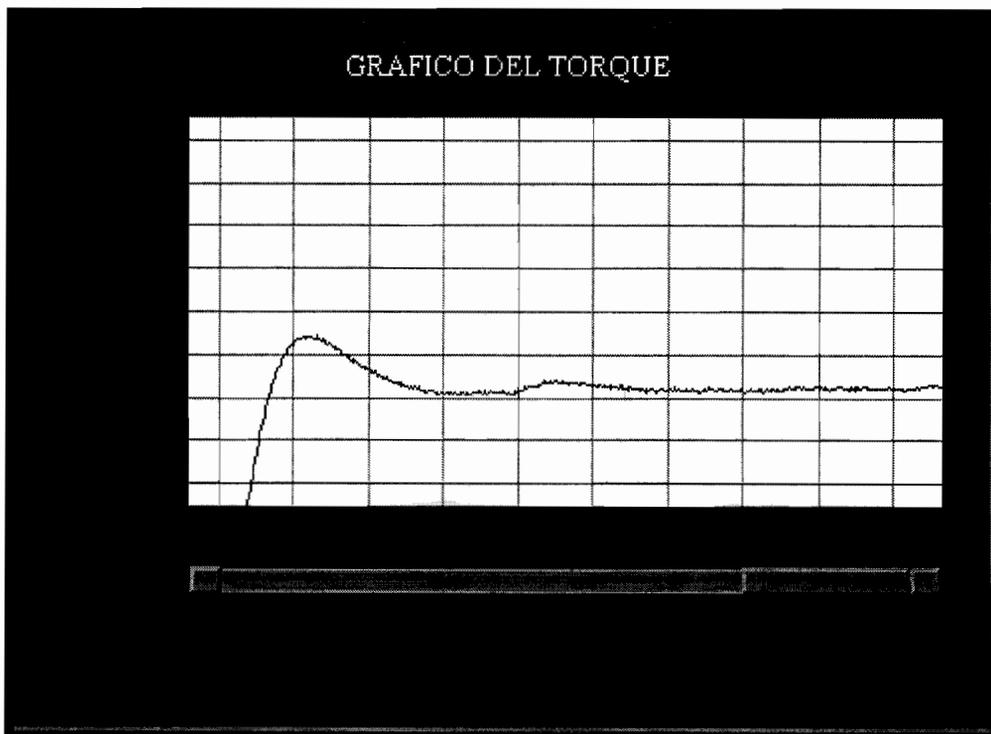
Se arranca el motor con dos pasos de carga y se obtiene la siguiente lectura:

Lectura	Torque
1.15 kgf	2.25 Nm

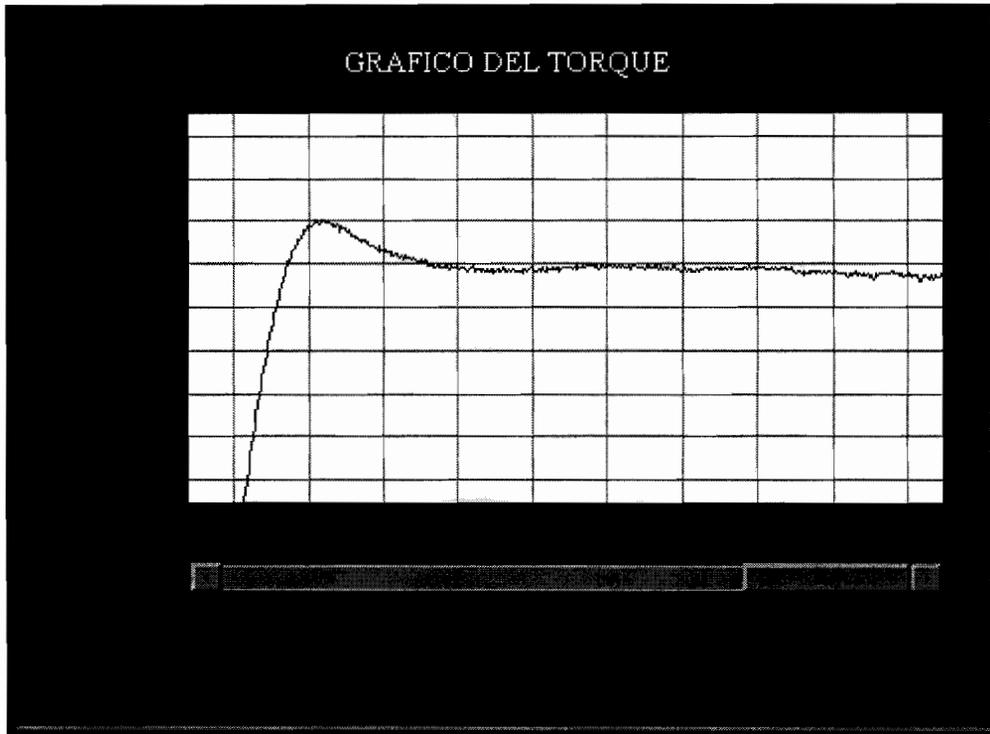
Para dos pasos adicionales de carga la lectura fue:

Lectura	Torque
3.8 kgf	7.45 Nm

El resultado obtenido con el DSK para distintos pasos de carga se presenta en las figuras 5.4 y 5.5.



**Fig. 5.4** Arranque con dos pasos de carga



**Fig. 5.5** Arranque con cuatro pasos de carga

#### 5.1.4 DISMINUCIÓN DE LA CARGA

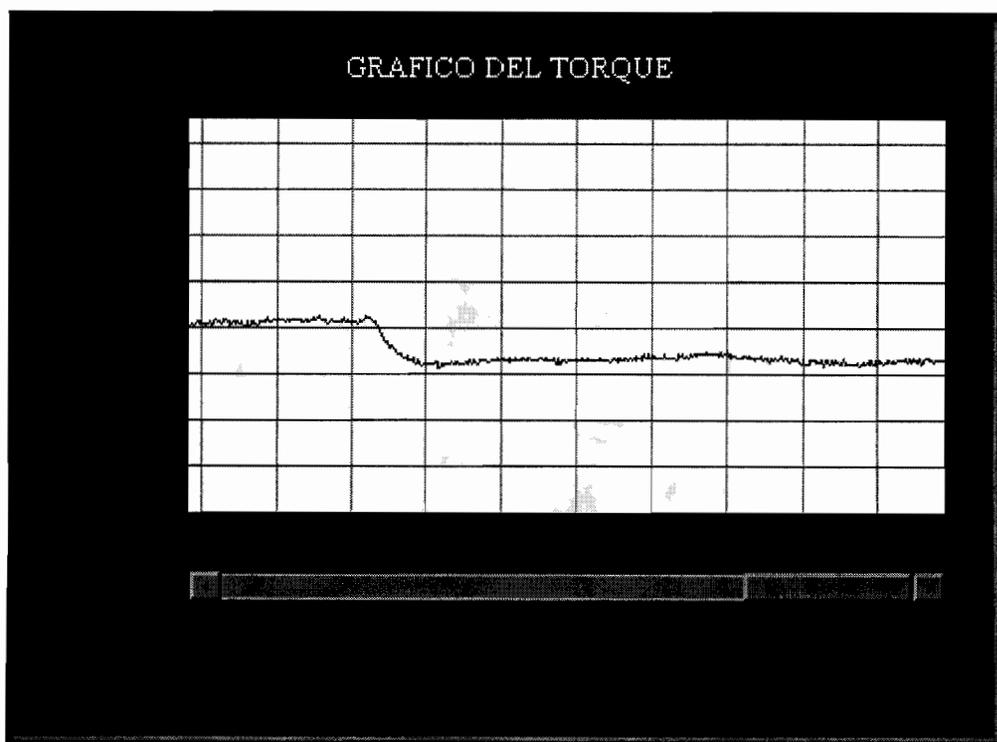
A continuación, estando el motor con cinco pasos de carga, se procede a quitar varios pasos de carga, obteniendo las siguientes lecturas:

	<b>Lectura</b>	<b>Torque</b>
menos 2 pasos de carga	3.15 kgf	6.17 Nm
menos 3 pasos	2.2 kgf	4.31 Nm
menos 5 pasos	0.28 kgf	0.55 Nm

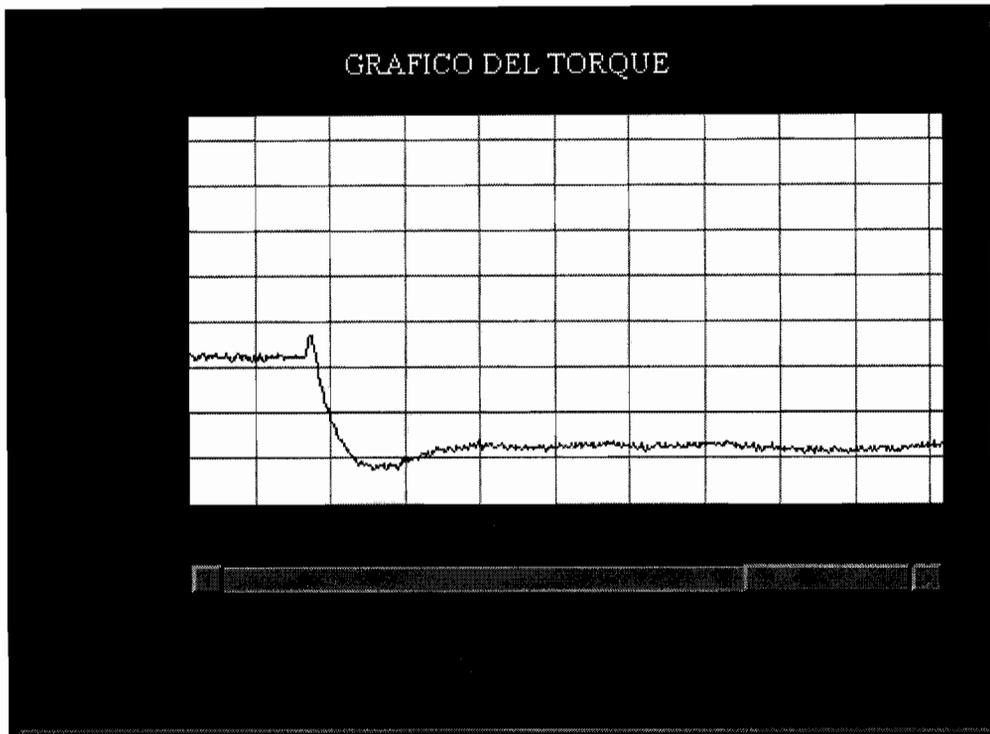
El resultado obtenido con el DSK para distintos pasos de carga se presenta en las figuras 5.6, 5.7 y 5.8.



**Fig. 5.6** Menos 2 pasos de carga



**Fig. 5.7** Menos 3 pasos de carga



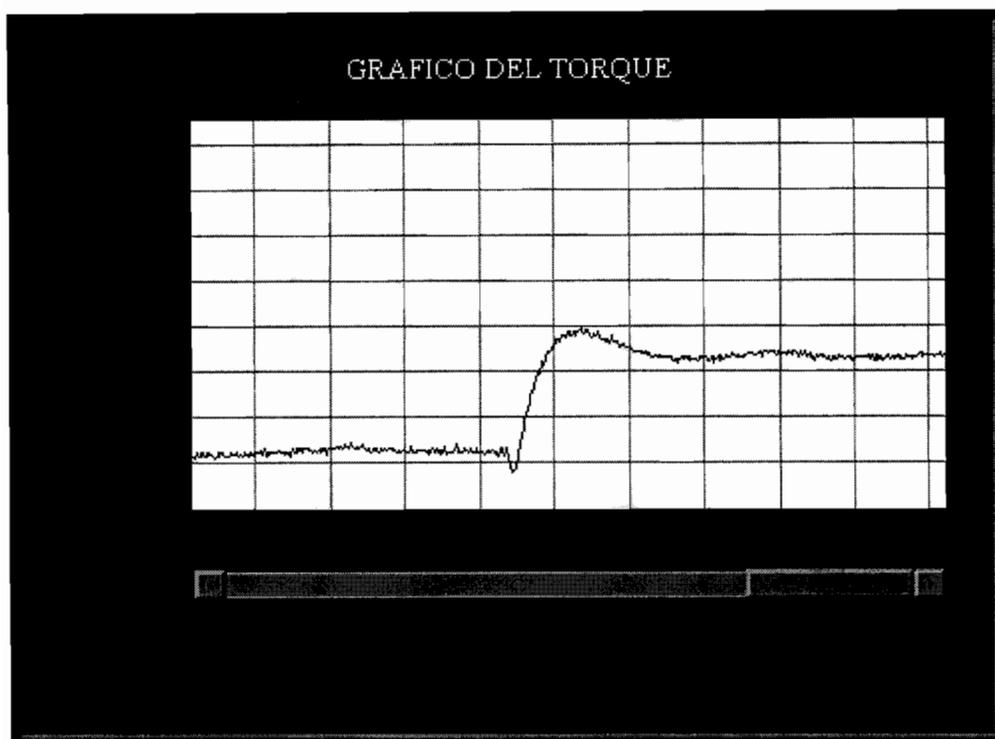
**Fig. 5.8** Menos 5 pasos de carga

### 5.1.5 AUMENTO DE LA CARGA

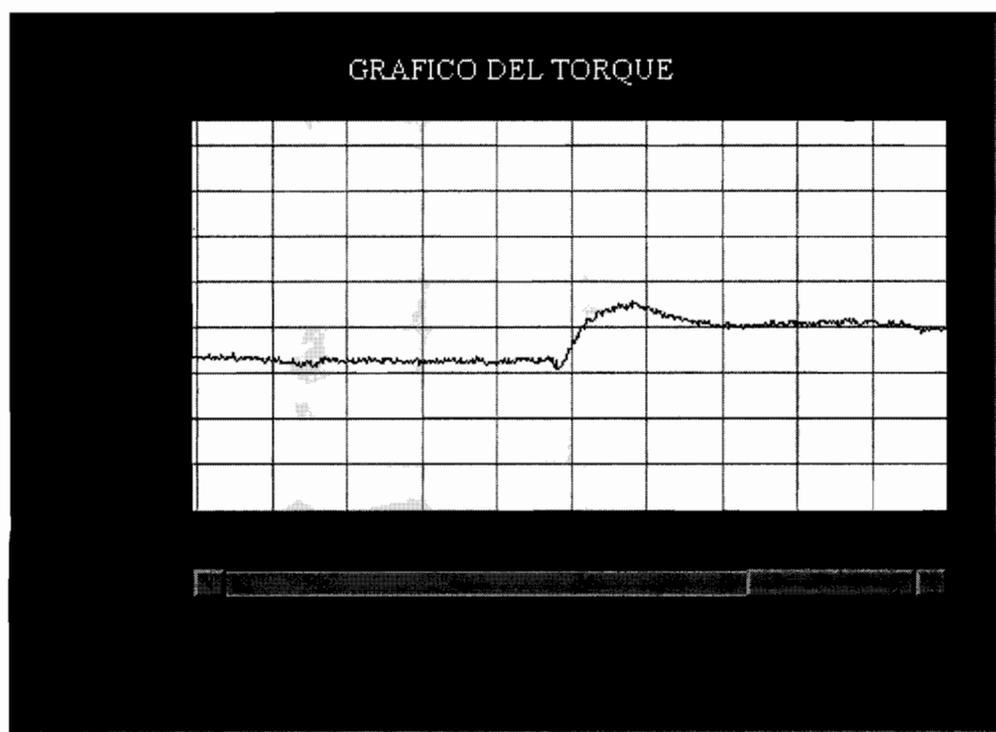
Se procede a aumentar distintos pasos de carga, para los cuales la lectura fue:

	<b>Lectura</b>	<b>Torque</b>
más 2 pasos de carga	2.3 kgf	4.51 Nm
más 4 pasos	3.2 kgf	6.27 Nm
más 5 pasos	4.8 kgf	9.41 Nm

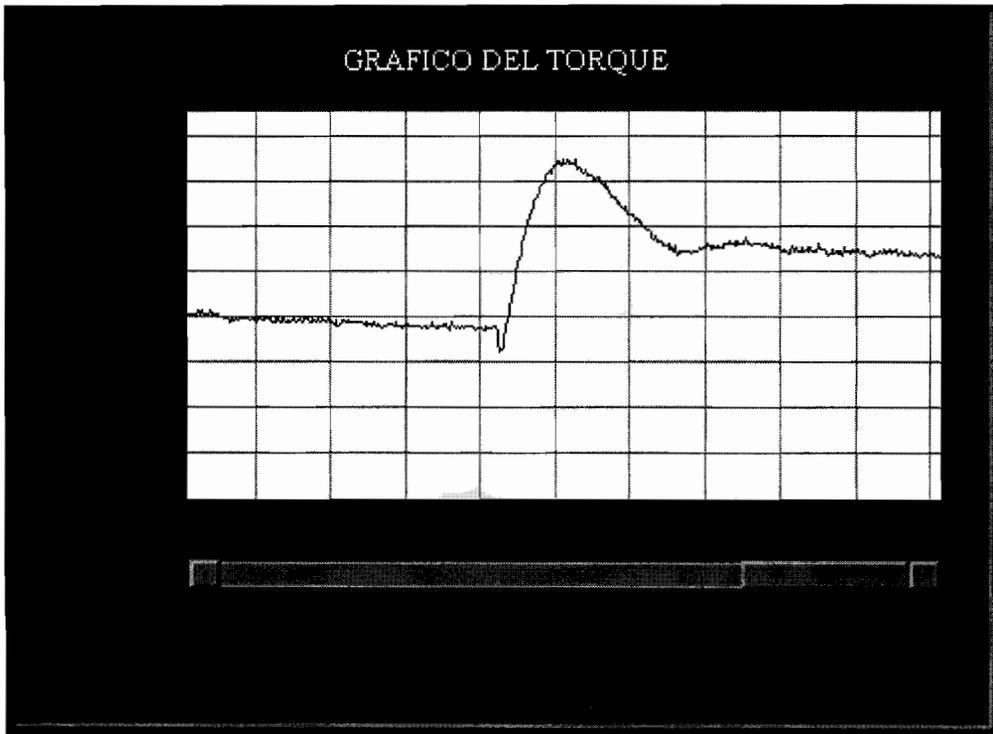
El resultado obtenido con el DSK para distintos pasos de carga se presenta en las figuras 5.9, 5.10 y 5.11.



**Fig. 5.9** Más dos pasos de carga



**Fig. 5.10** Más cuatro pasos de carga



**Fig. 5.11** Más cinco pasos de carga

### 5.1.6 CAMBIANDO PASOS DE CARGA

El resultado obtenido con el DSK se presenta en la figura 5.12.



**Fig. 5.12** Cambiando pasos de carga

## 5.2 CONCLUSIONES Y RECOMENDACIONES

### 5.2.1 CONCLUSIONES

En el presente trabajo se han descrito las principales características del módulo de evaluación TMS320C31 con el fin de realizar sobre el mismo la adquisición de las señales necesarias y el desarrollo del algoritmo que permitió obtener el Torque de una máquina de inducción trifásica. Para ello, se estudió la máquina en variables de estado, concluyendo con la modelación de la misma en un marco de referencia estacionario. Obteniéndose así las ecuaciones que rigen la misma tomando en consideración los parámetros del motor en estudio. Se puede concluir que al finalizar el presente trabajo, con el uso del Procesador Digital de Señales se puede simular casi en tiempo real el funcionamiento del motor y obtener variables como torque, corrientes y flujos, los mismos que pueden ser utilizados para el control de velocidad del motor trifásico de inducción.

Si bien, los principios de funcionamiento de los diferentes módulos de evaluación con DSPs, son los mismos; cada familia de estos dispositivos posee características individuales, tales como diferentes circuitos de interfaz análoga, números de bits, velocidad de procesamiento, etc, concluyéndose que el módulo TMS320C31, a más de presentar las ventajas explicadas en el capítulo 3, también presenta la desventaja de no poder hacer un muestreo de las dos señales al mismo tiempo.

Al implementar los circuitos para adquirir las señales, se tomó como consideraciones desacoplar las señales y el nivel de tensión que se introduce en los canales del AIC, observándose que es innecesario diseñar complejos circuitos de acondicionamiento, debido a que la señal de velocidad se la controla mediante el nivel de tensión del taco generador, y el nivel de voltaje con un transformador adecuadamente calculado, simplemente se eliminó el ruido obtenido en la señal de velocidad.

Una vez que ambas señales han sido adquiridas y acondicionadas vía software, se realiza el algoritmo del Torque, el cual, luego de introducir los parámetros del motor trifásico de inducción, calcula los demás términos involucrados en las ecuaciones de un marco de referencia estacionario, concluyéndose que para este estudio fue una buena solución el desarrollar en el algoritmo una subrutina de Runge-Kutta de cuarto orden, la cual es muy utilizada por algunos paquetes computacionales por su precisión frente a los demás métodos numéricos para el cálculo de la solución de ecuaciones diferenciales.

En el algoritmo, la subrutina de Runge-Kutta, puede realizar más iteraciones tomando los nuevos datos calculados como condiciones iniciales de una segunda iteración, sin embargo se observó que es más preciso volver a adquirir las señales para tener una confiabilidad mayor en el cálculo del Torque.

Se debe concluir además que a pesar de que el torque refleje su valor en por unidad en una localidad de memoria del DSP, a éste valor se lo debe acondicionar para que pueda ser transmitido como un valor de voltaje a través del canal analógico del conversor D/A, si no se lo acondicionara, los valores en punto flotante del torque en por unidad reflejan 0 V, debido a que son muy pequeños y el conversor D/A transmite números en punto flotante con tres dígitos decimales. Sin embargo al acondicionarlos, cada uno de los valores de torque, resulta un nivel que puede ser interpretado por la tarjeta del LabVIEW, el cual también necesita ser interpretado para ser presentado en la pantalla del computador, es necesario para esto configurar la entrada análoga de la tarjeta, esto es muy sencillo de realizar si se tiene presente que cada nivel de voltaje transmitido significa un nivel de torque, que se lo contrasta con el medidor de fuerza análogo. Sin embargo, no se logró encerrar correctamente el medidor de fuerza análogo que se dispone en el Laboratorio de Máquinas Eléctricas, especialmente en valores bajos de torque este medidor no es confiable, la razón se debe a que es un medidor de más de 20 años de funcionamiento.

Otra conclusión referida al funcionamiento del algoritmo, tiene que ver cuando se corre el programa cuando el motor está funcionando; se observa que la simulación empieza con condiciones iniciales de flujo cero y toma a la máquina como si empezara desde el arranque hasta alcanzar la estabilidad.

Luego de realizar las pruebas, se observan en los gráficos picos de torque, los mismos se deben básicamente a las corrientes, las cuales crecen o decrecen bruscamente al variar la carga antes de alcanzar su estabilidad.

### **5.2.2 RECOMENDACIONES**

Con el fin de tomar las dos señales en un mínimo tiempo posible, se recomienda no utilizar la versión original del programa assembler (dsktools), que viene disponible cuando se adquiere el módulo de evaluación, debido a que con éste, la adquisición de dos señales análogas resulta muy prolongada, por esto se utiliza una nueva versión del programa que incluye directivas que permiten que las dos señales se tomen en un tiempo menor que el de la versión anterior y que para esta aplicación fue de vital importancia.

Se debe tener mucho cuidado al energizar las entradas del conversor A/D del AIC con niveles mayores a los que puede soportar, además más peligroso es aún cuando el AIC no está energizado. Cuando esto ocurre, el circuito presenta un valor de impedancia por el que circula una indeterminada corriente que puede dañarlo.

Se recomienda, si se desea apreciar el torque máximo, variar el acondicionamiento vía software que se realiza para que el conversor D/A pueda transmitir valores que los pueda leer la tarjeta y variando nuevamente la configuración de la entrada análoga.

## BIBLIOGRAFÍA

### LIBROS

- a. Chapra Steven C. / Canale Raymond P. (1988): "Métodos Numéricos para Ingenieros", segunda edición, Editorial Mc Graw Hill / Interamericana de México, S.A. de C.V.
- b. Krause Paul C. (1996): "Analysis of Electric Machinery", Mc Graw Hill, México.
- c. National Instruments Corporation (1996): "LabVIEW User Manual", National Instruments Corporate Headquarters. Austin, Texas, 1996
- d. National Instruments Corporation (1996): "LabVIEW Tutorial Manual", National Instruments Corporate Headquarters. Austin, Texas, 1996
- e. Ong Chee-Mun (1998): "Dynamic Simulation of Electric Machinery using Matlab-Simulink, segunda edición, Prentice Hall, USA.
- f. Proakis John G. / Manolakis Dimitris G. (1998): "Tratamiento digital de señales – Principios, Algoritmos y Aplicaciones", tercera edición, Prentice Hall, Madrid.
- g. Stoer J. / Bulirsch R. (1980): "Introduction to Numerical Analysis", Springer-Verlag, New York.
- h. Texas Instruments Incorporated (1998): "Datasheet TLC32040M Analog Interface Circuit", Texas Instruments Incorporated, Dallas, Texas.
- i. Texas Instruments Incorporated (1999): "Understanding Data Converters Application Report", Texas Instruments Incorporated, Dallas, Texas.

- j. Texas Instruments Incorporated (1995): "Interfacing the TLC32040 Family to the TMS320 Family", Texas Instruments Incorporated, Dallas, Texas.
- k. Texas Instruments Incorporated (1999): "Datasheet TMS320C31 Digital Signal Processor", Texas Instruments Incorporated, Dallas, Texas.
- l. Texas Instruments Incorporated (1997): "TMS320C3x User's Guide", Texas Instruments Incorporated, Dallas, Texas.
- m. Texas Instruments Incorporated (1996): "TMS320C3x DSP Starter Kit User's Guide", Texas Instruments Incorporated, Dallas, Texas.
- n. Texas Instruments Incorporated (1998): "TMS320C3x/C4x Assembly Language Tools User's Guide", Texas Instruments Incorporated, Dallas, Texas.
- o. Texas Instruments Incorporated (1998): "TMS320C3x General Purpose Applications User's Guide", Texas Instruments Incorporated, Dallas, Texas.

## **INTERNET**

- p. Texas Instruments  
[www.ti.com](http://www.ti.com)

## REFERENCIAS

- [1] Rivera Argoti Pablo (2000), Folleto de Control de Máquinas Eléctricas, Facultad de Ingeniería Eléctrica, EPN, Quito-Ecuador
- [2] Krause Paul C. (1996): "Analysis of Electric Machinery", Mc Graw Hill, México, pág. 165-172.
- [3] Ong Chee-Mun (1998): "Dynamic Simulation of Electric Machinery using Matlab-Simulink, segunda edición, Prentice Hall, USA, pág. 169.
- [4] Ong Chee-Mun (1998): "Dynamic Simulation of Electric Machinery using Matlab-Simulink, segunda edición, Prentice Hall, USA, pág. 171-172
- [5] Krause Paul C. (1996): "Analysis of Electric Machinery", Mc Graw Hill, México, pág. 167-168.
- [6] Krause Paul C. (1996): "Analysis of Electric Machinery", Mc Graw Hill, México, pág. 174.
- [7] Pozo Acosta Mauricio (1989): Tesis "Modelo Digital de la Máquina Trifásica de Inducción con Rotor Bobinado, Jaula y Doble Jaula de Ardilla para Estudio Transitorio", EPN, pág. 160.

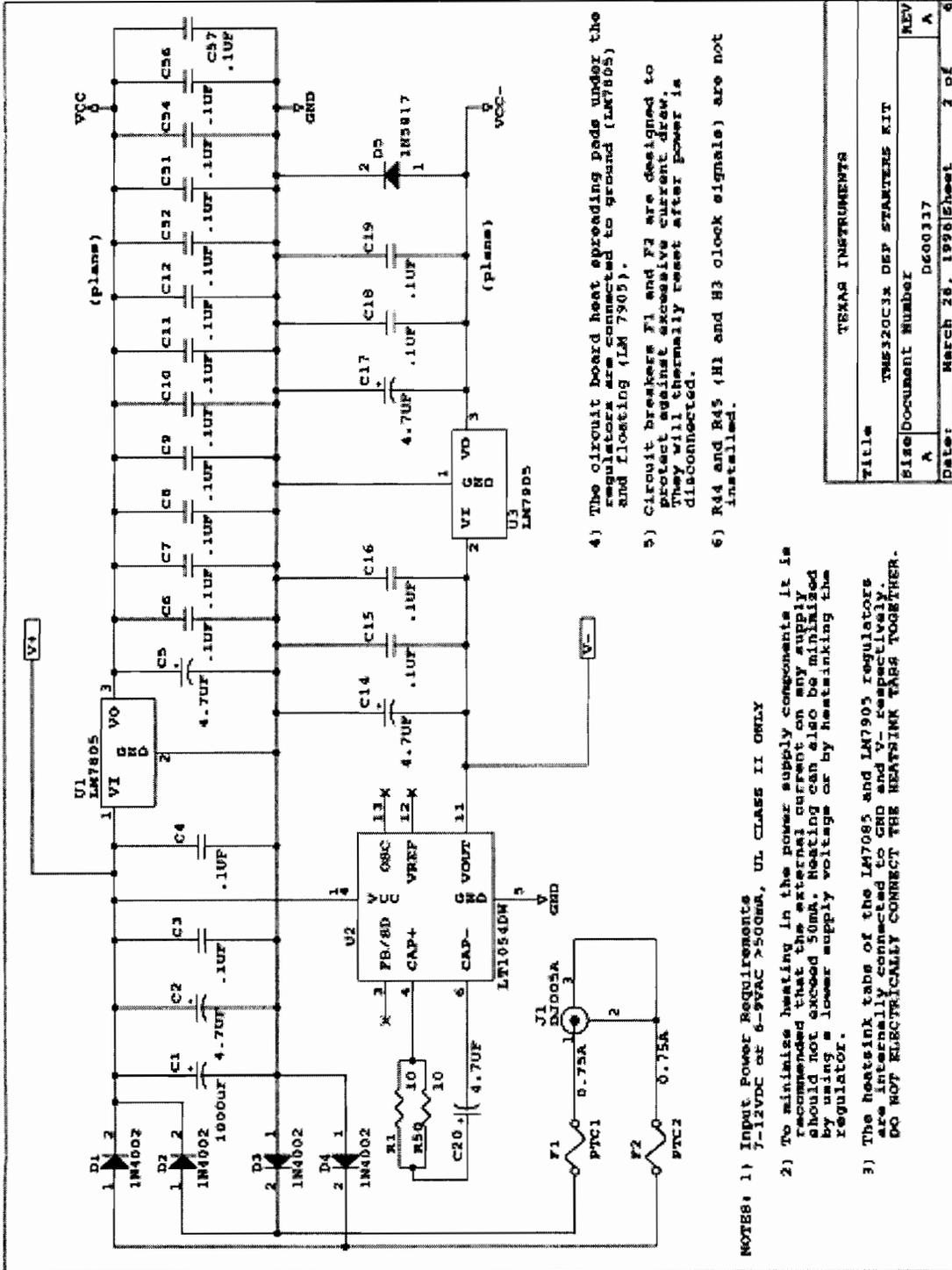
## **A Dimensiones de la Tarjeta DSK y Diagramas Esquemáticos**

---

La figura A.1 representa las dimensiones de la tarjeta DSK, el resto del anexo contiene los diagramas esquemáticos de la misma.







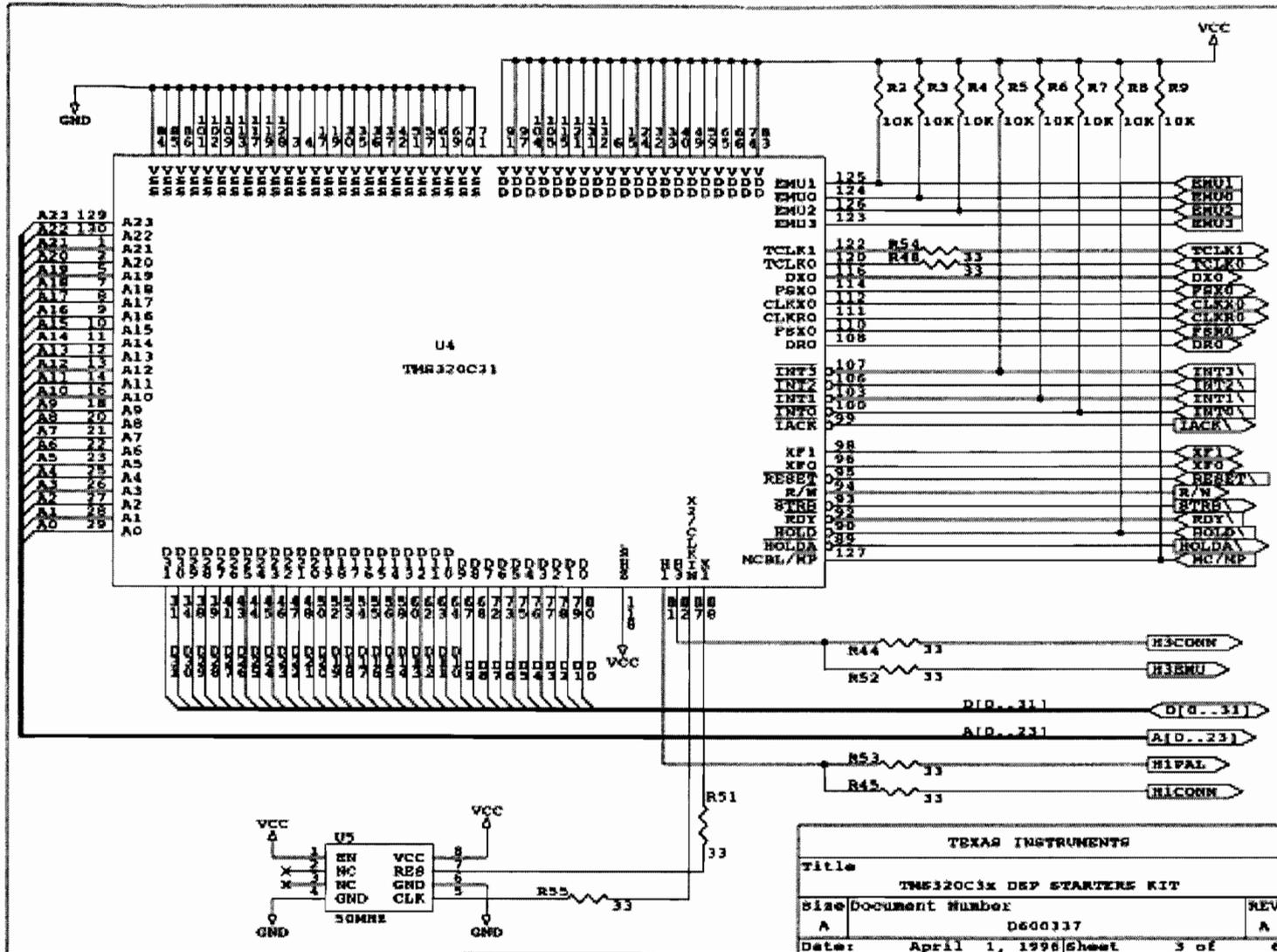
NOTES: 1) Input Power Requirements 7-12VDC at 8-5VAC >500mA, UL CLASS II ONLY

2) To minimize heating in the power supply components it is recommended that the external current on any supply should not exceed 50mA. Heating can also be minimized by using a lower supply voltage or by heat sinking the regulator.

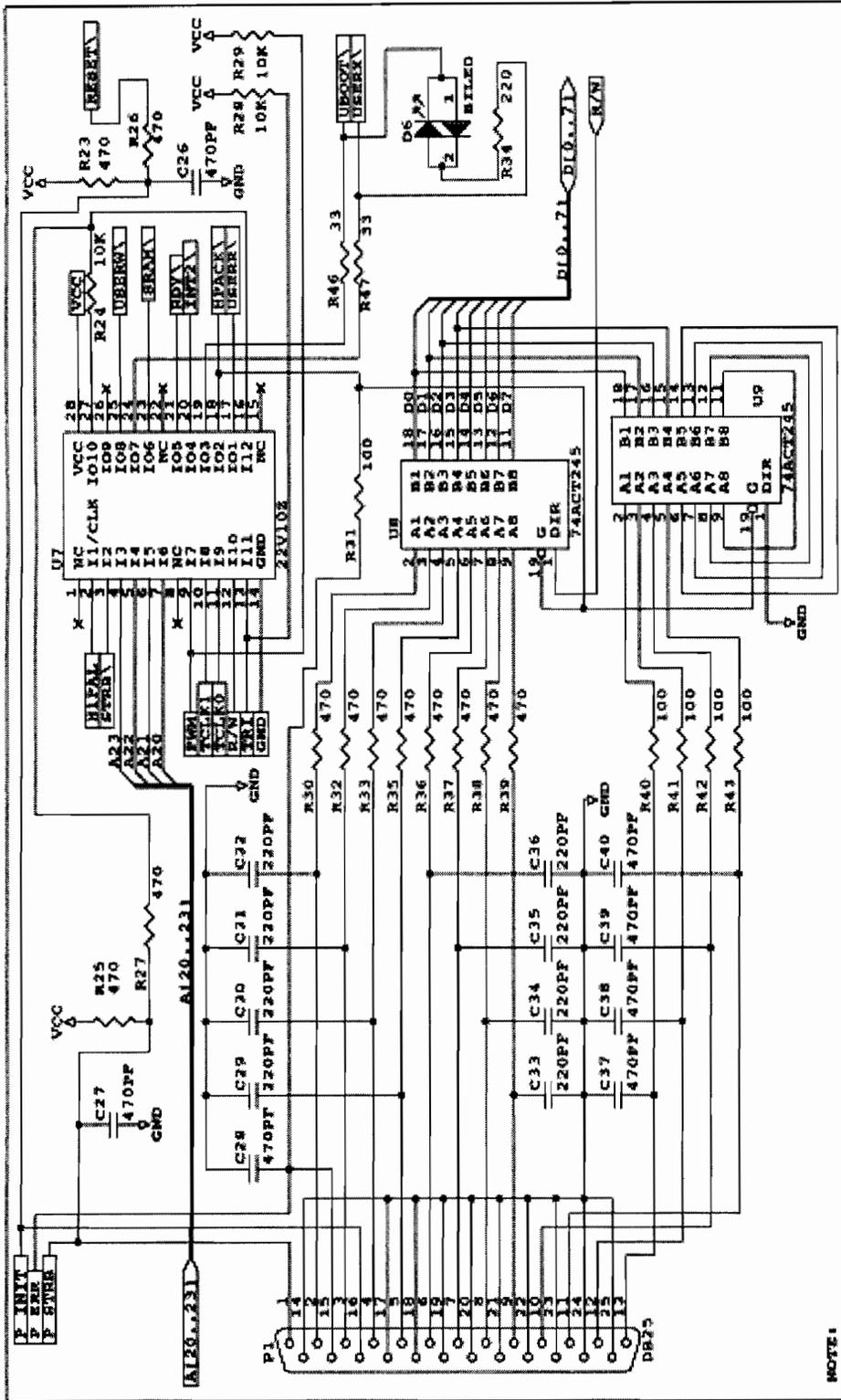
3) The heatsink tabs of the LM7805 and LM7905 regulators are internally connected to GND and V- respectively. DO NOT ELECTRICALLY CONNECT THE HEATSINK TABS TOGETHER.

- 4) The circuit board heat spreading pads under the regulators are connected to ground (LM7805) and floating (LM7905).
- 5) Circuit breakers F1 and F2 are designed to protect against excessive current draw. They will thermally reset after power is disconnected.
- 6) R44 and R45 (H1 and H3 clock signals) are not installed.

TEXAS INSTRUMENTS	
Title	TMS320C3X DEF STARTERS KIT
Size/Document Number	PC00317
REV	A
Date:	March 26, 1998 Sheet 2 of 6



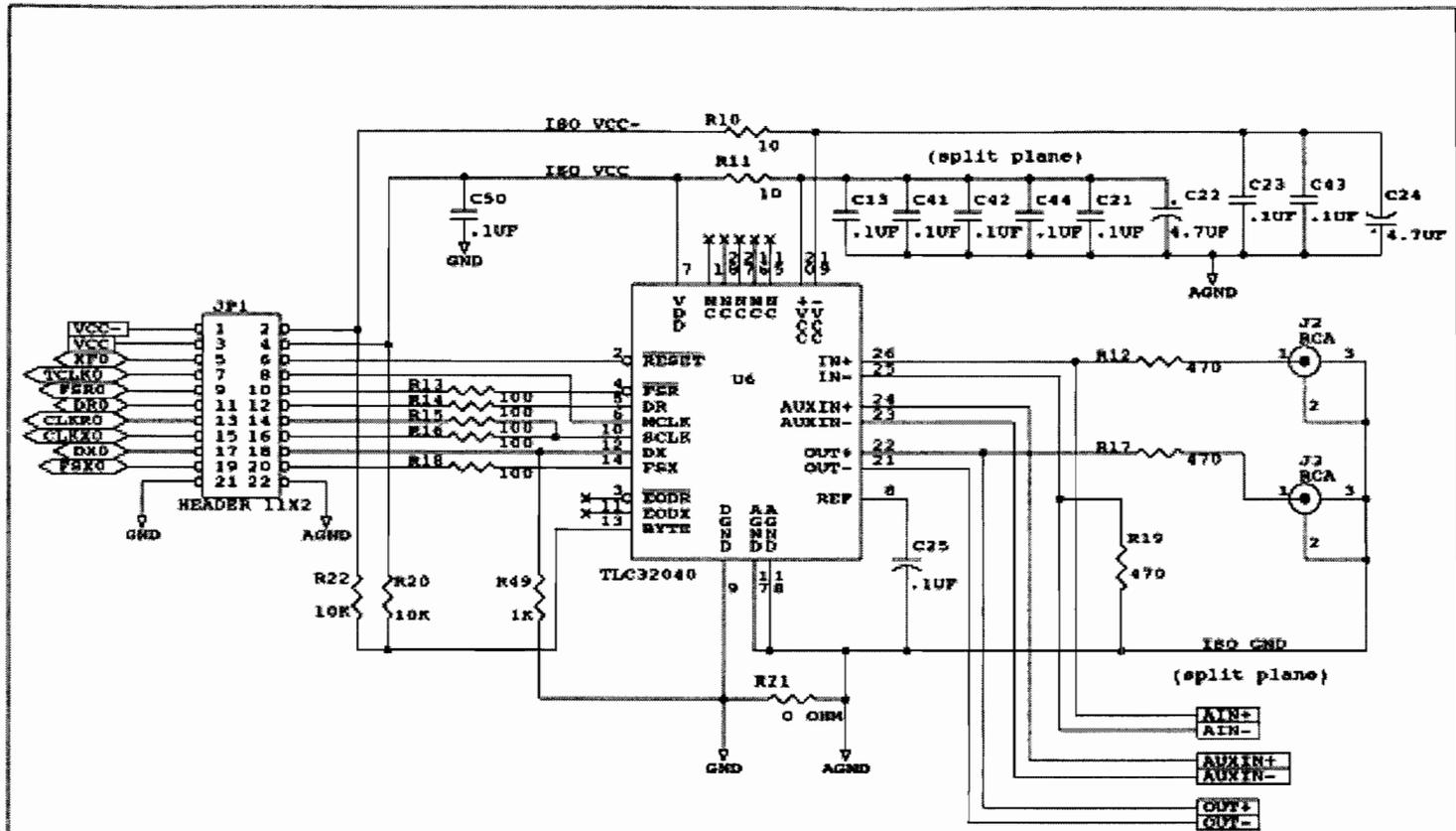
TEXAS INSTRUMENTS		
Title		
TMS320C31 DSP STARTER KIT		
Size	Document Number	REV
A	D600317	A
Date:	April 1, 1998	Sheet 3 of 6



NOTE:

By configuring TCLK0 and TCLK1 as clock outputs and driving the PWR high (default), the output pins UB00T\ and USERX\ drive the LED with a PWM signal. The PWM voltage is the phase difference between the two clocks and can be configured to be a DC output or a triangle wave (see DSP documentation).

TEXAS INSTRUMENTS	
Title	TMS320C3X DSP STARTERS KIT
Doc. Number	DS00237
Rev	A
Date	April 1, 1995
Sheet	4 of 6



Note:

Install R20 for TLC32040 or R22 for TLC32046

TEXAS INSTRUMENTS		
Title		
TMS320C1x DEP STARTERS KIT		
Size	Document Number	REV
A	D600337	A
Date:	April 1, 1998	Sheet 5 of 6



## **B Hojas de Datos del DSP TMS320C31**

---

Este anexo contiene las hojas de datos del Procesador Digital de Señales TMS320C31.

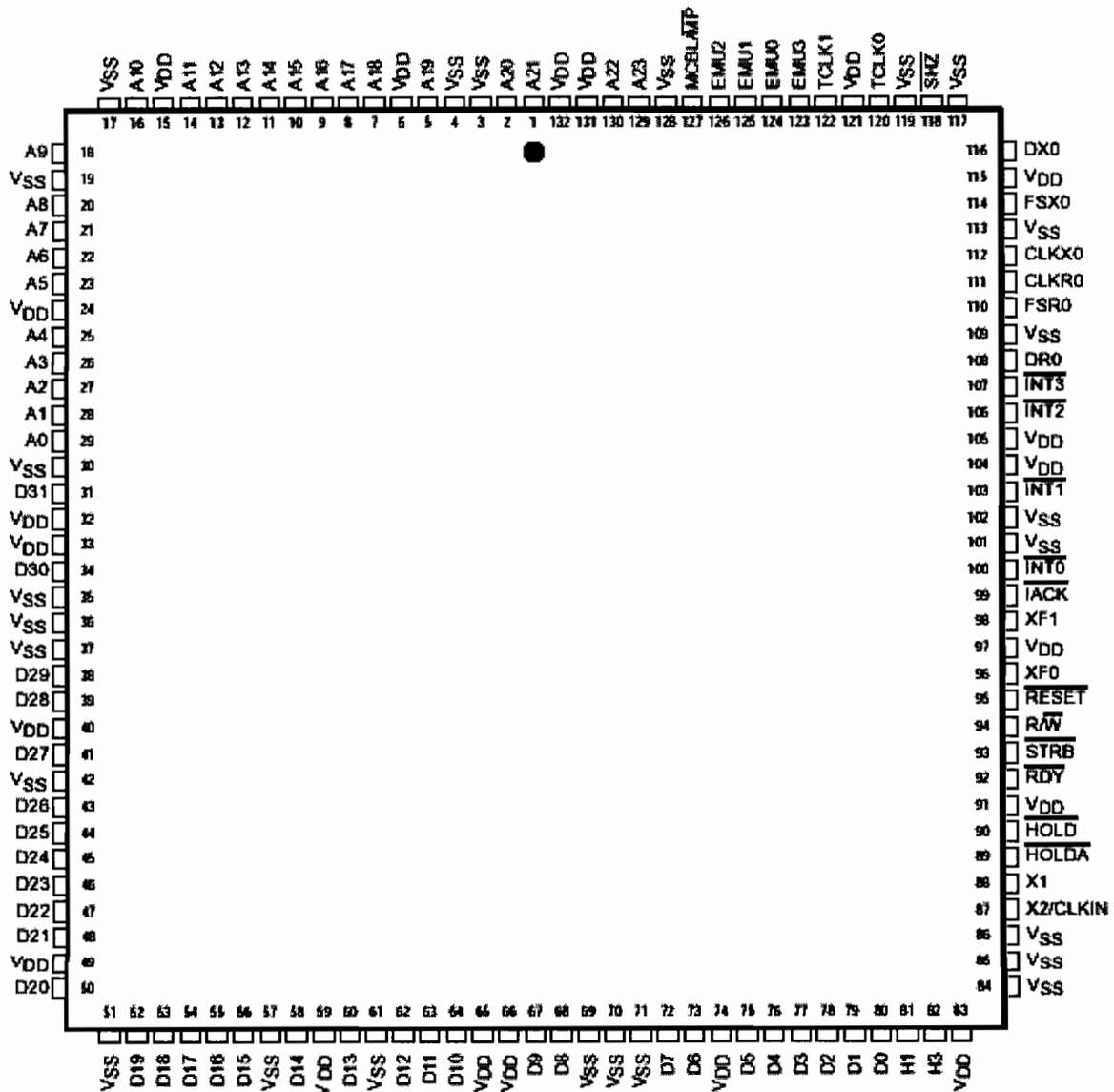
## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

### TMS320C31 and TMS320LC31 pinout (top view)

The TMS320C31 and TMS320LC31 devices are packaged in 132-pin plastic quad flatpacks (PQ Suffix).

#### PQ PACKAGE (TOP VIEW)



## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1986 – REVISED JANUARY 1999

TMS320C31 and TMS320LC31 Terminal Assignments (Alphabetical)<sup>†</sup>

TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.	TERMINAL NAME	NO.
A0	29	D4	76	EMU0	124	VDD	40	VSS	84
A1	28	D5	75	EMU1	125	VDD	49	VSS	85
A2	27	D6	73	EMU2	126	VDD	59	VSS	88
A3	26	D7	72	EMU3	123	VDD	65	VSS	101
A4	25	D8	68	FSR0	110	VDD	66	VSS	102
A5	23	D9	67	FSX0	114	VDD	74	VSS	109
A6	22	D10	64	H1	81	VDD	83	VSS	113
A7	21	D11	63	H3	82	VDD	91	VSS	117
A8	20	D12	62	$\overline{\text{HOLD}}$	90	VDD	97	VSS	119
A9	18	D13	60	$\overline{\text{HOLDA}}$	89	VDD	104	VSS	128
A10	16	D14	58	$\overline{\text{IACK}}$	99	VDD	105	X1	88
A11	14	D15	56	$\overline{\text{INT0}}$	100	VDD	115	X2/CLKIN	87
A12	13	D16	55	$\overline{\text{INT1}}$	103	VDD	121	XF0	96
A13	12	D17	54	$\overline{\text{INT2}}$	106	VDD	131	XF1	98
A14	11	D18	53	$\overline{\text{INT3}}$	107	VDD	132		
A15	10	D19	52	$\overline{\text{MCBLMP}}$	127	VSS	3		
A16	9	D20	50	$\overline{\text{RDY}}$	92	VSS	4		
A17	8	D21	48	$\overline{\text{RESET}}$	95	VSS	17		
A18	7	D22	47	$\overline{\text{RAW}}$	94	VSS	19		
A19	5	D23	46	$\overline{\text{SHZ}}$	118	VSS	30		
A20	2	D24	45	$\overline{\text{STRB}}$	93	VSS	35		
A21	1	D25	44	TCLK0	120	VSS	36		
A22	130	D26	43	TCLK1	122	VSS	37		
A23	129	D27	41			VSS	42		
CLKR0	111	D28	39			VSS	51		
CLKX0	112	D29	38	VDD	6	VSS	57		
D0	80	D30	34	VDD	15	VSS	61		
D1	79	D31	31	VDD	24	VSS	69		
D2	78	DR0	108	VDD	32	VSS	70		
D3	77	DX0	116	VDD	33	VSS	71		

<sup>†</sup> VDD and VSS pins are on a common plane internal to the device.

# TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1986 – REVISED JANUARY 1999

## TMS320C31 and TMS320LC31 Terminal Functions

TERMINAL NAME	QTY	TYPE†	DESCRIPTION	CONDITIONS WHEN SIGNAL IS Z TYPE‡
<b>PRIMARY-BUS INTERFACE</b>				
D31–D0	32	I/O/Z	32-bit data port	S H R
A23–A0	24	O/Z	24-bit address port	S H R
R/W	1	O/Z	Read/write. R/W is high when a read is performed and low when a write is performed over the parallel interface.	S H R
STRB	1	O/Z	External-access strobe	S H
RDY	1	I	Ready. RDY indicates that the external device is prepared for a transaction completion.	
HOLD	1	I	Hold. When HOLD is a logic low, any ongoing transaction is completed. A23–A0, D31–D0, STRB, and R/W are placed in the high-impedance state and all transactions over the primary-bus interface are held until HOLD becomes a logic high or until the NOHOLD bit of the primary-bus-control register is set.	
HOLDA	1	O/Z	Hold acknowledge. HOLDA is generated in response to a logic low on HOLD. HOLDA indicates that A23–A0, D31–D0, STRB, and R/W are in the high-impedance state and that all transactions over the bus are held. HOLDA is high in response to a logic high of HOLD or the NOHOLD bit of the primary-bus-control register is set.	S
<b>CONTROL SIGNALS</b>				
RESET	1	I	Reset. When RESET is a logic low, the device is in the reset condition. When RESET becomes a logic high, execution begins from the location specified by the reset vector.	
INT3–INT0	4	I	External interrupts	
IACK	1	O/Z	Interrupt acknowledge. IACK is generated by the IACK instruction. IACK can be used to indicate the beginning or the end of an interrupt-service routine.	S
MCBL/MP	1	I	Microcomputer boot-loader/microprocessor mode-select	
SHZ	1	I	Shutdown high impedance. When active, SHZ shuts down the device and places all pins in the high-impedance state. SHZ is used for board-level testing to ensure that no dual-drive conditions occur. CAUTION: A low on SHZ corrupts the device memory and register contents. Reset the device with SHZ high to restore it to a known operating condition.	
XF1, XF0	2	I/O/Z	External flags. XF1 and XF0 are used as general-purpose I/Os or to support interlocked processor instruction.	S R
<b>SERIAL PORT 0 SIGNALS</b>				
CLKR0	1	I/O/Z	Serial port 0 receive clock. CLKR0 is the serial shift clock for the serial port 0 receiver.	S R
CLKX0	1	I/O/Z	Serial port 0 transmit clock. CLKX0 is the serial shift clock for the serial port 0 transmitter.	S R
DR0	1	I/O/Z	Data-receive. Serial port 0 receives serial data on DR0.	S R
DX0	1	I/O/Z	Data-transmit output. Serial port 0 transmits serial data on DX0.	S R
FSR0	1	I/O/Z	Frame-synchronization pulse for receive. The FSR0 pulse initiates the data-receive process using DR0.	S R
FSX0	1	I/O/Z	Frame-synchronization pulse for transmit. The FSX0 pulse initiates the data-transmit process using DX0.	S R
<b>TIMER SIGNALS</b>				
TCLK0	1	I/O/Z	Timer clock 0. As an input, TCLK0 is used by timer 0 to count external pulses. As an output, TCLK0 outputs pulses generated by timer 0.	S R
TCLK1	1	I/O/Z	Timer clock 1. As an input, TCLK1 is used by timer 1 to count external pulses. As an output, TCLK1 outputs pulses generated by timer 1.	S R

† I = input, O = output, Z = high-impedance state

‡ S = SHZ active, H = HOLD active, R = RESET active

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1986 – REVISED JANUARY 1999

### TMS320C31 and TMS320LC31 Terminal Functions (Continued)

TERMINAL NAME	QTY	TYPE†	DESCRIPTION	CONDITIONS WHEN SIGNAL IS Z TYPE‡
<b>SUPPLY AND OSCILLATOR SIGNALS</b>				
H1	1	O/Z	External H1 clock. H1 has a period equal to twice CLKIN.	S
H3	1	O/Z	External H3 clock. H3 has a period equal to twice CLKIN.	S
VDD	20	I	5-V supply for 'C31 devices and 3.3-V supply for 'LC31 devices. All must be connected to a common supply plane.§	
VSS	25	I	Ground. All grounds must be connected to a common ground plane.	
X1	1	O	Output from the internal-crystal oscillator. If a crystal is not used, X1 should be left unconnected.	
X2/CLKIN	1	I	Internal-oscillator input from a crystal or a clock	
<b>RESERVED¶</b>				
EMU2-EMU0	3	I	Reserved for emulation. Use pullup resistors to VDD	
EMU3	1	O/Z	Reserved for emulation	S

† I = input, O = output, Z = high-impedance state

‡ S = SHZ active, H = HOLD active, R = RESET active

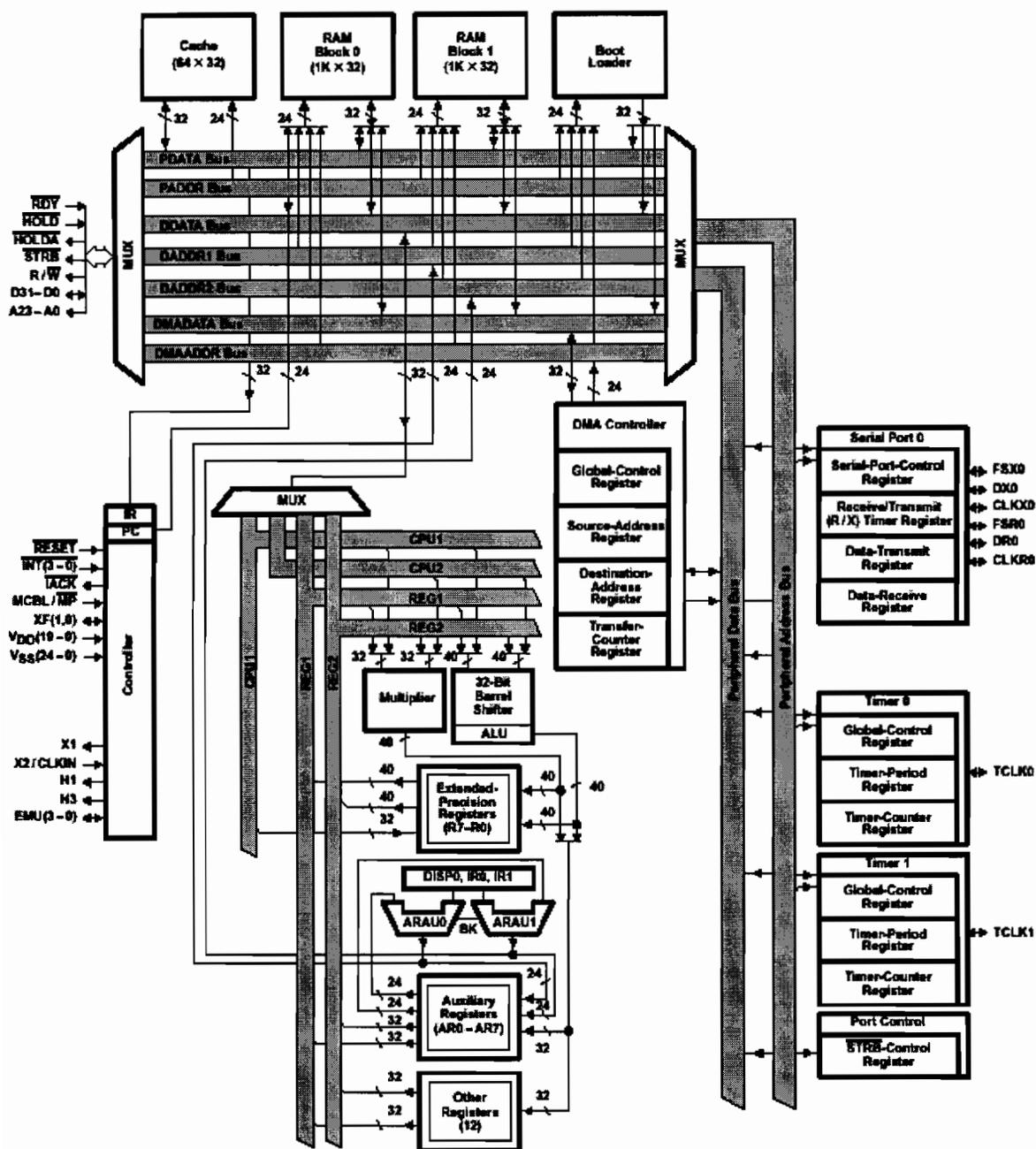
§ Recommended decoupling capacitor value is 0.1  $\mu$ F.¶ Follow the connections specified for the reserved pins. Use 18-k $\Omega$ –22-k $\Omega$  pullup resistors for best results. All VDD supply pins must be connected to a common supply plane, and all ground pins must be connected to a common ground plane.

- NOTES:
1. A test mode for measuring leakage currents in the TMS320C31 is implemented. This test mode powers down the clock oscillator circuit resulting in currents below 10  $\mu$ A. The test mode is entered by asserting SHZ low, which tri-states all output pins and then holds both H1 and H3 at logic high. The test mode is not intended for application use because it does not preserve the processor state.
  2. Since SHZ is a synchronized input and the clock is disabled, exiting the test mode occurs only when at least one of the H1/H3 pins is pulled low. Reset cannot be used to wake up in test mode since the SHZ pin is sampled and the clocks are not running.
  3. On power up, the processor can be in an indeterminate state. If the state is SHZ mode and H1 and H3 are both held logic high by pull-ups, then shutdown will occur. Normally, if H1 and H3 do not have pull-ups, the rise time lag due to capacitive loading on a tri-state pin is enough to ensure a clean start. However, a slowly rising supply and board leakages to VCC may be enough to cause a bad start. Therefore, a pulldown resistor on either H1 or H3 is recommended for proper wakeup.

# TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B - MARCH 1986 - REVISED JANUARY 1999

functional block diagram



# TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

## memory map

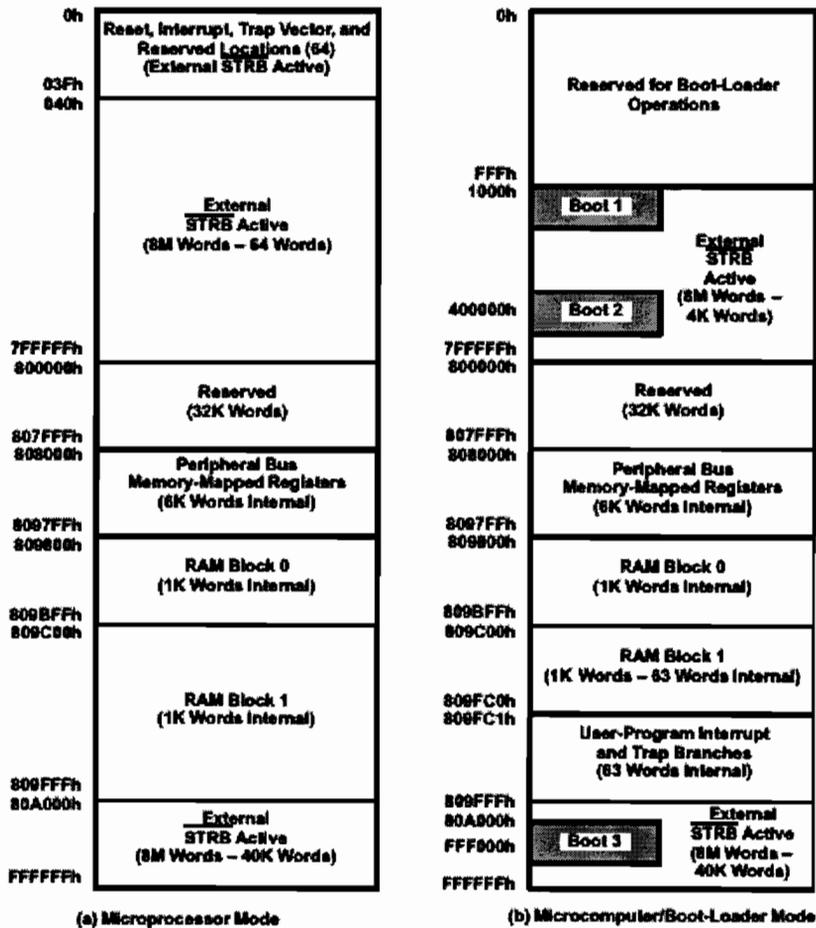


Figure 1. TMS320C31 Memory Maps

# TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

## memory map (continued)

00h	Reset	809FC1h	INT0
01h	INT0	809FC2h	INT1
02h	INT1	809FC3h	INT2
03h	INT2	809FC4h	INT3
04h	INT3	809FC5h	XINT0
05h	XINT0	809FC6h	RINT0
06h	RINT0	809FC7h	Reserved
07h	Reserved	809FC8h	Reserved
08h	Reserved	809FC9h	TINT0
09h	TINT0	809FCAh	TINT1
0Ah	TINT1	809FCBh	DINT
0Bh	DINT	809FCCh	Reserved
0Ch	Reserved	809FDFh	Reserved
1Fh	Reserved	809FE0h	TRAP 0
20h	TRAP 0		•
	•		•
	•		•
3Bh	TRAP 27	809FFBh	TRAP 27
3Ch	Reserved	809FFCh	Reserved
3Fh	Reserved	809FFFh	Reserved

(a) Microprocessor Mode

(b) Microcomputer/Boot-Loader Mode

Figure 2. Reset, Interrupt, and Trap Vector/Branches Memory-Map Locations

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

### memory map (continued)

808000h	DMA Global Control
808004h	DMA Source Address
808006h	DMA Destination Address
808008h	DMA Transfer Counter
808020h	Timer 0 Global Control
808024h	Timer 0 Counter
808028h	Timer 0 Period Register
808030h	Timer 1 Global Control
808034h	Timer 1 Counter
808038h	Timer 1 Period Register
808040h	Serial Global Control
808042h	FSX/DX/CLKX Serial Port Control
808043h	FSR/DR/CLKR Serial Port Control
808044h	Serial R/X Timer Control
808045h	Serial R/X Timer Counter
808046h	Serial R/X Timer Period Register
808048h	Data-Transmit
80804Ch	Data-Receive
808064h	Primary-Bus Control

†Shading denotes reserved address locations

**Figure 3. Peripheral Bus Memory-Mapped Registers†**

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

### absolute maximum ratings over specified temperature range (unless otherwise noted)†

	'C31	'LC31
Supply voltage range, $V_{DD}$ (see Note 1)	–0.3 V to 7 V	–0.3 V to 5 V
Input voltage range, $V_I$	–0.3 V to 7 V	–0.3 V to 5 V
Output voltage range, $V_O$	–0.3 V to 7 V	–0.3 V to 5 V
Continuous power dissipation (worst case) (see Note 5)	2.6 W (for TMS320C31-80)	850 mW (for TMS320LC31-33)
Operating case temperature range, $T_C$	PQL (commercial)	0°C to 85°C
	PQA (industrial)	–40°C to 125°C
Storage temperature range, $T_{stg}$	–55°C to 150°C	–55°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 4. All voltage values are with respect to  $V_{SS}$ .

5. Actual operating power is less. This value was obtained under specially produced worst-case test conditions for the TMS320C31, which are not sustained during normal device operation. These conditions consist of continuous parallel writes of a checkerboard pattern to both primary and extension buses at the maximum rate possible. See normal ( $I_{CC}$ ) current specification in the electrical characteristics table and also read *Calculation of TMS320C30 Power Dissipation Application Report* (literature number SPRA020).

### recommended operating conditions (see Note 6)

	'C31			'LC31			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$V_{DD}$ Supply voltage ( $DV_{DD}$ , etc.)	4.75	5	5.25	3.13	3.3	3.47	V
$V_{SS}$ Supply voltage ( $CV_{SS}$ , etc.)	0			0			V
$V_{IH}$ High-level input voltage	2	$V_{DD} + 0.3\ddagger$		1.8	$V_{DD} + 0.3\ddagger$		V
$V_{IL}$ Low-level input voltage	–0.3 $\ddagger$	0.8		–0.3 $\ddagger$	0.6		V
$I_{OH}$ High-level output current	–300			–300			$\mu$ A
$I_{OL}$ Low-level output current	2			2			mA
$T_C$ Operating case temperature (commercial)	0	85		0	85		°C
	–40	125					°C
$V_{TH}$ High-level input voltage for CLKIN	2.6	$V_{DD} + 0.3\ddagger$		2.5	$V_{DD} + 0.3\ddagger$		V

‡ These values are derived from characterization and not tested.

NOTE 6: All voltage values are with respect to  $V_{SS}$ . All input and output voltage levels are TTL-compatible. CLKIN can be driven by a CMOS clock.

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

electrical characteristics over recommended ranges of supply voltage (unless otherwise noted)  
(see Note 3)<sup>†</sup>

PARAMETER	TEST CONDITIONS	'C31			'LC31			UNIT	
		MIN	TYP <sup>‡</sup>	MAX	MIN	TYP <sup>‡</sup>	MAX		
V <sub>OH</sub> High-level output voltage	V <sub>DD</sub> = MIN. I <sub>OH</sub> = MAX	2.4	3		2			V	
V <sub>OL</sub> Low-level output voltage	V <sub>DD</sub> = MIN. I <sub>OH</sub> = MAX		0.3	0.6		0.4		V	
I <sub>Z</sub> High-impedance current	V <sub>DD</sub> = MAX	-20		+20	-20		+20	μA	
I <sub>I</sub> Input current	V <sub>I</sub> = V <sub>SS</sub> to V <sub>DD</sub>	-10		+10	-10		+10	μA	
I <sub>IP</sub> Input current (with internal pullup)	Inputs with internal pullups <sup>§</sup>	-600		20	-600		10	μA	
I <sub>CC</sub> Supply current <sup>¶</sup>	T <sub>A</sub> = 25°C, V <sub>DD</sub> = MAX	f <sub>x</sub> = 33 MHz	'C31-33		150	325	120	250	mA
		f <sub>x</sub> = 33 MHz	'C31-33 (ext. temp)		150	325			
		f <sub>x</sub> = 40 MHz	'C31-40		160	390	150	300	
		f <sub>x</sub> = 50 MHz	'C31-50		200	425			
		f <sub>x</sub> = 60 MHz	'C31-60		225	475			
		f <sub>x</sub> = 80 MHz	'C31-80		275	550			
I <sub>DD</sub> Supply current	Standby, IDLE2 Clocks shut off		50			20		μA	
C <sub>i</sub> Input capacitance	All inputs except CLKIN			15			15	pF	
	CLKIN			25			25		
C <sub>O</sub> Output capacitance				20			20	pF	

<sup>†</sup> All input and output voltage levels are TTL compatible.

<sup>‡</sup> For 'C31, all typical values are at V<sub>DD</sub> = 5 V, T<sub>A</sub> (air temperature) = 25°C. For 'LC31, all typical values are at V<sub>DD</sub> = 3.3 V, T<sub>A</sub> (air temperature) = 25°C.

<sup>§</sup> Pins with internal pullup devices: INT3–INT0, MCBL/MP.

<sup>¶</sup> Actual operating current is less than this maximum value. This value was obtained under specially produced worst-case test conditions, which are not sustained during normal device operation. These conditions consist of continuous parallel writes of a checkerboard pattern at the maximum rate possible. See *Calculation of TMS320C30 Power Dissipation Application Report* (literature number SPRA020).

<sup>#</sup> f<sub>x</sub> is the input clock frequency.

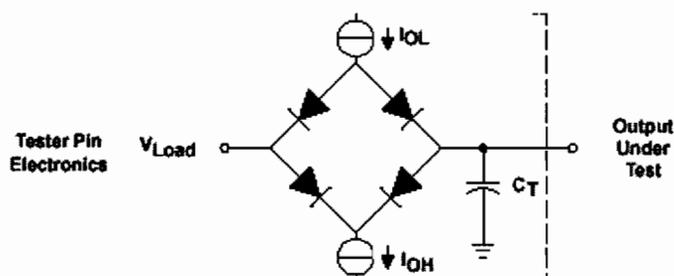
|| Specified by design but not tested

NOTE 6: All voltage values are with respect to V<sub>SS</sub>. All input and output voltage levels are TTL-compatible. CLKIN can be driven by a CMOS clock.

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

### PARAMETER MEASUREMENT INFORMATION



Where:  $I_{OL}$  = 2 mA (all outputs)  
 $I_{OH}$  = 300  $\mu$ A (all outputs)  
 $V_{LOAD}$  = 2.15 V  
 $C_T$  = 80-pF typical load-circuit capacitance

Figure 4. TMS320C31 Test Load Circuit

### signal transition levels for 'C31 (see Figure 5 and Figure 6)

TTL-level outputs are driven to a minimum logic-high level of 2.4 V and to a maximum logic-low level of 0.6 V. Output transition times are specified as follows:

- For a high-to-low transition on a TTL-compatible output signal, the level at which the output is said to be no longer high is 2 V and the level at which the output is said to be low is 1 V.
- For a low-to-high transition, the level at which the output is said to be no longer low is 1 V and the level at which the output is said to be high is 2 V.

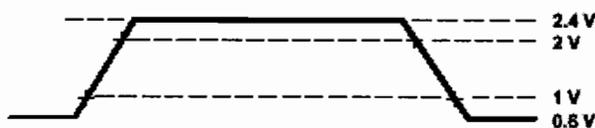


Figure 5. TTL-Level Outputs

Transition times for TTL-compatible inputs are specified as follows:

- For a high-to-low transition on an input signal, the level at which the input is said to be no longer high is 2 V and the level at which the input is said to be low is 0.8 V.
- For a low-to-high transition on an input signal, the level at which the input is said to be no longer low is 0.8 V and the level at which the input is said to be high is 2 V.

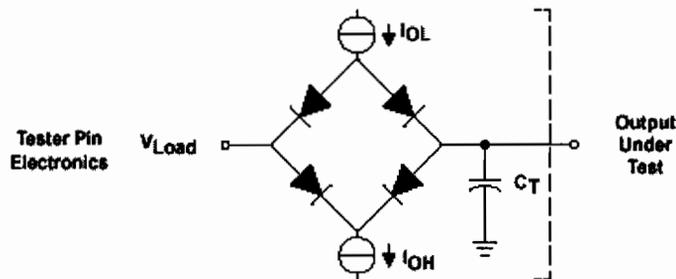


Figure 6. TTL-Level Inputs

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1986 – REVISED JANUARY 1999

### PARAMETER MEASUREMENT INFORMATION



Where:  $I_{OL}$  = 2 mA (all outputs)  
 $I_{OH}$  = 300  $\mu$ A (all outputs)  
 $V_{LOAD}$  = 2.15 V  
 $C_T$  = 80-pF typical load-circuit capacitance

Figure 7. TMS320LC31 Test Load Circuit

### signal transition levels for 'LC31 (see Figure 8 and Figure 9)

Outputs are driven to a minimum logic-high level of 2 V and to a maximum logic-low level of 0.4 V. Output transition times are specified as follows:

- For a high-to-low transition on an output signal, the level at which the output is said to be no longer high is 2 V and the level at which the output is said to be low is 1 V.
- For a low-to-high transition, the level at which the output is said to be no longer low is 1 V and the level at which the output is said to be high is 2 V.

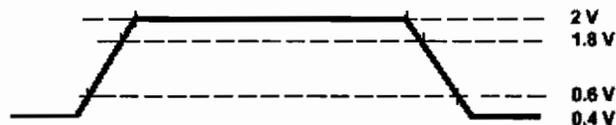


Figure 8. 'LC31 Output Levels

Transition times for inputs are specified as follows:

- For a high-to-low transition on an input signal, the level at which the input is said to be no longer high is 1.8 V and the level at which the input is said to be low is 0.6 V.
- For a low-to-high transition on an input signal, the level at which the input is said to be no longer low is 0.6 V and the level at which the input is said to be high is 1.8 V.

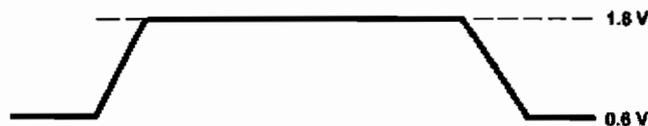


Figure 9. 'LC31 Input Levels

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

---

### PARAMETER MEASUREMENT INFORMATION

#### timing parameter symbology

Timing parameter symbols used herein were created in accordance with JEDEC Standard 100-A. In order to shorten the symbols, some of the pin names and other related terminology have been abbreviated as follows, unless otherwise noted:

A	A23–A0	H	H1 and H3
ASYNCH	Asynchronous reset signals	HOLD	$\overline{\text{HOLD}}$
C	CLKX0	HOLDA	$\overline{\text{HOLDA}}$
CI	CLKIN	IACK	$\overline{\text{IACK}}$
CLKR	CLKR0	INT	$\overline{\text{INT3}}\text{--}\overline{\text{INT0}}$
CONTROL	Control signals	RDY	$\overline{\text{RDY}}$
D	D31–D0	RW	$\overline{\text{R/W}}$
DR	DR	RESET	$\overline{\text{RESET}}$
DX	DX	S	$\overline{\text{STRB}}$
FS	FSX/R	SCK	CLKX/R
FSX	FSX0	SHZ	$\overline{\text{SHZ}}$
FSR	FSR0	TCLK	TCLK0, TCLK1, or TCLKx
GPI	General-purpose input	XF	XF0, XF1, or XFx
GPIO	General-purpose input/output; peripheral pin	XFIO	XFx switching from input to output
GPO	General-purpose output		

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1986 – REVISED JANUARY 1999

### timing

Timing specifications apply to the TMS320C31 and TMS320LC31.

### X2/CLKIN, H1, and H3 timing

The following table defines the timing parameters for the X2/CLKIN, H1, and H3 interface signals. The numbers shown in Figure 10 and Figure 11 correspond with those in the NO. column of the table below.

### timing parameters for X2/CLKIN, H1, H3 (see Figure 10 and Figure 11)

NO.		'LC31		'C31-40 'LC31-40		'C31-50		'C31-60		'C31-80		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_f(\text{Cl})$	Fall time, CLKIN		5†		5†		5†		4†		ns
2	$t_w(\text{ClL})$	Pulse duration, CLKIN low $t_c(\text{Cl}) = \text{min}$		10		9		7		6		ns
3	$t_w(\text{ClH})$	Pulse duration, CLKIN high $t_c(\text{Cl}) = \text{min}$		10		9		7		6		ns
4	$t_r(\text{Cl})$	Rise time, CLKIN		5†		5†		5†		4†		ns
5	$t_c(\text{Cl})$	Cycle time, CLKIN		30 303		25 303		20 303		16.67 303		ns
6	$t_f(\text{H})$	Fall time, H1 and H3		3		3		3		3		ns
7	$t_w(\text{HL})$	Pulse duration, H1 and H3 low		P-6‡		P-5‡		P-5‡		P-4‡		ns
8	$t_w(\text{HH})$	Pulse duration, H1 and H3 high		P-7‡		P-6‡		P-6‡		P-5‡		ns
9	$t_r(\text{H})$	Rise time, H1 and H3		4		3		3		3		ns
10	$t_d(\text{HL-HH})$	Delay time, from H1 low to H3 high or from H3 low to H1 high		0 5		0 4		0 4		0 3		ns
11	$t_c(\text{H})$	Cycle time, H1 and H3		60 606		50 606		40 606		33.3 606		ns

† Specified by design but not tested

‡ P =  $t_c(\text{Cl})$

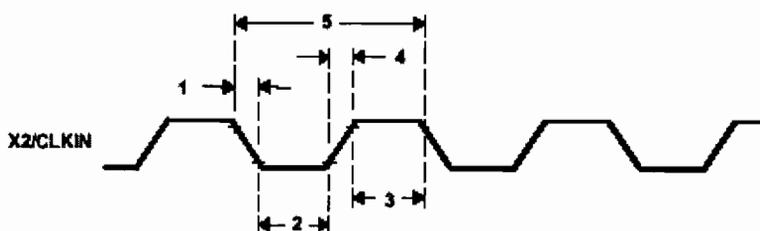


Figure 10. Timing for X2/CLKIN

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

### reset timing

**RESET** is an asynchronous input that can be asserted at any time during a clock cycle. If the specified timings are met, the exact sequence shown in Figure 21 occurs; otherwise, an additional delay of one clock cycle is possible.

The asynchronous reset signals include XF0/1, CLKX0, DX0, FSX0, CLKR0, DR0, FSR0, and TCLK0/1.

The following table defines the timing parameters for the **RESET** signal. The numbers shown in Figure 21 correspond with those in the NO. column of the following table.

Resetting the device initializes the bus control register to seven software wait states and therefore results in slow external accesses until these registers are initialized.

**HOLD** is an asynchronous input and can be asserted during reset.

### timing parameters for **RESET** for the TMS320C31 and TMS320LC31 (see Figure 21)

NO.		'LC31-33		'C31-40 'LC31-40		'LC31-40		'C31-50		'C31-60		'C31-80		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
38	$t_{su}(\text{RESET-CLK})$ Setup time, <b>RESET</b> before CLKIN low	10	pt‡	10	pt‡	10	pt‡	10	pt‡	7	pt‡	4	pt‡	ns
39	$t_d(\text{CLKINH-H1H})$ Delay time, CLKIN high to H1 high‡	2	12	2	12¶	2	14	2	10	2	10	2	8	ns
40	$t_d(\text{CLKINH-H1L})$ Delay time, CLKIN high to H1 low‡	2	12	2	12¶	2	14	2	10	2	10	2	8	ns
41	$t_{su}(\text{RESET-H1L})$ Setup time, <b>RESET</b> high before H1 low and after ten H1 clock cycles	10		9		9		7		6		5		ns
42	$t_d(\text{CLKINH-H3L})$ Delay time, CLKIN high to H3 low‡	2	12¶	2	12	2	14	2	10	2	10	2	8	ns
43	$t_d(\text{CLKINH-H3H})$ Delay time, CLKIN high to H3 high‡	2	12¶	2	12	2	14	2	10	2	10	2	8	ns
44	$t_{dis}(\text{H1H-DZ})$ Disable time, H1 high to D (high impedance)		15#		13#		13#		12#		11#		9#	ns
45	$t_{dis}(\text{H3H-AZ})$ Disable time, H3 high to A (high impedance)		10#		9#		9#		8#		7#		6#	ns
46	$t_d(\text{H3H-CONTROLH})$ Delay time, H3 high to control signals high		10#		9#		9#		8#		7#		6#	ns
47	$t_d(\text{H1H-RWH})$ Delay time, H1 high to RW high		10#		9#		9#		8#		7#		6#	ns
48	$t_d(\text{H1H-IACKH})$ Delay time, H1 high to IACK high		10#		9#		9#		8#		7#		6#	ns
49	$t_{dis}(\text{RESETL-ASYNCH})$ Disable time, <b>RESET</b> low to asynchronous reset signals disabled (high impedance)		25#		21#		21#		17#		14#		12#	ns

† P =  $t_c(\text{Cl})$

‡ Specified by design but not tested

§ See Figure 22 for temperature dependence.

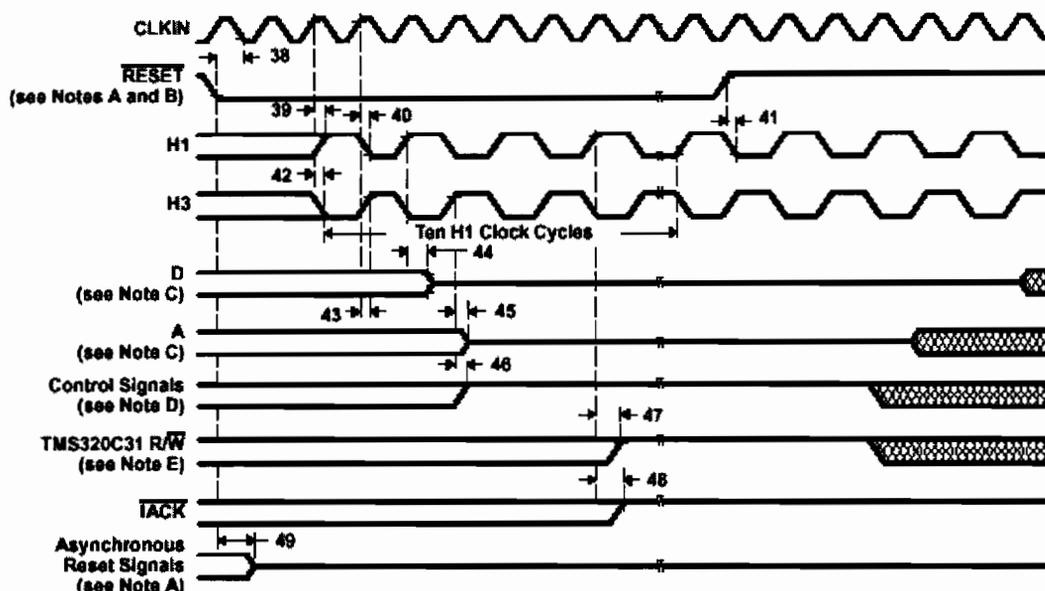
¶ 14 ns for the extended temperature 'C31-40

# This value is characterized but not tested

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

timing parameters for  $\overline{\text{RESET}}$  for the TMS320C31 and TMS320LC31 (continued)



- NOTES:
- Asynchronous reset signals include XF0/1, CLKX0, DX0, FSX0, CLKR0, DR0, FSR0, and TCLK0/1.
  - RESET is an asynchronous input and can be asserted at any point during a clock cycle. If the specified timings are met, the exact sequence shown occurs; otherwise, an additional delay of one clock cycle is possible.
  - In microprocessor mode, the reset vector is fetched twice, with seven software wait states each time. In microcomputer mode, the reset vector is fetched twice, with no software wait states.
  - Control signals include STRB.
  - The R/W outputs are placed in a high-impedance state during reset and can be provided with a resistive pullup, nominally 18–22 k $\Omega$ , if undesirable spurious writes are caused when these outputs go low.

Figure 21. Timing for  $\overline{\text{RESET}}$

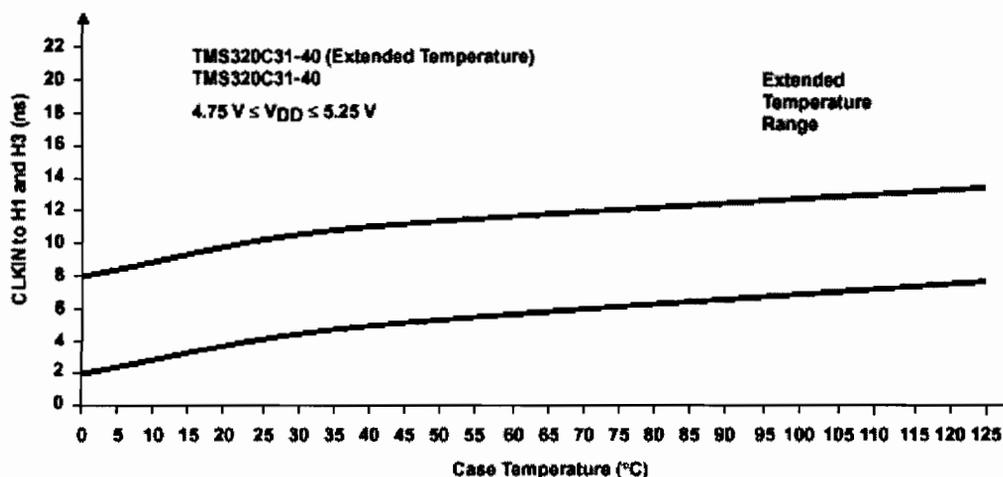


Figure 22. CLKIN to H1 and H3 as a Function of Temperature

## TMS320C31, TMS320LC31 DIGITAL SIGNAL PROCESSORS

SPRS035B – MARCH 1996 – REVISED JANUARY 1999

### Interrupt response timing

The following table defines the timing parameters for the  $\overline{\text{INT}}$  signals. The numbers shown in Figure 23 correspond with those in the NO. column of the table below.

### timing parameters for $\overline{\text{INT3}}-\overline{\text{INT0}}$ response (see Figure 23)

NO.		'LC31-33		'C31-40 'LC31-40		'C31-50		'C31-60		'C31-80		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
50	$t_{su}(\text{INT-H1L})$ Setup time, $\overline{\text{INT3}}-\overline{\text{INT0}}$ before H1 low	15		13		10		8		5		ns
51	$t_w(\text{INT})$ Pulse duration, interrupt to ensure only one interrupt	P	$2P\ddagger$	P	$2P\ddagger$	P	$2P\ddagger$	P	$2P\ddagger$	P	$2P\ddagger$	ns

† This value is characterized but not tested.

‡  $P = t_{c(H)}$

The interrupt ( $\overline{\text{INT}}$ ) pins are asynchronous inputs that can be asserted at any time during a clock cycle. The TMS320C3x interrupts are level-sensitive, not edge-sensitive. Interrupts are detected on the falling edge of H1. Therefore, interrupts must be set up and held to the falling edge of H1 for proper detection. The CPU and DMA respond to detected interrupts on instruction-fetch boundaries only.

For the processor to recognize only one interrupt on a given input, an interrupt pulse must be set up and held to:

- A minimum of one H1 falling edge
- No more than two H1 falling edges

The TMS320C3x can accept an interrupt from the same source every two H1 clock cycles.

If the specified timings are met, the exact sequence shown in Figure 23 occurs; otherwise, an additional delay of one clock cycle is possible.

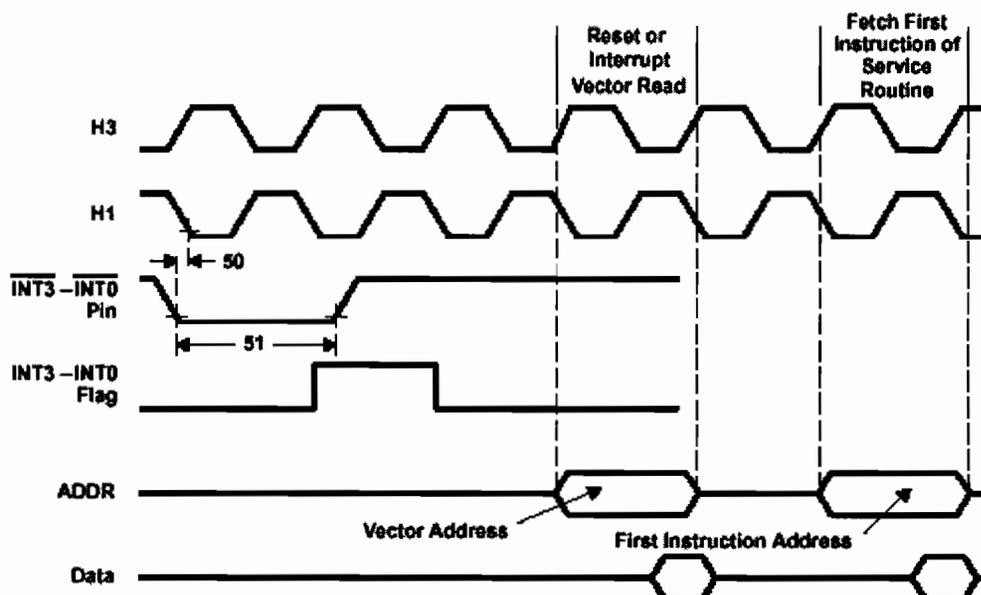


Figure 23. Timing for  $\overline{\text{INT3}}-\overline{\text{INT0}}$  Response

**TMS320C31, TMS320LC31  
DIGITAL SIGNAL PROCESSORS**

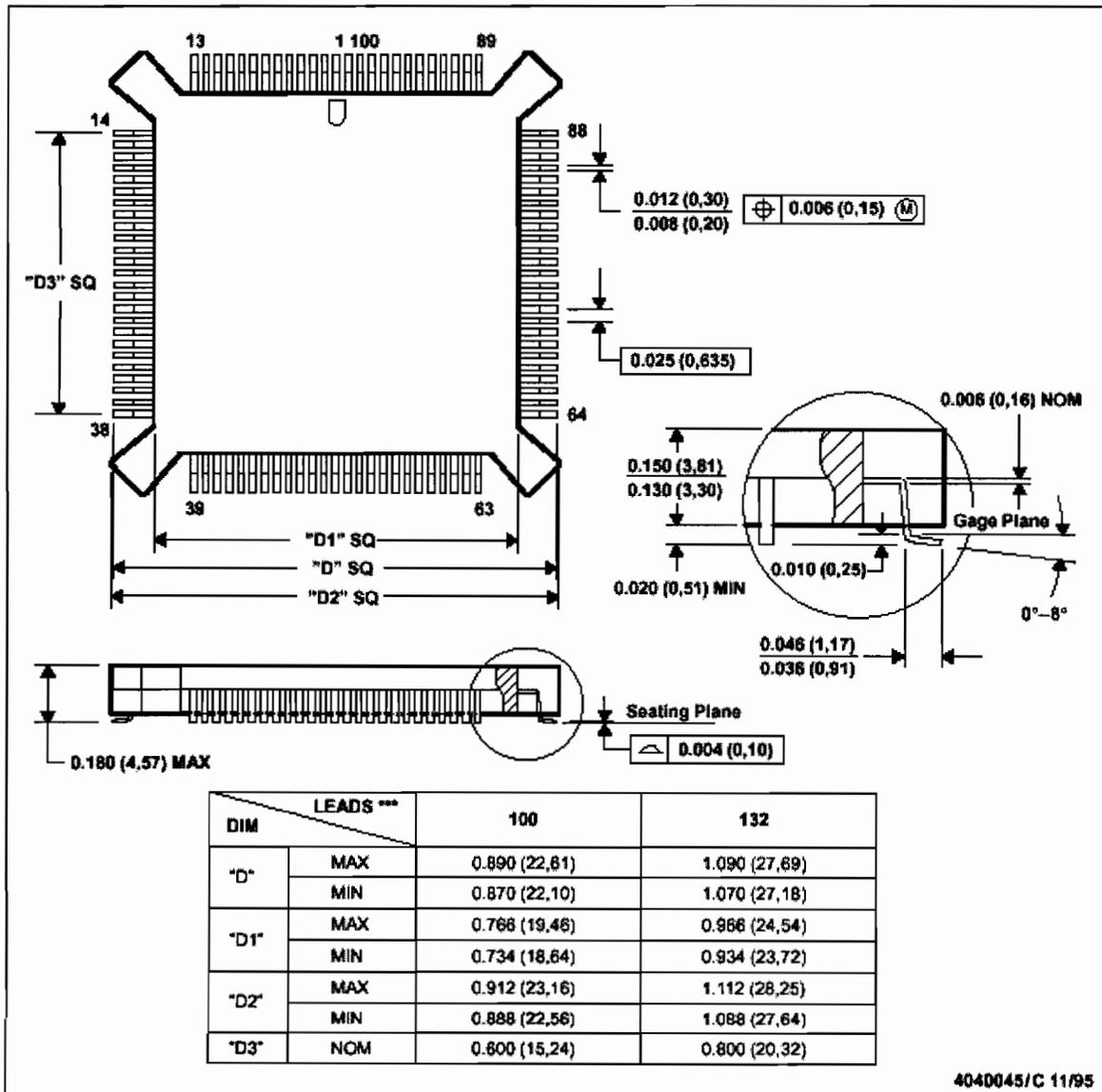
SPR5035B - MARCH 1996 - REVISED JANUARY 1996

**MECHANICAL DATA**

**PQ (S-PQFP-G\*\*\*)**

**PLASTIC QUAD FLATPACK**

100 LEAD SHOWN



4040045/C 11/95

- NOTES: A. All linear dimensions are in inches (millimeters).  
 B. This drawing is subject to change without notice.  
 C. Falls within JEDEC MO-069

## **C Set de Instrucciones del DSP TMS320C31**

---

Este anexo contiene el resumen del set de instrucciones y los códigos condicionales del Procesador Digital de Señales TMS320C31.

La siguiente tabla representa el set de instrucciones del 'C31.

### Instruction Set Summary

Mnemonic	Description	Operation
ABSF	Absolute value of a floating-point number	$ src  \rightarrow Rn$
ABSI	Absolute value of an integer	$ src  \rightarrow Dreg$
ADDC	Add integers with carry	$src + Dreg + C \rightarrow Dreg$
ADDC3	Add integers with carry (3-operand)	$src1 + src2 + C \rightarrow Dreg$
ADDF	Add floating-point values	$src + Rn \rightarrow Rn$
ADDF3	Add floating-point values (3-operand)	$src1 + src2 \rightarrow Rn$
ADDI	Add integers	$src + Dreg \rightarrow Dreg$
ADDI3	Add integers (3 operand)	$src1 + src2 + \rightarrow Dreg$
AND	Bitwise-logical AND	$Dreg \text{ AND } src \rightarrow Dreg$
AND3	Bitwise-logical AND (3-operand)	$src1 \text{ AND } src2 \rightarrow Dreg$
ANDN	Bitwise-logical AND with complement	$Dreg \text{ AND } \overline{src} \rightarrow Dreg$
ANDN3	Bitwise-logical ANDN (3-operand)	$src1 \text{ AND } \overline{src2} \rightarrow Dreg$
ASH	Arithmetic shift	If $count \geq 0$ : (Shifted $Dreg$ left by $count$ ) $\rightarrow Dreg$ Else: (Shifted $Dreg$ right by $ count $ ) $\rightarrow Dreg$
ASH3	Arithmetic shift (3-operand)	If $count \geq 0$ : (Shifted $src$ left by $count$ ) $\rightarrow Dreg$ Else: (Shifted $src$ right by $ count $ ) $\rightarrow Dreg$

Legend:			
ARn	auxiliary register n (AR7–AR0)	RE	repeat interrupt register
C	carry bit	RM	repeat mode bit
Csrc	conditional-branch addressing modes	Rn	register address (R7–R0)
count	shift value (general addressing modes)	RS	repeat start register
cond	condition code	SP	stack pointer
Daddr	destination memory address	Sreg	register address (any register)
Dreg	register address (any register)	ST	status register
GIE	global interrupt enable register	src	general addressing modes
N	any trap vector 0–27	src1	3-operand addressing modes
PC	program counter	src2	3-operand addressing modes
RC	repeat counter register	TOS	top of stack

## Instruction Set Summary

Mnemonic	Description	Operation
<i>Bcond</i>	Branch conditionally (standard)	If <i>cond</i> = true: If <i>Csrc</i> is a register, <i>Csrc</i> → PC If <i>Csrc</i> is a value, <i>Csrc</i> + PC → PC Else, PC + 1 → PC
<i>BcondD</i>	Branch conditionally (delayed)	If <i>cond</i> = true: If <i>Csrc</i> is a register, <i>Csrc</i> → PC If <i>Csrc</i> is a value, <i>Csrc</i> + PC + 3 → PC Else, PC + 1 → PC
BR	Branch unconditionally (standard)	Value → PC
BRD	Branch unconditionally (delayed)	Value → PC
CALL	Call subroutine	PC + 1 → TOS Value → PC
<i>CALLcond</i>	Call subroutine conditionally	If <i>cond</i> = true: PC + 1 → TOS If <i>Csrc</i> is a register, <i>Csrc</i> → PC If <i>Csrc</i> is a value, <i>Csrc</i> + PC → PC Else, PC + 1 → PC
CMPF	Compare floating-point values	Set flags on <i>Rn</i> – <i>src</i>
CMPF3	Compare floating-point values (3-operand)	Set flags on <i>src1</i> – <i>src2</i>
CMPI	Compare integers	Set flags on <i>Dreg</i> – <i>src</i>
CMPI3	Compare integers (3-operand)	Set flags on <i>src1</i> – <i>src2</i>
<b>Legend:</b>	<i>ARn</i> auxiliary register <i>n</i> (AR7–AR0) <i>C</i> carry bit <i>Csrc</i> conditional-branch addressing modes <i>count</i> shift value (general addressing modes) <i>cond</i> condition code <i>Daddr</i> destination memory address <i>Dreg</i> register address (any register) <i>GIE</i> global interrupt enable register <i>N</i> any trap vector 0–27 <i>PC</i> program counter <i>RC</i> repeat counter register	<i>RE</i> repeat interrupt register <i>RM</i> repeat mode bit <i>Rn</i> register address (R7–R0) <i>RS</i> repeat start register <i>SP</i> stack pointer <i>Sreg</i> register address (any register) <i>ST</i> status register <i>src</i> general addressing modes <i>src1</i> 3-operand addressing modes <i>src2</i> 3-operand addressing modes <i>TOS</i> top of stack

## Instruction Set Summary

Mnemonic	Description	Operation
DB <i>cond</i>	Decrement and branch conditionally (standard)	$ARn - 1 \rightarrow ARn$ If <i>cond</i> = true and $ARn \geq 0$ : If <i>Csrc</i> is a register, $Csrc \rightarrow PC$ If <i>Csrc</i> is a value, $Csrc + PC + 1 \rightarrow PC$ Else, $PC + 1 \rightarrow PC$
DB <i>cond</i> D	Decrement and branch conditionally (delayed)	$ARn - 1 \rightarrow ARn$ If <i>cond</i> = true and $ARn \geq 0$ : If <i>Csrc</i> is a register, $Csrc \rightarrow PC$ If <i>Csrc</i> is a value, $Csrc + PC + 3 \rightarrow PC$ Else, $PC + 1 \rightarrow PC$
FIX	Convert floating-point value to integer	Fix( <i>src</i> ) $\rightarrow$ Dreg
FLOAT	Convert integer to floating-point value	Float( <i>src</i> ) $\rightarrow$ R <i>n</i>
IACK	Interrupt acknowledge	Dummy read of <i>src</i> IACK toggled low, then high
IDLE	Idle until interrupt	$PC + 1 \rightarrow PC$ Idle until next interrupt
IDLE2	Low-power idle	Idle until next interrupt stopping internal clocks
LDE	Load floating-point exponent	<i>src</i> (exponent) $\rightarrow$ R <i>n</i> (exponent)
LDF	Load floating-point value	<i>src</i> $\rightarrow$ R <i>n</i>
LDF <i>cond</i>	Load floating-point value conditionally	If <i>cond</i> = true, <i>src</i> $\rightarrow$ R <i>n</i> Else, R <i>n</i> is not changed
LDFI	Load floating-point value, interlocked	Signal interlocked operation <i>src</i> $\rightarrow$ R <i>n</i>
LDI	Load integer	<i>src</i> $\rightarrow$ Dreg
<b>Legend:</b>	AR <i>n</i> auxiliary register <i>n</i> (AR7–AR0) C carry bit Csrc conditional-branch addressing modes count shift value (general addressing modes) <i>cond</i> condition code Daddr destination memory address Dreg register address (any register) GIE global interrupt enable register N any trap vector 0–27 PC program counter RC repeat counter register	RE repeat interrupt register RM repeat mode bit R <i>n</i> register address (R7–R0) RS repeat start register SP stack pointer Sreg register address (any register) ST status register <i>src</i> general addressing modes <i>src</i> 1 3-operand addressing modes <i>src</i> 2 3-operand addressing modes TOS top of stack

## Instruction Set Summary

Mnemonic	Description	Operation
LD  <i>cond</i>	Load integer conditionally	If <i>cond</i> = true, <i>src</i> → Dreg Else, Dreg is not changed
LDII	Load integer, interlocked	Signal interlocked operation <i>src</i> → Dreg
LDM	Load floating-point mantissa	<i>src</i> (mantissa) → R <i>n</i> (mantissa)
LDP	Load data page pointer	<i>src</i> → data page pointer
LOPOWER	Divide clock by 16	H1/16 → H1
LSH	Logical shift	If <i>count</i> ≥ 0: (Dreg left-shifted by <i>count</i> ) → Dreg Else: (Dreg right-shifted by   <i>count</i>  ) → Dreg
LSH3	Logical shift (3-operand)	If <i>count</i> ≥ 0: ( <i>src</i> left-shifted by <i>count</i> ) → Dreg Else: ( <i>src</i> right-shifted by   <i>count</i>  ) → Dreg
MAXSPEED	Restore clock to regular speed	H1/16 → H1
MPYF	Multiply floating-point values	<i>src</i> × R <i>n</i> → R <i>n</i>
MPYF3	Multiply floating-point value (3-operand)	<i>src1</i> × <i>src2</i> → R <i>n</i>
MPYI	Multiply integers	<i>src</i> × Dreg → Dreg
MPYI3	Multiply integers (3-operand)	<i>src1</i> × <i>src2</i> → Dreg
NEGB	Negate integer with borrow	0 – <i>src</i> – C → Dreg
NEGF	Negate floating-point value	0 – <i>src</i> → R <i>n</i>
NEGI	Negate integer	0 – <i>src</i> → Dreg
<b>Legend:</b>	<p>AR<i>n</i> auxiliary register <i>n</i> (AR7–AR0)</p> <p>C carry bit</p> <p>C<i>src</i> conditional-branch addressing modes</p> <p><i>count</i> shift value (general addressing modes)</p> <p><i>cond</i> condition code</p> <p>D<i>addr</i> destination memory address</p> <p>Dreg register address (any register)</p> <p>GIE global interrupt enable register</p> <p>N any trap vector 0–27</p> <p>PC program counter</p> <p>RC repeat counter register</p>	<p>RE repeat interrupt register</p> <p>RM repeat mode bit</p> <p>R<i>n</i> register address (R7–R0)</p> <p>RS repeat start register</p> <p>SP stack pointer</p> <p>Sreg register address (any register)</p> <p>ST status register</p> <p><i>src</i> general addressing modes</p> <p><i>src1</i> 3-operand addressing modes</p> <p><i>src2</i> 3-operand addressing modes</p> <p>TOS top of stack</p>

## Instruction Set Summary

Mnemonic	Description	Operation
NOP	No operation	Modify ARn if specified
NORM	Normalize floating-point value	Normalize ( <i>src</i> ) → Rn
NOT	Bitwise-logical complement	$\overline{\text{src}} \rightarrow \text{Dreg}$
OR	Bitwise-logical OR	Dreg OR <i>src</i> → Dreg
OR3	Bitwise-logical OR (3-operand)	<i>src1</i> OR <i>src2</i> → Dreg
POP	Pop integer from stack	*SP-- → Dreg
POPF	Pop floating-point value from stack	*SP-- → Rn
PUSH	Push integer on stack	Sreg → *++ SP
PUSHF	Push floating-point value on stack	Rn → *++ SP
RETI $\text{cond}$	Return from interrupt conditionally	If <i>cond</i> = true or missing: *SP-- → PC 1 → ST (GIE) Else, continue
RETS $\text{cond}$	Return from subroutine conditionally	If <i>cond</i> = true or missing: *SP-- → PC Else, continue
RND	Round floating-point value	Round ( <i>src</i> ) → Rn
ROL	Rotate left	Dreg rotated left 1 bit → Dreg
ROLC	Rotate left through carry	Dreg rotated left 1 bit through carry → Dreg
ROR	Rotate right	Dreg rotated right 1 bit → Dreg
RORC	Rotate right through carry	Dreg rotated right 1 bit through carry → Dreg
<b>Legend:</b>	ARn auxiliary register n (AR7–AR0) C carry bit Csrc conditional-branch addressing modes count shift value (general addressing modes) cond condition code Daddr destination memory address Dreg register address (any register) GIE global interrupt enable register N any trap vector 0–27 PC program counter RC repeat counter register	RE repeat interrupt register RM repeat mode bit Rn register address (R7–R0) RS repeat start register SP stack pointer Sreg register address (any register) ST status register src general addressing modes src1 3-operand addressing modes src2 3-operand addressing modes TOS top of stack

## Instruction Set Summary

Mnemonic	Description	Operation
RPTB	Repeat block of instructions	$src \rightarrow RE$ $1 \rightarrow ST (RM)$ Next PC $\rightarrow RS$
RPTS	Repeat single instruction	$src \rightarrow RC$ $1 \rightarrow ST (RM)$ Next PC $\rightarrow RS$ Next PC $\rightarrow RE$
SIGI	Signal, interlocked	Signal interlocked operation Wait for interlock acknowledge Clear interlock
STF	Store floating-point value	$Rn \rightarrow Daddr$
STFI	Store floating-point value, interlocked	$Rn \rightarrow Daddr$ Signal end of interlocked operation
STI	Store integer	$Sreg \rightarrow Daddr$
STII	Store integer, interlocked	$Sreg \rightarrow Daddr$ Signal end of interlocked operation
SUBB	Subtract integers with borrow	$Dreg - src - C \rightarrow Dreg$
SUBB3	Subtract integers with borrow (3-operand)	$src1 - src2 - C \rightarrow Dreg$
SUBC	Subtract integers conditionally	If $Dreg - src \geq 0$ : [( $Dreg - src$ ) $\ll 1$ ] OR 1 $\rightarrow Dreg$ Else, $Dreg \ll 1 \rightarrow Dreg$
SUBF	Subtract floating-point values	$Rn - src \rightarrow Rn$
SUBF3	Subtract floating-point values (3-operand)	$src1 - src2 \rightarrow Rn$
<b>Legend:</b>	$ARn$ auxiliary register $n$ (AR7–AR0) C carry bit Csrc conditional-branch addressing modes count shift value (general addressing modes) cond condition code Daddr destination memory address Dreg register address (any register) GIE global interrupt enable register N any trap vector 0–27 PC program counter RC repeat counter register	RE repeat interrupt register RM repeat mode bit $Rn$ register address (R7–R0) RS repeat start register SP stack pointer Sreg register address (any register) ST status register src general addressing modes $src1$ 3-operand addressing modes $src2$ 3-operand addressing modes TOS top of stack

---

**Instruction Set Summary**


---

<b>Mnemonic</b>	<b>Description</b>	<b>Operation</b>
SUBI	Subtract integers	$Dreg - src \rightarrow Dreg$
SUBI3	Subtract integers (3-operand)	$src1 - src2 \rightarrow Dreg$
SUBRB	Subtract reverse integer with borrow	$src - Dreg - C \rightarrow Dreg$
SUBRF	Subtract reverse floating-point value	$src - Rn \rightarrow Rn$
SUBRI	Subtract reverse integer	$src - Dreg \rightarrow Dreg$
SWI	Software interrupt	Perform emulator interrupt sequence
TRAP $cond$	Trap conditionally	If $cond = true$ or missing: Next PC $\rightarrow * ++ SP$ Trap vector N $\rightarrow PC$ 0 $\rightarrow ST$ (GIE) Else, continue
TSTB	Test bit fields	$Dreg AND src$
TSTB3	Test bit fields (3-operand)	$src1 AND src2$
XOR	Bitwise-exclusive OR	$Dreg XOR src \rightarrow Dreg$
XOR3	Bitwise-exclusive OR (3-operand)	$src1 XOR src2 \rightarrow Dreg$

---

<b>Legend:</b>	<b>ARn</b> auxiliary register $n$ (AR7–AR0)	<b>RE</b> repeat interrupt register
	<b>C</b> carry bit	<b>RM</b> repeat mode bit
	<b>Csrc</b> conditional-branch addressing modes	<b>Rn</b> register address (R7–R0)
	<b>count</b> shift value (general addressing modes)	<b>RS</b> repeat start register
	<b>cond</b> condition code	<b>SP</b> stack pointer
	<b>Daddr</b> destination memory address	<b>Sreg</b> register address (any register)
	<b>Dreg</b> register address (any register)	<b>ST</b> status register
	<b>GIE</b> global interrupt enable register	<b>src</b> general addressing modes
	<b>N</b> any trap vector 0–27	<b>src1</b> 3-operand addressing modes
	<b>PC</b> program counter	<b>src2</b> 3-operand addressing modes
	<b>RC</b> repeat counter register	<b>TOS</b> top of stack

## Parallel Instruction Set Summary

## (a) Parallel arithmetic with store instructions

Mnemonic	Description	Operation
ABSF	Absolute value of a floating point	$ src2  \rightarrow dst1$
STF		$src3 \rightarrow dst2$
ABSI	Absolute value of an integer	$ src2  \rightarrow dst1$
STI		$src3 \rightarrow dst2$
ADDF3	Add floating-point value	$src1 + src2 \rightarrow dst1$
STF		$src3 \rightarrow dst2$
ADDI3	Add integer	$src1 + src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$
AND3	Bitwise-logical AND	$src1 \text{ AND } src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$
ASH3	Arithmetic shift	If $count \geq 0$ :
STI		$(src2 \ll count) \rightarrow dst1$
		$src3 \rightarrow dst2$
		Else:
		$(src2 \gg  count ) \rightarrow dst1$
		$src3 \rightarrow dst2$
FIX	Convert floating-point value to integer	$\text{Fix}(src2) \rightarrow dst1$
STI		$src3 \rightarrow dst2$
FLOAT	Convert integer to floating-point value	$\text{Float}(src2) \rightarrow dst1$
STF		$src3 \rightarrow dst2$

**Legend:**

<i>count</i>	register addr (R7–R0)	<i>op3</i>	register addr (R0 or R1)
<i>dst1</i>	register addr (R7–R0)	<i>op6</i>	register addr (R2 or R3)
<i>dst2</i>	indirect addr ( <i>disp</i> = 0, 1, IR0, IR1)	<i>src1</i>	register addr (R7–R0)
<i>op1, op2, op4, and op5</i>	Any two of these operands must be specified using register addr; the remaining two must be specified using indirect.	<i>src2</i>	indirect addr ( <i>disp</i> = 0, 1, IR0, IR1)
		<i>src3</i>	register addr (R7–R0)

## Parallel Instruction Set Summary

## (a) Parallel arithmetic with store instructions (Continued)

Mnemonic	Description	Operation
LDF	Load floating-point value	$src2 \rightarrow dst1$
STF		$src3 \rightarrow dst2$
LDI	Load integer	$src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$
LSH3	Logical shift	If $count \geq 0$ :
STI		$src2 \ll count \rightarrow dst1$
		$src3 \rightarrow dst2$
		Else:
		$src2 \gg  count  \rightarrow dst1$
		$src3 \rightarrow dst2$
MPYF3	Multiply floating-point value	$src1 \times src2 \rightarrow dst1$
STF		$src3 \rightarrow dst2$
MPYI3	Multiply integer	$src1 \times src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$
NEGF	Negate floating-point value	$0 - src2 \rightarrow dst1$
STF		$src3 \rightarrow dst2$
NEGI	Negate integer	$0 - src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$
NOT	Complement	$\overline{src1} \rightarrow dst1$
STI		$src3 \rightarrow dst2$
OR3	Bitwise-logical OR	$src1 \text{ OR } src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$
STF	Store floating-point value	$src1 \rightarrow dst1$
STF		$src3 \rightarrow dst2$
STI	Store integer	$src1 \rightarrow dst1$
STI		$src3 \rightarrow dst2$

**Legend:***count* register addr (R7–R0)*dst1* register addr (R7–R0)*dst2* indirect addr (*disp* = 0, 1, IR0, IR1)*op1*, *op2*, *op4*, and *op5*

Any two of these operands must be specified using register addr; the remaining two must be specified using indirect.

*op3* register addr (R0 or R1)*op6* register addr (R2 or R3)*src1* register addr (R7–R0)*src2* indirect addr (*disp* = 0, 1, IR0, IR1)*src3* register addr (R7–R0)

**Parallel Instruction Set Summary**


---

**(a) Parallel arithmetic with store instructions (Continued)**

Mnemonic	Description	Operation
SUBF3	Subtract floating-point value	$src1 - src2 \rightarrow dst1$
STF		$src3 \rightarrow dst2$
SUBI3	Subtract integer	$src1 - src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$
XOR3	Bitwise-exclusive OR	$src1 \text{ XOR } src2 \rightarrow dst1$
STI		$src3 \rightarrow dst2$

**(b) Parallel load instructions**

Mnemonic	Description	Operation
LDF	Load floating-point value	$src2 \rightarrow dst1$
LDF		$src4 \rightarrow dst2$
LDI	Load integer	$src2 \rightarrow dst1$
LDI		$src4 \rightarrow dst2$

**(c) Parallel multiply and add/subtract instructions**

Mnemonic	Description	Operation
MPYF3	Multiply and add floating-point value	$op1 \times op2 \rightarrow op3$
ADDF3		$op4 + op5 \rightarrow op6$
MPYF3	Multiply and subtract floating-point value	$op1 \times op2 \rightarrow op3$
SUBF3		$op4 - op5 \rightarrow op6$
MPYI3	Multiply and add integer	$op1 \times op2 \rightarrow op3$
ADDI3		$op4 + op5 \rightarrow op6$
MPYI3	Multiply and subtract integer	$op1 \times op2 \rightarrow op3$
SUBI3		$op4 - op5 \rightarrow op6$

**Legend:**

count	register addr (R7–R0)	op3	register addr (R0 or R1)
dst1	register addr (R7–R0)	op6	register addr (R2 or R3)
dst2	indirect addr (disp = 0, 1, IR0, IR1)	src1	register addr (R7–R0)
op1, op2, op4, and op5		src2	indirect addr (disp = 0, 1, IR0, IR1)
	Any two of these operands must be specified using register addr; the remaining two must be specified using indirect.	src3	register addr (R7–R0)

La siguiente tabla muestra los códigos condicionales del set de instrucciones del 'C31.

*(a) Unconditional compares*

Condition	Code	Description	Flag†
U	00000	Unconditional	Irrelevant

*(b) Unsigned compares*

Condition	Code	Description	Flag†
LO	00001	Lower than	C
LS	00010	Lower than or same as	C OR Z
HI	00011	Higher than	-C AND -Z
HS	00100	Higher than or same as	-C
EQ	00101	Equal to	Z
NE	00110	Not equal to	-Z

*(c) Signed compares*

Condition	Code	Description	Flag†
LT	00111	Less than	N
LE	01000	Less than or equal to	N OR Z
GT	01001	Greater than	-N AND -Z
GE	01010	Greater than or equal to	-N
EQ	00101	Equal to	Z
NE	00110	Not equal to	-Z

*(d) Compare to zero*

Condition	Code	Description	Flag†
Z	00101	Zero	Z
NZ	00110	Not zero	-Z
P	01001	Positive	-N AND -Z
N	00111	Negative	N
NN	01010	Non-negative	-N

*(e) Compare to condition flags*

Condition	Code	Description	Flag†
NN	01010	Non-negative	-N
N	00111	Negative	N
NZ	00110	Nonzero	-Z
Z	00101	Zero	Z
NV	01100	No overflow	-V
V	01101	Overflow	V
NUF	01110	No underflow	-UF
UF	01111	Underflow	UF
NC	00100	No carry	-C
C	00001	Carry	C
NLV	10000	No latched overflow	-LV
LV	10001	Latched overflow	LV
NLUF	10010	No latched floating-point underflow	-LUF
LUF	10011	Latched floating-point underflow	LUF
ZUF	10100	Zero or floating-point underflow	Z OR UF

† - = logical complement (not true condition)

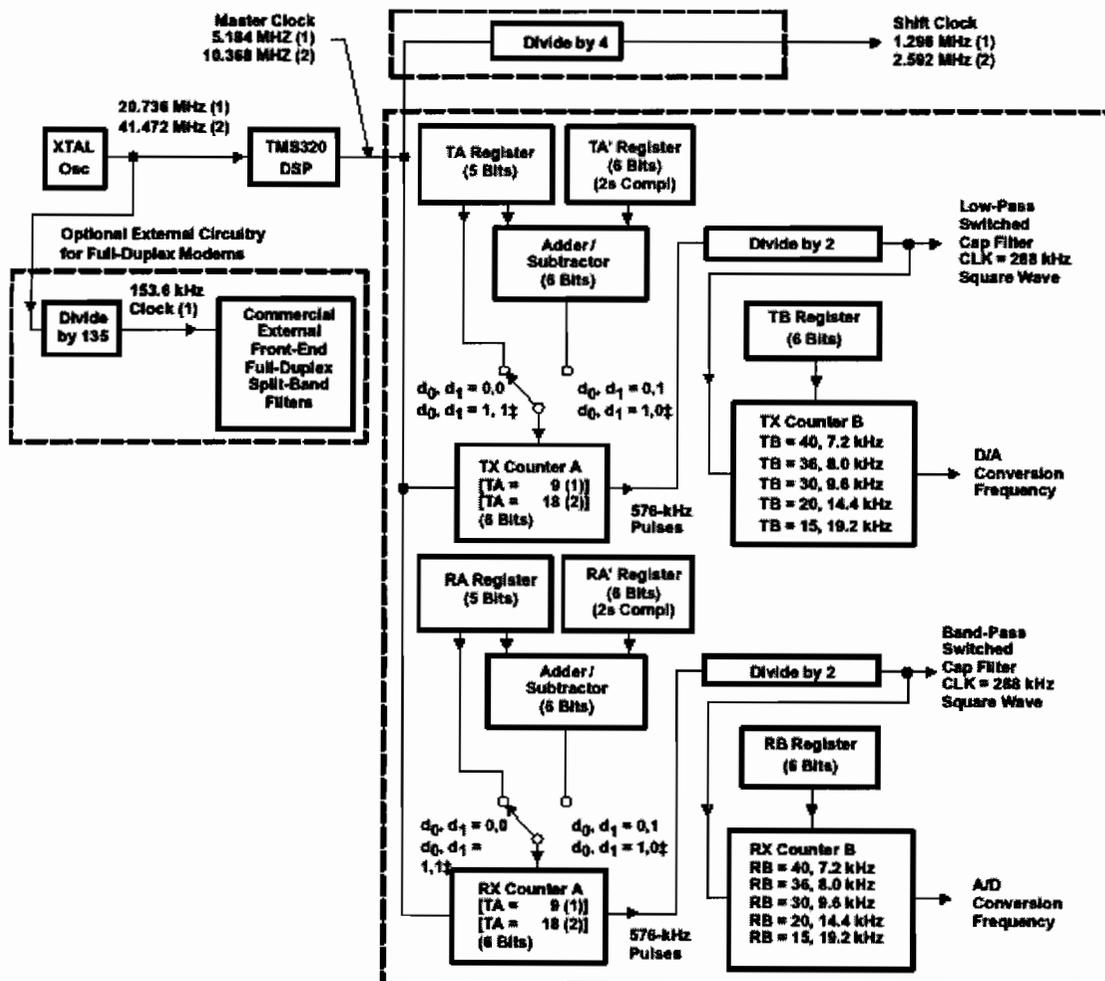
**Anexo D****D Hojas de Datos del AIC TLC32040**

---

Este anexo contiene las hojas de datos del circuito de interfaz análogo AIC TLC32040 proporcionado por la tarjeta DSK



## INTERNAL TIMING CONFIGURATION



$$\text{SCF Clock Frequency} = \frac{\text{Master Clock Frequency}}{2 \times \text{Contents of Counter A}}$$

† Split-band filtering can alternatively be performed after the analog input function via software in the SMJ320.

‡ These control bits are described in the AIC DX data word format section.

NOTE: Frequency 1, 20.736 MHz is used to show how 153.6 kHz (for a commercially available modem split-band filter clock), popular speech and modem sampling signal frequencies, and an internal 288-kHz switched-capacitor filter clock can be derived synchronously and as submultiples of the crystal oscillator frequency. Since these derived frequencies are synchronous submultiples of the crystal frequency, aliasing does not occur as the sampled analog signal passes between the analog converter and switched-capacitor filter stages. Frequency 2, 41.472 MHz is used to show that the AIC can work with high-frequency signals, which are used by high-speed digital signal processors.

### explanation of internal timing configuration

All of the internal timing of the AIC is derived from the high-frequency clock signal that drives the master clock input. The shift clock signal, which strobes the serial port data between the AIC and DSP, is derived by dividing the master clock input signal frequency by four.

$$\text{SCF Clock Frequency} = \frac{\text{Master Clock Frequency}}{2 \times \text{Contents of Counter A}}$$

$$\text{Conversion Frequency} = \frac{\text{SCF Clock Frequency}}{\text{Contents of Counter B}}$$

$$\text{Shift Clock Frequency} = \frac{\text{Master Clock Frequency}}{4}$$

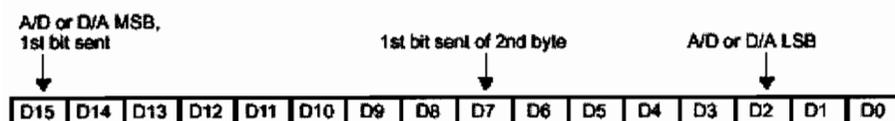
TX Counter A and TX Counter B, which are driven by the master clock signal, determine the D/A conversion timing. Similarly, RX Counter A and RX Counter B determine the A/D conversion timing. In order for the switched-capacitor low-pass and band-pass filters to meet their transfer function specifications, the frequency of the clock inputs of the switched-capacitor filters must be 288 kHz. If the frequencies of the clock inputs are not 288 kHz, the filter transfer function frequencies are scaled by the ratios of the clock frequencies to 288 kHz. Thus, to obtain the specified filter responses, the combination of master clock frequency and TX Counter A and RX Counter A values must yield 288-kHz switched-capacitor clock signals. These 288-kHz clock signals can then be divided by TX Counter B and RX Counter B to establish the D/A and A/D conversion timings.

TX Counter A and TX Counter B are reloaded every D/A conversion period, while RX Counter A and RX Counter B are reloaded every A/D conversion period. TX Counter B and RX Counter B are loaded with the values in the TB and RB Registers, respectively. Via software control, TX Counter A can be loaded with either TA Register, the TA Register less the TA' Register, or the TA Register plus the TA' Register. By selecting the TA Register less the TA' Register option, the upcoming conversion timing will occur earlier by an amount of time that equals TA' times the signal period of the master clock. By selecting the TA Register plus the TA' Register option, the upcoming conversion timing will occur later by an amount of time that equals TA' times the signal period of the master clock. Thus the D/A conversion timing can be advanced or retarded. An identical ability to alter the A/D conversion timing is provided. In this case, however, the RX Counter A can be programmed via software control with the RA Register, the RA Register less the RA' Register, or the RA Register plus the RA' Register.

The ability to advance or retard conversion timing is particularly useful for modem applications. This feature allows controlled changes in the A/D and D/A conversion timing. This feature can be used to enhance signal-to-noise performance, to perform frequency-tracking functions, and to generate nonstandard modem frequencies.

If the transmit and receive sections are configured to be synchronous (see the WORD/BYTE description in the Terminal Functions table), then both the low-pass and band-pass switched-capacitor filter clocks are derived from TX Counter A. Also, both the D/A and A/D conversion timing are derived from TX Counter A and TX Counter B. When the transmit and receive sections are configured to be asynchronous, the RX Counter A, RX Counter B, RA Register, RA' Register, and RB Registers are not used.

#### AIC DR or DX word bit pattern



#### AIC DX data word format section

d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d2	d1	d0	COMMENTS	
Primary DX serial communication protocol																
← d15 (MSB) through d2 go to the D/A converter register												→		0	0	The TX and RX Counter As are loaded with the TA and RA register values. The TX and RX Counter Bs are reloaded with TB and RB register values.
← d15 (MSB) through d2 go to the D/A converter register												→		0	1	The TX and Counter As are loaded with the TA + TA' and RA + RA' register values. The TX and RX Counter Bs are loaded with the TB and RB register values. NOTE: d1 = 0, d0 = 1 will cause the next D/A and A/D conversion periods to be changed by the addition of TA' and RA' master clock cycles, in which TA' and RA' can be positive or negative or zero. Please refer to Table 1. AIC Responses to Improper Conditions.
← d15 (MSB) through d2 go to the D/A converter register												→		1	0	The TX and Counter As are loaded with the TA - TA' and RA - RA' register values. The TX and RX Counter Bs are loaded with the TB and RB register values. NOTE: d1 = 0, d0 = 1 will cause the next D/A and A/D conversion periods to be changed by the subtraction of TA' and RA' Master Clock cycles, in which TA' and RA' can be positive or negative or zero. Please refer to Table 1. AIC Responses to Improper Conditions.

← d15 (MSB) through d2 go to the D/A converter register	→   1 1	The TX and Counter As are loaded with the TA and RA register values. The TX and RX Counter Bs are loaded with the TB and RB register values. After a delay of four shift-clock cycles, a secondary transmission will immediately follow to program the AIC to operate in the desired configuration.
---	---------	---

NOTE: Setting the two least significant bits to 1 in the normal transmission of DAC information (Primary Communications) to the AIC will initiate Secondary Communications upon completion of the Primary Communications. Upon completion of the Primary Communication, FSR will remain high for four shift-clock cycles and will then go low and initiate the Secondary Communication. The timing specifications for the Primary and Secondary Communications are identical. In this manner, the Secondary Communication, if initiated, is interleaved between successive Primary Communications. This interleaving prevents the Secondary Communication from interfering with the Primary Communications and DAC timing, thus preventing the AIC from skipping a DAC output. It is important to note that in the synchronous mode, FSR will not be asserted during Secondary Communications.

### secondary DX serial communication protocol

x x   ← to TA register →   x x   ← to RA register →   0 0	d13 and d6 are MSBs (unsigned binary)
x   ← to TA' register →   x   ← to RA' register →   0 1	d14 and d7 are 2s complement sign bits
x   ← to TB register →   x   ← to RB register →   1 0	d14 and d7 are MSBs (unsigned binary)
x x x x x x x x x   d7 d6 d5 d4 d3 d2   1 1	
<div style="text-align: center;">           ← Control Register →         </div>	d2 = 0/1 deletes/inserts the band-pass filter d3 = 0/1 disables/enables the loopback function d4 = 0/1 disables/enables the AUX IN+ and AUX IN- terminals d5 = 0/1 asynchronous/synchronous transmit and receive sections d6 = 0/1 gain control bits (see gain control section) d7 = 0/1 gain control bits (see gain control section)

### reset function

A reset function is provided to initiate serial communications between the AIC and DSP. The reset function will initialize all AIC registers, including the control register. After power has been applied to the AIC, a negative-going pulse on  $\overline{\text{RESET}}$  will initialize the AIC registers to provide an 8-kHz A/D and D/A conversion rate for a 5.184-MHz master clock input signal. The AIC, except the control register, will be initialized as follows (see AIC DX data word format section):

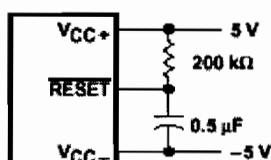
REGISTER	INITIALIZED REGISTER VALUE (HEX)
TA	9
TA'	1
TB	24
RA	9
RA'	1
RB	24

The control register bits will be reset as follows (see AIC DX data word format section):

$$d7 = 1, d6 = 1, d5 = 1, d4 = 0, d3 = 0, d2 = 1$$

This initialization allows normal serial port communications to occur between the AIC and DSP. If the transmit and receive sections are configured to operate synchronously and the user wishes to program different conversion rates, only the TA, TA', and TB registers need to be programmed, since both transmit and receive timing are synchronously derived from these registers (see the Terminal Functions table and AIC DX data word format section).

The circuit shown below provides a reset on power up when power is applied in the sequence given under power-up sequence. The circuit depends on the power supplies' reaching their recommended values a minimum of 800 ns before the capacitor charges to 0.8 V above DGTL GND.



## power-up sequence

To ensure proper operation of the AIC and as a safeguard against latch-up, it is recommended that a Schottky diode with a forward voltage less than or equal to 0.4 V be connected from  $V_{CC-}$  to ANLG GND (see Figure 16). In the absence of such a diode, power should be applied in the following sequence: ANLG GND and DGTL GND,  $V_{CC-}$ , then  $V_{CC+}$  and  $V_{DD}$ . Also, no input signal should be applied until after power up.

## AIC responses to improper conditions

The AIC has provisions for responding to improper conditions. These improper conditions and the response of the AIC to these conditions are presented in Table 1 below.

## AIC register constraints

The following constraints are placed on the contents of the AIC registers:

1. TA register must be  $\geq 4$  in WORD mode (WORD/ $\overline{\text{BYTE}}$  = high).
2. TA register must be  $\geq 5$  in  $\overline{\text{BYTE}}$  mode (WORD/ $\overline{\text{BYTE}}$  = low).
3. TA' register can be either positive, negative, or zero.
4. RA register must be  $\geq 4$  in WORD mode (WORD/ $\overline{\text{BYTE}}$  = high).
5. RA register must be  $\geq 5$  in  $\overline{\text{BYTE}}$  mode (WORD/ $\overline{\text{BYTE}}$  = low).
6. RA' register can be either positive, negative, or zero.
7. (TA register  $\pm$  TA' register) must be  $> 1$ .
8. (RA register  $\pm$  RA' register) must be  $> 1$ .
9. TB register must be  $> 1$ .

Table 1. AIC Responses to Improper Conditions

IMPROPER CONDITION	AIC RESPONSE
TA register + TA' register = 0 or 1 TA register – TA' register = 0 or 1	Reprogram TX Counter A with TA register value
TA register + TA' register < 0	MODULO 64 arithmetic is used to ensure that a positive value is loaded into the TX Counter A, i.e., TA register + TA' register + 40 HEX is loaded into TX Counter A.
RA register + RA' register = 0 or 1 RA register – RA' register = 0 or 1	Reprogram RX Counter A with RA register value
RA register + RA' register = 0 or 1	MODULO 64 arithmetic is used to ensure that a positive value is loaded into the RX Counter A, i.e., RA register + RA' register + 40 HEX is loaded into RX Counter A.
TA register = 0 or 1 RA register = 0 or 1	AIC is shutdown.
TA register < 4 in WORD mode TA register < 5 in $\overline{\text{BYTE}}$ mode RA register < 4 in WORD mode RA register < 5 in $\overline{\text{BYTE}}$ mode	The AIC serial port no longer operates.
TB register = 0 or 1	Reprogram TB register with 24 HEX
RB register = 0 or 1	Reprogram TB register with 24 HEX
AIC and DSP cannot communicate	Hold last DAC output

## absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage range, $V_{CC+}$ (see Note 1)	–0.3 V to 15 V
Supply voltage range, $V_{DD}$	–0.3 V to 15 V
Output voltage range, $V_O$	–0.3 V to 15 V
Input voltage range, $V_I$	–0.3 V to 15 V
Digital ground voltage range	–0.3 V to 15 V
Operating free-air temperature range	–55°C to 125°C
Storage temperature range	–65°C to 150°C
Case temperature for 60 seconds: FK package	280°C
Lead temperature 1,6 mm (1/16 inch) from case for 60 seconds: J package	300°C

NOTE 1: Voltage values for maximum ratings are with respect to  $V_{CC-}$ .

## recommended operating conditions

PARAMETER	MIN	NOM	MAX	UNIT
Supply voltage, $V_{CC+}$ (see Note 2)	4.75	5	5.25	V
Supply voltage, $V_{CC-}$ (see Note 2)	-4.75	-5	-5.25	V
Digital supply voltage, $V_{DD}$ (see Note 2)	4.75	5	5.25	V
Digital ground voltage with respect to ANLG GND, DGTL GND		0		V
Reference input voltage, $V_{ref(ext)}$ (see Note 2)	2		4	V
High-level input voltage, $V_{IH}$	2	$V_{DD}+0.3$		V
Low-level input voltage, $V_{IL}$ (see Note 3)	-0.3		0.8	V
Maximum peak output voltage swing across $R_L$ at OUT+ or OUT- (single ended) (see Note 4)	$\pm 3$			V
Load resistance at OUT+ and/or OUT-, $R_L$	300			$\Omega$
Load capacitance at OUT+ and/or OUT-, $C_L$			100	pF
MSTR CLK frequency (see Note 5)	0.075	5	10.368	MHz
Analog input amplifier common-mode input voltage (see Note 6)			$\pm 1.5$	V
A/D or D/A conversion rate			20	kHz
Operating free-air temperature, $T_A$	-55		125	$^{\circ}\text{C}$

NOTES: 2. Voltages at analog inputs and outputs, REF,  $V_{CC+}$ , and  $V_{CC-}$ , are with respect to the ANLG GND terminal. Voltages at digital inputs and outputs and  $V_{DD}$  are with respect to the DGTL GND terminal.

3. The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels and temperature only.

4. This applies when  $R_L \geq 300 \Omega$  and offset voltage = 0.

5. The band-pass and low-pass switched-capacitor filter response specifications apply only when the switched-capacitor clock frequency is 288 kHz. For switched-capacitor filter clocks at frequencies other than 288 kHz, the filter response is shifted by the ratio of switched-capacitor filter clock frequency to 288 kHz.

6. This range applies when  $(IN+ - IN-)$  or  $(AUX IN+ - AUX IN-)$  equals  $\pm 6$  V.

**electrical characteristics over recommended operating free-air temperature range,  $V_{CC+} = 5$  V,  $V_{CC-} = -5$  V,  $V_{DD} = 5$  V (unless otherwise noted)**
**total device, MSTR CLK frequency = 5.184 MHz, outputs not loaded**

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$V_{OH}$ High-level output voltage	$V_{DD} = 4.75$ V, $I_{OH} = -300 \mu\text{A}$	2.4			V
$V_{OL}$ Low-level output voltage	$V_{DD} = 4.75$ V, $I_{OL} = 2$ mA			0.9	V
$I_{CC+}$ Supply current from $V_{CC+}$				40	mA
$I_{CC-}$ Supply current from $V_{CC-}$				-40	mA
$I_{DD}$ Supply current from $V_{DD}$	$\dot{M}$ MSTR CLK = 5.184 MHz			7	mA
$V_{ref}$ Internal reference output voltage		2.9		3.3	V
$\alpha_{Vref}$ Temperature coefficient of internal reference voltage			200		ppm/ $^{\circ}\text{C}$
$r_o$ Output resistance at REF			100		k $\Omega$

## receive amplifier input

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
A/D converter offset error (filters bypassed)			25	65	mV
A/D converter offset error (filters in)			25	65	mV
CMRR Common-mode rejection ratio at IN+, IN-, or AUX IN+, AUX IN-	See Note 7	35	55		dB
$r_i$ Input resistance at IN+, IN-, or AUX IN+, AUX IN-, REF			100		k $\Omega$

NOTE 7: The test condition is a 0-dBm, 1-kHz input signal with an 8-kHz conversion rate.

## transmit filter output

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$V_{OO}$ Output offset voltage at OUT+ or OUT- (single ended relative to ANLG GND)			15	75	mV
$V_{OM}$ Maximum peak output voltage swing between OUT+ and OUT- (differential output)	$R_L \geq 300 \Omega$	$\pm 6$			V

## system distortion specifications, SCF clock frequency = 288 kHz

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Attenuation of second harmonic of A/D input signal	Single ended	$V_I = -0.5$ dB to $-24$ dB referred to $V_{ref}$ .	62	70		dB
	Differential	Single-ended tested at $25^\circ\text{C}$ . See Note 8	62	70		
Attenuation of third and higher harmonics of A/D input signal	Single ended	$V_I = -0.5$ dB to $-24$ dB referred to $V_{ref}$ .	57	65		dB
	Differential	Single-ended tested at $25^\circ\text{C}$ . See Note 8	57	65		
Attenuation of second harmonic of D/A input signal	Single ended	$V_I = -0$ dB to $-24$ dB referred to $V_{ref}$ .		70		dB
	Differential	See Note 8	62	70		
Attenuation of third and higher harmonics of D/A input signal	Single ended	$V_I = -0$ dB to $-24$ dB referred to $V_{ref}$ .		65		dB
	Differential	See Note 8	57	65		

† All typical values are at  $T_A = 25^\circ\text{C}$ .NOTE 8: The test condition is a 1-kHz input signal with an 8-kHz conversion rate (0 dB relative to  $V_{ref}$ ). The load impedance for the DAC is  $300\ \Omega$ .electrical characteristics over recommended operating free-air temperature range,  $V_{CC+} = 5\ \text{V}$ ,  $V_{CC-} = -5\ \text{V}$ ,  $V_{DD} = 5\ \text{V}$  (unless otherwise noted) (continued)

## A/D channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 8)	$A_V = 1^\ddagger$		$A_V = 2^\ddagger$		$A_V = 4^\ddagger$		UNIT
		MIN	MAX	MIN	MAX	MIN	MAX	
A/D channel signal-to-distortion ratio	$V_I = -6$ dB to $-0.5$ dB	58		$>58^\ddagger$		$>58^\ddagger$		dB
	$V_I = -12$ dB to $-6$ dB	58		58		$>58^\ddagger$		
	$V_I = -18$ dB to $-12$ dB	56		58		58		
	$V_I = -24$ dB to $-18$ dB	50		56		58		
	$V_I = -30$ dB to $-24$ dB	44		50		56		
	$V_I = -36$ dB to $-30$ dB	38		44		50		
	$V_I = -42$ dB to $-36$ dB	32		38		44		
	$V_I = -48$ dB to $-42$ dB	26		32		38		
	$V_I = -54$ dB to $-48$ dB	20		26		32		

†  $A_V$  is the programmable gain of the input amplifier.‡ A value  $> 58$  is overrange and signal clipping occurs over range.

## D/A channel signal-to-distortion ratio

PARAMETER	TEST CONDITIONS (see Note 8)	MIN	MAX	UNIT
D/A channel signal-to-distortion ratio	$V_I = -6$ dB to $0$ dB		58	dB
	$V_I = -12$ dB to $-6$ dB		58	
	$V_I = -18$ dB to $-12$ dB		56	
	$V_I = -24$ dB to $-18$ dB		50	
	$V_I = -30$ dB to $-24$ dB		44	
	$V_I = -36$ dB to $-30$ dB		38	
	$V_I = -42$ dB to $-36$ dB		32	
	$V_I = -48$ dB to $-42$ dB		26	
	$V_I = -54$ dB to $-48$ dB		20	

NOTE 8: The test condition is a 1-kHz input signal with an 8-kHz conversion rate (0 dB relative to  $V_{ref}$ ). The load impedance for the DAC is  $300\ \Omega$ .

## gain and dynamic range

PARAMETER	TEST CONDITIONS	MIN	TYP§	MAX	UNIT
Absolute transmit gain tracking error while transmitting into $300\ \Omega$	$-48$ dB to $0$ dB signal range, See Note 9		$\pm 0.05$	$\pm 0.15$	dB
Absolute receive gain tracking error	$-48$ dB to $0$ dB signal range, See Note 9		$\pm 0.05$	$\pm 0.15$	dB
Absolute gain of the A/D channel	Signal input is a $-0.5$ dB, 1-kHz sine wave		0.2		dB
Absolute gain of the D/A channel	Signal input is a $0$ dB, 1-kHz sine wave		$-0.3$		dB

§ All typical values are at  $T_A = 25^\circ\text{C}$ .NOTE 9: Gain tracking is relative to the absolute gain at 1 kHz and 0 dB (0 dB relative to  $V_{ref}$ ).

electrical characteristics over recommended operating free-air temperature range,  $V_{CC+} = 5\text{ V}$ ,  $V_{CC-} = -5\text{ V}$ ,  $V_{DD} = 5\text{ V}$  (unless otherwise noted) (continued)

power supply rejection and crosstalk attenuation

PARAMETER	TEST CONDITIONS		MIN	TYP†	MAX	UNIT
$V_{CC+}$ or $V_{CC-}$ supply voltage rejection ratio, receive channel	$f = 0$ to $30\text{ kHz}$	Idle channel, Supply signal at $200\text{ mV p-p}$ measured at DR (ADC output)		30		dB
	$f = 30\text{ kHz}$ to $50\text{ kHz}$			45		
$V_{CC+}$ or $V_{CC-}$ supply voltage rejection ratio, transmit channel (single ended)	$f = 0$ to $30\text{ kHz}$	Idle channel, Supply signal at $200\text{ mV p-p}$ measured at OUT+		30		dB
	$f = 30\text{ kHz}$ to $50\text{ kHz}$			45		
Crosstalk attenuation (differential)	Transmit-to-receive	DX = 0000000000000	70	80		dB
	Receive-to-transmit	Inputs grounded	70	80		

delay distortion, SCF clock frequency =  $288\text{ kHz} \pm 2\%$ , Input (IN+ – IN–) is  $\pm 3\text{-V}$  sinewave

Please refer to filter response graphs for delay distortion specifications.

band-pass filter transfer function (see curves), SCF clock frequency =  $288\text{ kHz} \pm 2\%$ , Input (IN+ – IN–) is  $\pm 3\text{-V}$  sinewave (see Note 9)

PARAMETER	TEST CONDITIONS		MIN	MAX	UNIT
Filter gain (see Note 10)	Input signal reference is $0\text{ dB}$	$f = 100\text{ Hz}$		-42	dB
		$f = 170\text{ Hz}$		-25	
		$300\text{ Hz} \leq f \leq 3.4\text{ kHz}$		$\pm 0.5$	
		$f = 4\text{ kHz}$		-16	
		$f \geq 4.6\text{ kHz}$		-58	

low-pass filter transfer function, SCF clock frequency =  $288\text{ kHz} \pm 2\%$  (see Note 10)

PARAMETER	TEST CONDITIONS		MIN	MAX	UNIT
Filter gain (see Note 11)	Output signal reference is $0\text{ dB}$	$f \leq 3.4\text{ kHz}$		$\pm 0.5$	dB
		$f = 3.6\text{ kHz}$		-4	
		$f = 4\text{ kHz}$		-30	
		$f \geq 4.4\text{ kHz}$		-58	

serial port

PARAMETER	TEST CONDITIONS		MIN	TYP†	MAX	UNIT
$V_{OH}$ High-level output voltage		$I_{OH} = -300\text{ }\mu\text{A}$	2.4			V
$V_{OL}$ Low-level output voltage		$I_{OL} = 2\text{ mA}$			0.4	V
$I_I$ Input current					$\pm 10$	$\mu\text{A}$
$C_I$ Input capacitance				15		pF
$C_O$ Output capacitance				15		pF

† All typical values are at  $T_A = 25^\circ\text{C}$ .

NOTES: 9. Gain tracking is relative to the absolute gain at  $1\text{ kHz}$  and  $0\text{ dB}$  ( $-0\text{ dB}$  relative to  $V_{ref}$ ).

10. The above filter specifications are for a switched-capacitor filter clock range of  $288\text{ kHz} \pm 2\%$ . For switched-capacitor filter clocks

at frequencies other than  $288\text{ kHz} \pm 2\%$ , the filter response is shifted by the ratio of switched-capacitor filter clock frequency to  $288\text{ kHz}$ .

11. The filter gain outside of the pass band is measured with respect to the gain at  $1\text{ kHz}$ . The filter gain within the pass band is measured with respect to the average gain within the pass band. The pass bands are  $300$  to  $3400\text{ Hz}$  and  $0$  to  $3400\text{ Hz}$  for the band pass and low-pass filters respectively.

operating characteristics over recommended operating free-air temperature range,  $V_{CC+} = 5\text{ V}$ ,  $V_{CC-} = -5\text{ V}$ ,  $V_{DD} = 5\text{ V}$

noise (measurement includes low-pass and band-pass switched-capacitor filters)

PARAMETER	TEST CONDITIONS		TYP†	MAX	UNIT
Transmit noise	Single ended	DX input = 0000000000000, constant input code	200		$\mu\text{V rms}$
			300	500	$\mu\text{V rms}$
	Differential		20		$\text{dBmco}$
Receive noise (see Note 12)	Inputs grounded, gain = 1		300	475	$\mu\text{V rms}$
			20		$\text{dBmco}$

NOTE 12. This noise is referred to the input with a buffer gain of one. If the buffer gain is two or four, the noise figure will be correspondingly reduced. The noise is computed by statistically evaluating the digital output of the A/D converter.

## timing requirements

### serial port recommended input signals

PARAMETER	MIN	MAX	UNIT
$t_c(\text{MCLK})$ Master clock cycle time	100	192	ns
$t_r(\text{MCLK})$ Master clock rise time		10	ns
$t_f(\text{MCLK})$ Master clock fall time		10	ns
Master clock duty cycle	42%	58%	
RESET pulse duration (see Note 13)	800		ns
$t_{su}(\text{DX})$ DX setup time before SCLK↓	28		ns
$t_h(\text{DX})$ DX hold time before SCLK↓	$t_c(\text{SCLK})/4$		ns

NOTE 13. RESET pulse duration is the amount of time that the reset pin is held below 0.8 V after the power supplies have reached the recommended values.

### serial port – AIC output signals

PARAMETER	MIN	TYPT	MAX	UNIT
$t_c(\text{SCLK})$ Shift clock (SCLK) cycle time	400			ns
$t_f(\text{SCLK})$ Shift clock (SCLK) fall time		50		ns
$t_r(\text{SCLK})$ Shift clock (SCLK) rise time		50		ns
Shift clock (SCLK) duty cycle	50%			
$t_d(\text{CH-FL})$ Delay from SCLK↑ to FSR/FSX↓			260	ns
$t_d(\text{CH-FH})$ Delay from SCLK↑ to FSR/FSX↑			260	ns
$t_d(\text{CH-DR})$ DR valid after SCLK↑			318	ns
$t_{dw}(\text{CH-EL})$ Delay from SCLK↑ to EODX/EODR↓ in WORD mode			280	ns
$t_{dw}(\text{CH-EH})$ Delay from SCLK↑ to EODX/EODR↑ in WORD mode			280	ns
$t_f(\text{EODX})$ EODX fall time		15		ns
$t_f(\text{EODR})$ EODR fall time		15		ns
$t_{db}(\text{CH-EL})$ Delay from SCLK↑ to EODX/EODR↓ in BYTE mode		100		ns
$t_{db}(\text{CH-EH})$ Delay from SCLK↑ to EODX/EODR↑ in BYTE mode		100		ns
$t_d(\text{MH-SL})$ Delay from MSTR CLK ↑ to SCLK↓	65	105		ns
$t_d(\text{MH-SH})$ Delay from MSTR CLK ↑ to SCLK↓	65			ns

† All typical values are at  $T_A = 25^\circ\text{C}$ .

Table 2. Gain Control Table  
(Analog Input Signal Required for Full-Scale A/D Conversion)

INPUT CONFIGURATIONS	CONTROL REGISTER BITS		ANALOG INPUT†	A/D CONVERSION RESULT
	d6	d7		
Differential configuration Analog input = IN+ – IN– = AUX IN+ – AUX IN–	1	1	±6 V	Full scale
	0	0	±3 V	Full scale
	1	0	±1.5 V	Full scale
Single-ended configuration Analog input = IN+ – ANLG GND = AUX IN+ – ANLG GND	1	1	±3 V	Half scale
	0	0	±3 V	Full scale
	1	0	±1.5 V	Full scale

† In this example,  $V_{ref}$  is assumed to be 3 V. In order to minimize distortion, it is recommended that the analog input not exceed 0.1 dB below full scale.

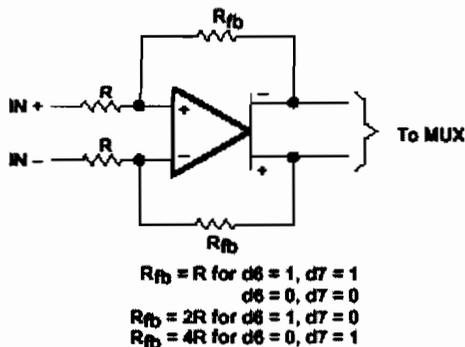


Figure 1. IN+ and IN– Gain Control Circuitry

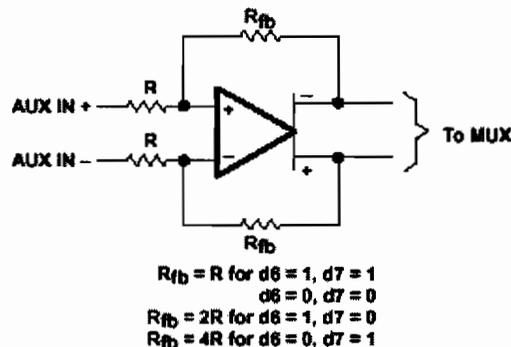


Figure 2. AUX IN+ and AUX IN– Gain Control Circuitry

### (sin x)/x correction section

The AIC does not have (sin x)/x correction circuitry after the digital-to-analog converter. (sin x)/x correction can be accomplished easily and efficiently in digital signal processor (DSP) software. Excellent correction accuracy can be achieved to a band edge of 3000 Hz by using a first-order digital correction filter. The results, which are shown on the next page, are typical of the numerical correction accuracy that can be achieved for sample rates of interest. The filter requires only seven instruction cycles per sample on the SMJ320 DSPs. With a 200-ns instruction cycle, nine instructions per sample represents an overhead factor of 1.4% and 1.7% for sampling rates of 8000 Hz and 9600 Hz, respectively. This correction will add a slight amount of group delay at the upper edge of the 300 – 3000-Hz band.

### (sin x)/x roll-off for a zero-order hold function

The (sin x)/x roll-off for the AIC DAC zero-order hold function at a band-edge frequency of 3000 Hz for the various sampling rates is shown in the following table.

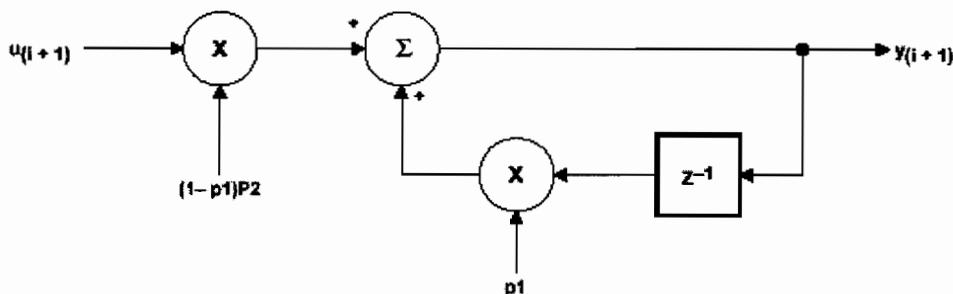
Table 3. (sin x)/x Roll-Off

$f_s$ (Hz)	$20 \log \frac{\sin \pi f/f_s}{\pi f/f_s}$ ( $f = 3000$ Hz) (dB)
7200	-2.64
8000	-2.11
9600	-1.44
14400	-0.63
19200	-0.35

Note that the actual AIC (sin x)/x roll-off will be slightly less than the above figures because the AIC has less than a 100% duty cycle hold interval.

### correction filter

To compensate for the (sin x)/x roll-off of the AIC, a first-order correction filter shown below, is recommended.



The difference equation for this correction filter is:

$$Y_i + 1 = p_2(1 - p_1)(u_i + 1) + p_1 Y_i$$

where the constant  $p_1$  determines the pole locations.

The resulting squared magnitude transfer function is:

$$|H(f)|^2 = \frac{p_2^2 (1 - p_1)^2}{1 - 2p_1 \cos(2\pi f/f_s) + p_1^2}$$

## correction results

Table 4 below shows the optimum p values and the corresponding correction results for 8000-Hz and 9600-Hz sampling rates.

**Table 4. Optimum P Values**

f (Hz)	ERROR (dB)	ERROR (dB)
	$f_s = 8000$ Hz $p1 = -0.14813$ $p2 = 0.9888$	$f_s = 9600$ Hz $p1 = -0.1307$ $p2 = 0.9951$
300	-0.099	-0.043
600	-0.089	-0.043
900	-0.054	0
1200	-0.002	0
1500	0.041	0
1800	0.079	0.043
2100	0.100	0.043
2400	0.091	0.043
2700	-0.043	0
3000	-0.102	-0.043

### IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

**CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.**

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.