TITULO DEL PROYECTO:

VOZEX PPD

(VOZ EXPERIMENTAL POR PEDRO PERDOMO DELGADO)

AUTOR DEL PROYECTO:

NOMBRE: PEDRO PERDOMO DELGADO

FIRMA:

TUTOR: JOSE MANUEL SANTOS SUAREZ

FIRMA:

INDICE:

Pág.

INTRODUCCION:

El proyecto quiere ser un intento por parte del autor, de conocer un campo de la electrónica, que no está todo lo desarrollado que se quisiera.

La síntesis de voz, está en sus comienzos; no va tan rápida como se desearía, es tan complejo trabajar con la señal de voz, que aún hoy, cerca de veinte años despues de los primeros pasos en esta matéria, los avances logrados son casi ridículos o cuando menos minúsculos comparados con otras matérias.

Lo primero que me propuse fue informarme y reflejarlo en el proyecto, aproximadamente un 60% del total. Lo segundo fue construirme el prototipo de una tarjeta, con conversores A/D yD/A para introducir la señal de voz dentro del ordenador. Un tercer intento fue teniendo el Hardware necesario, comstruir programas que hicieran mucho más flexible, la comunicación hombre máquina mediante la voz.

La idea fundamental era reducir Hardware para aumentar el Software.Por qué, se preguntarán,al reducir componentes me ahorro bastante dinero y el Software lo desarrollo yo y por tanto es más barato.

La idea de reducir Hardware en la medida en que no lleguemos a complicar en exceso el Software no es nueva, se aplica en todas las empresas.

Espero sinceramente que el proyecto ayude a introducirse en el tema, así como sirva de trampolín para proyectos mayores y mucho más ambiciosos.

P.P.D.

VISION GENERAL

Como indica el título pretendo dar una visión general pero profunda de la voz. Partiendo de un estudio fisiológico de la misma, seguido de uno físico para terminar con una muestra de las diferentes formas de trabajar con ella de una forma digital, y por tanto electrónica.

Para una mayor información recomiendo los articulos: ORDENADORES QUE HABLAN Y ESCUCHAN de Malen Ruiz-Elvira y Gonzalo Moreno, publicado en la revista CHIP Nº37 RECONOCIMIENTO DEL HABLA POR MEDIO DE ORDENADORES de Stephen E. Levinson y Mark Y.Liberman.

VISION GENERAL

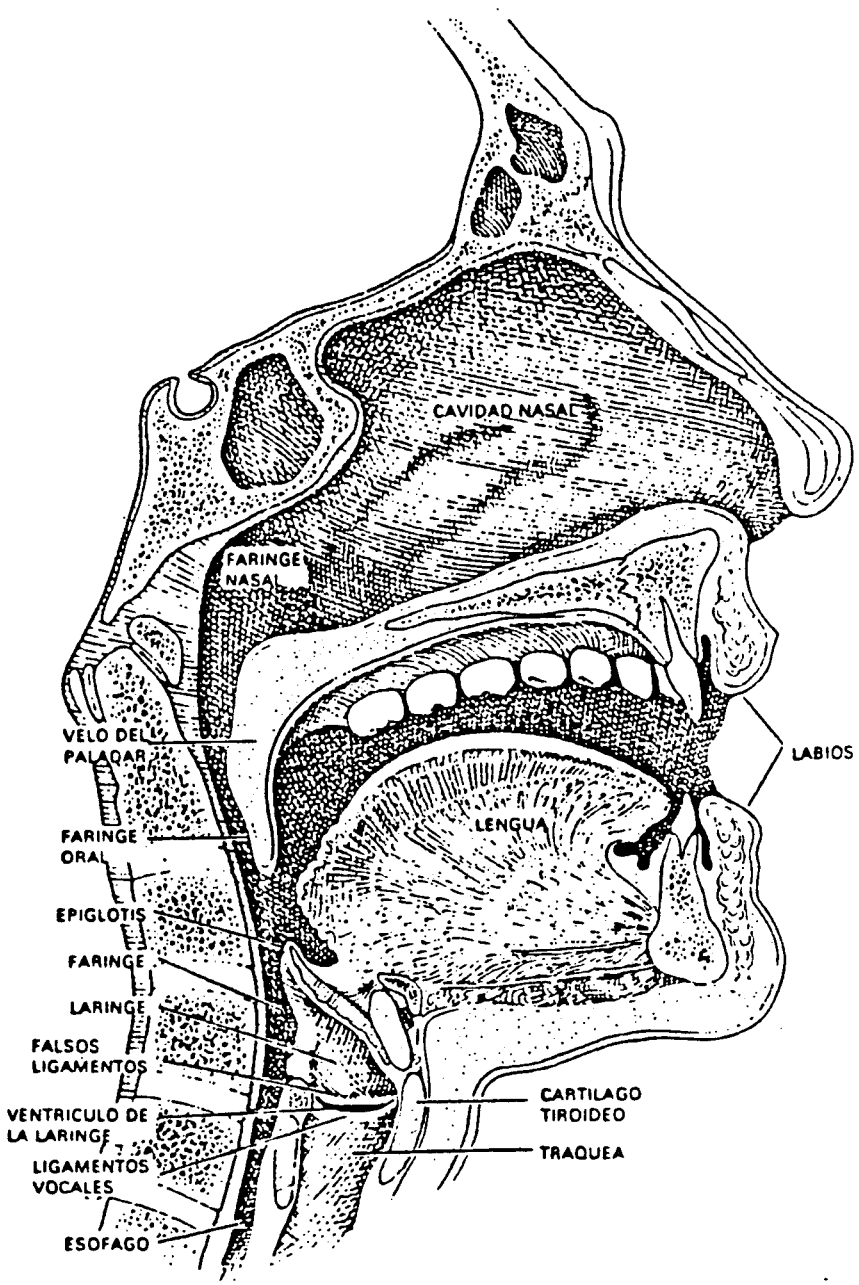
El proceso físico de la producción de la voz es realmente complejo, y la comunicación por medio de ella es algo más que simple fuerza pulmonar. Cuando el diafragma empuja el aire a través del tracto vocal se producen resonancias en la cavidad torásica, garganta y boca.

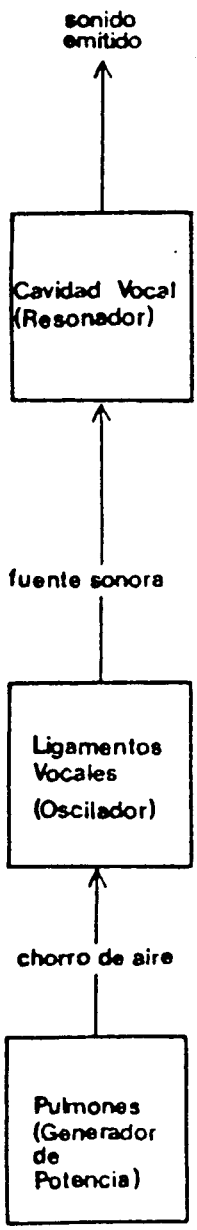
La diferencia entre voces de distintas personas se deben a tres parámetros fundamentales; volumen, frecuencia y timbre. El concepto de volumen está claro en lo que se refiere, al de frecuencia viene determinado por la velocidad de vibración de las cuerdas vocales. (Las cuerdas vocales de un triple vibran mucho más deprisa que las de un tenor) El timbre no es mas que otro nombre dado al tono vocal, es el resultado de las resonacias en pecho y garganta. Cuando pronunciamos una palabra, esta al salir por el tracto vocal , produce frecuencias secundarias llamadas (sobretonos) frecuencias que influyen en el tono y textura de la vocalización.

El habla: según la definición dada por Jane Flanagan, en su clásica obra "Análisis, síntesis y percepción del habla", es el producto final de movimientos voluntarios y no analizados de los aparatos respiratorios y masticatorio. Es una capacidad motora que requiere un aprendizaje por parte del individuo. Se desarrolla, controla y mantiene debido a la realimentación acústica por medio del aparato auditivo y la realimentación cinética de la musculatura que interviene en el proceso. La información procedente de estos sentidos se organiza y coordina en el sistema nervioso central y utilizado para dirigir la función del habla.

Es un sistema relativamente complejo.

Los pulmones y los músculos asociados del aparato respiratorio constituyen la fuente de potencia. En el caso de los sonidos vocalizados, el aire expedido hace vibrar las cuerdas vocales como una oscilación de relajación, y la corriente de aire resulta modulada en impulsos discretos.





Los sonidos no vocalizados se genera o bien al pasar la corriente de aire a través de un estrechamiento del tracto vocal, o cuando éste, después de cerrarse, se abre bruscamente. La configuración física del tracto vocal es muy variable y está dictada por las posiciomes de las partes articuladas : la mandíbula, la lengua , los labios, y el velo del paladar. Este último controla el grado de acoplamiento con el tracto nasal .

El sistema acústico equivalente está compuesto, por tubos con pérdidas y de sección variables que hacen muy difícil encontrar una ecuación de ondas del sonido propagado. Es preciso recurrir entonces a aproximaciones que consideran el tubo no uniforme como formado por numerosas secciones contiguas de diámetro fijo y distinto, y sección circular, Estas secciones tienen su equivalentes en cuadripolos elèctricos.

La obtencion de una red eléctrica al sistema vocal se basa en considerar el voltaje análogo a la presión y la corriente análoga a la velocidad polimétrica .

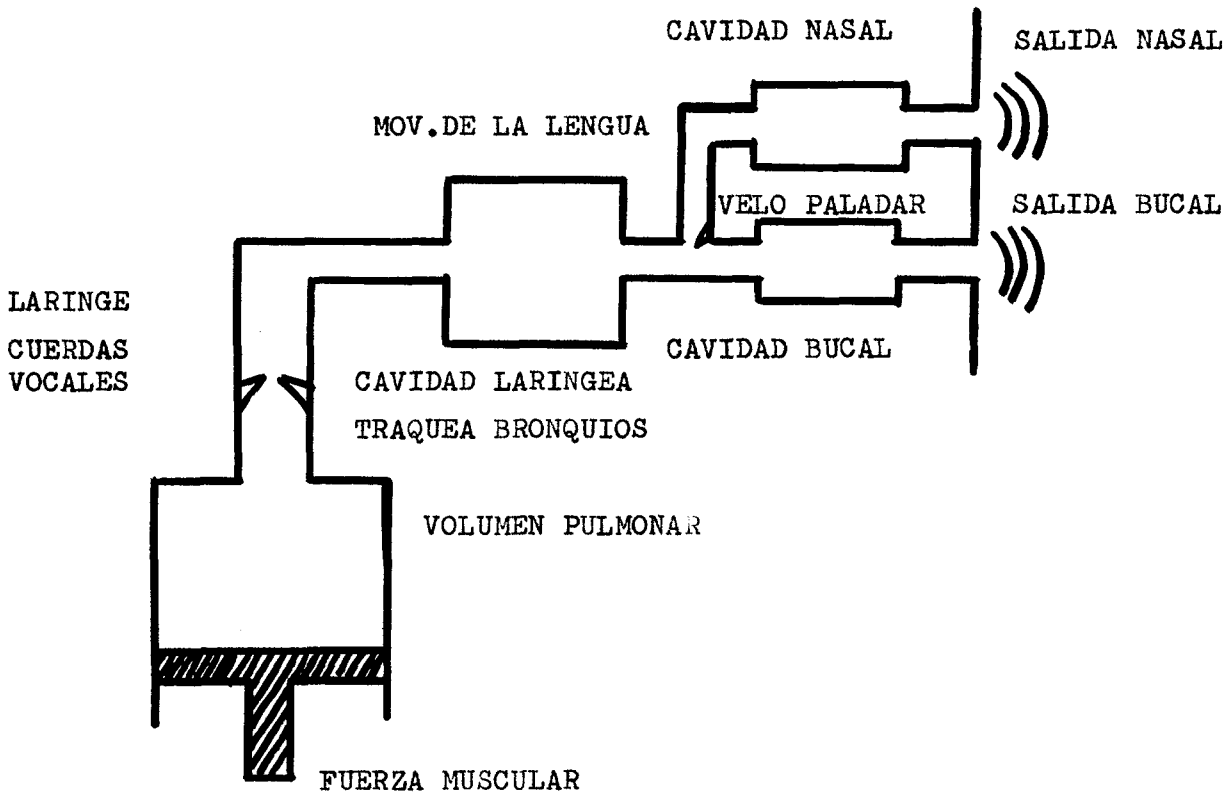
El volumen de los pulmones se puede representar por un capacitor y una resistencia. La traquea y los bronquios por una sección en T , con una impedancia en serie variable el tracto vocal está representado por cuadripolos en cascada y la línea terminada en una impedancia de carga en la boca.

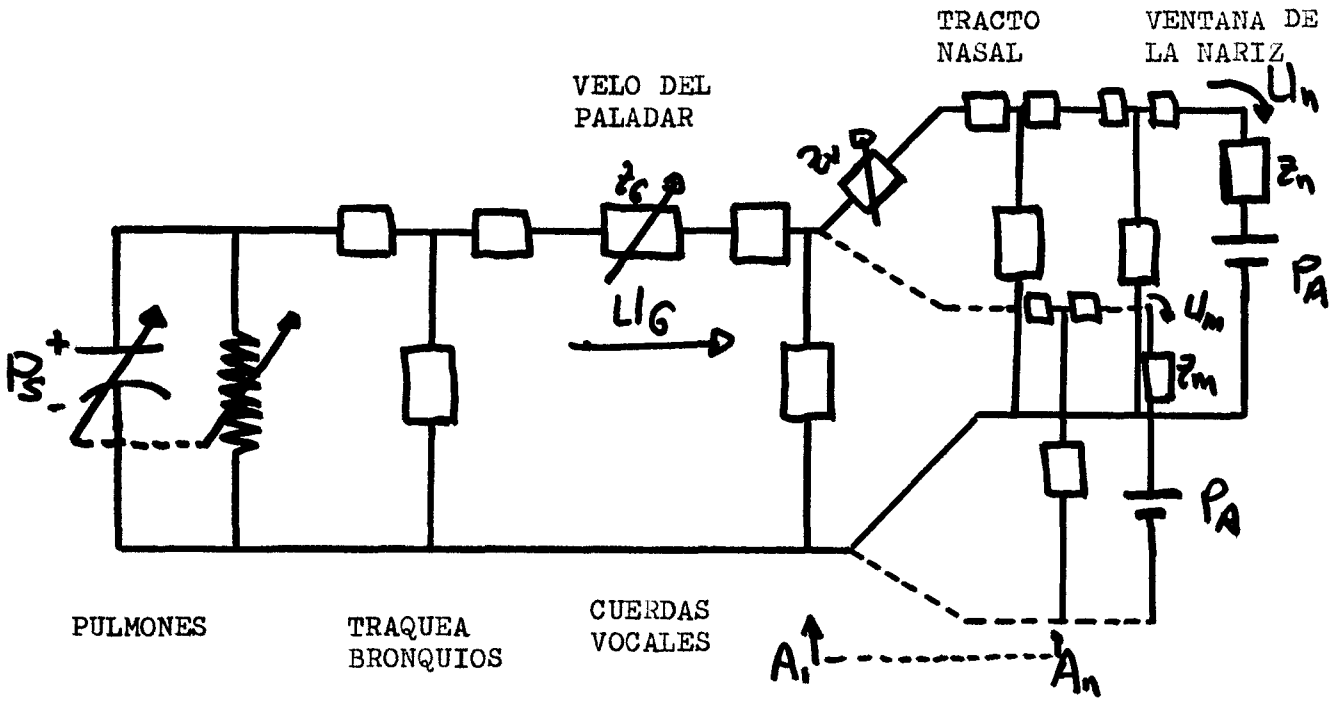
Pongo una bateria en serie con esta impedancia para representar la presión atmosférica .

El tracto nasal se representa en paralelo con el vocal acoplando ambos con una resistencia variable en función con el velo del paladar.

SIMULACION DEL SISTEMA VOCAL MEDIANTE TEORIA DE REDES

La mínima unidad del habla es el fonema, por lo que el software, que normalmente se útiliza, para la síntesis de voz en los ordenadores o en aparatos especializados en es





te tema, debe "romper" cada palabra en fonemas y luego reproducirlos encadenados. Al utilizar un micrófono para "intentar hablar" con el ordenador, este lo primero que debe hacer es convertir la voz en una corriente eléctrica donde los máximos y mínimos frecuencia definen las diferentes combinaciones entre sí, volumen y timbre. Después de este paso esta señal analógica tiene que ser codificada para que sea comprendida por el ordenador. Actualmente existen dos sistemas fundamentales de digitalización, la síntesis en el tiempo y la codificación predictiva lineal (LPC).

SINTESIS EN EL TIEMPO

Consiste en un muestreo de la señal a intervalos regulares. Esta información se convierte en ceros y unos mediante un convertidor analógico digital. En el caso de que queramos reproducirlas es decir que el ordenador nos "Hable", y en esta forma binaria, puede ser almacenada o evaluado el resultado de dicho muestreo. Unicamente tenemos que invertir el proceso, es decir hacer pasar el contenido en memoria por un convertidor digital-analógico obteniendo a la salida la señal original , la velocidad del muestreo tiene que ser como mínimo el doble de la mayor frecuencia contenida en la señal. Si consideramos que la señal de voz ocupa un ancho de banda espectoral de 4.000Hz el muestreo deberá realizarse como mínimo em 8.000Hz . Esto plantea un problema que es ¿Dónde almacenamos esta cantidad de datos.?

En los sistemas actuales claro que no, Un solo segundo de voz serían 8 Kbytes.

Si consideramos que una frase dura varios segundos nos encontramos que para una sola frase tenemos que disponer de varias decenas de Kbytes cosa imposible a no ser que dispongamos de un disco solo para ese.

CODIFICACION PREDICTIVA LINEAL

En este método se llega a un compromiso en bajar la calidad de la reproducción y disminuir el número de muestra por segundo para un mejor almacenamiento. Al muestrear por debajo de la frecuencia de nuestro Nyquist. Se pierde información mediante la codificación predictiva el ordenador "Supone" que información se pierde y lo prevee a la hora de reproducir la voz se la suma a las muestras hechas.

Reproduciendo totalmente la señal de la voz con una calidad, eso sí, menor que el método anterior.

Sólo se necesitan 100 bytes de memoria para un segundo esto que quiere decir que muestreamos a una frecuencia de 100Hz o de 100muestras por segundo.

RECONOCIMIENTO DE VOZ

Victor Zue, experto en reconocimiento de voz y reproducción de la voz del Massachussetts Institute of Technology (MIT) afirma que el desarrollo de un verdadero reconocimiento de la voz todavía tardará décadas en producirse, aunque en los próximos años se realizen grandes avances en aplicaciones de tratamiento de voz. Estas aplicaciones se subdividen en dos grandes grupos: las que depen - den de la persona que habla y aquellas otras que son independientes de este factor.

En el primer caso el usuario crea un vocabulario hablando a un micrófono, hace varias pasadas sobre la misma palabra, de modo que el ordenador pueda registrar el mayor número posible de inflecciones de voz.

estas pasadas sucesivas crean una especie de molde, es un patrón de la voz, contra el cual van a compararse todas las ordenes habladas que se emitan en el futuro. Este método utilizado con recursos de calidad logra un 90% de presición, es decir de cada 10 veces en ocasiones el ordenador reconocerá la palabra y ejecutara la instrucción correspondientes.

Las bibliotecas de palabras no esceden de 100 a 200 y además existe la posibilidad de que un simple resfriado haga que el ordenador no reconozca la voz de su usuario.

El mayor problema es que sólo puede ser utilizado por una persona, y ésta es la que ha creado el vocabulario.

Los sistemas que son independientes de la voz del usuario tienen una mayor probabilidad de reconocimiento de ésta, ya que pueden llegar hasta el 95% de fiabilidad de ésta.

Son capaces de reconocer cualquier tipo de voz o acento, pero tienen un vocabulario limitadísimo, en la mayoría de los casos, es simplemente reconocer si o no y los nº del cero al nueve o palabras igualmente simple.

GENERACION AUTOMATICA DE VOZ

La generación de la voz es una tecnologia que apareció hace unos 20 años, pero han sido los últimos desarrollos en la síntesis de voz los que han producido los mayores avances en este campo. Existen actualmente tres tipos fundamentales de sistemas para la generación de voz: síntesis de voz, codificación de la voz y almacenamiento y recuperación de la voz (Store and Forward).

Los sistemas de sintesis de la voz se basan en una memoria digital de estructuras fonéticas, la aproximación digital a la síntesis de voz permite la creación de un vocabulario de trabajo de más de 300.000 palabras, muy

superior al vocabulario inglés(hay que tener presente que la mayoria de estos trabajos se realizan en inglés) el inglés de habla cotidiana tiene unas 50.000 palabras.

Los sistemas de síntesis de voz se pueden basar en la conversación digital de una voz humana o bien en la implementación de un modelo de tracto vocal.La síntesis de voz forma las respuestas directamente partiendo de los fonemas. Las estructuras físicas que este tipo de sintetizadores deben reconstruir con toda presición incluyen un modelo eléctrico del tracto vocal humano, un programa que espesifique el sonido deseado de ese tracto vocal y el interface de control del tracto.

El problema fundamental que se nos presenta al sintetizar la voz es la corecta codificación de los fonemas. Utilizando todos los fonemas posibles e instrucciones para conseguir diferentes énfasis se puede reconstruir cualquier frase.

Hay otro método de síntesis de voz, que reducen drásticamente las necesidades de almacenamiento , pudiendo utilizar un vocabulario casi ilimitado. Es la conversión texto a fonema. Este método utiliza la codificación directa de los fonemas, el problema incide en que la calidad de voz no es muy buena.

Las representaciones paramétricas tales como la codificación predictiva lineal (LPC) o la autocorelación parcial (PARCOR) utilizar el hecho de la lenta variación en el tiempo de los parámetros que subyacen en la voz.

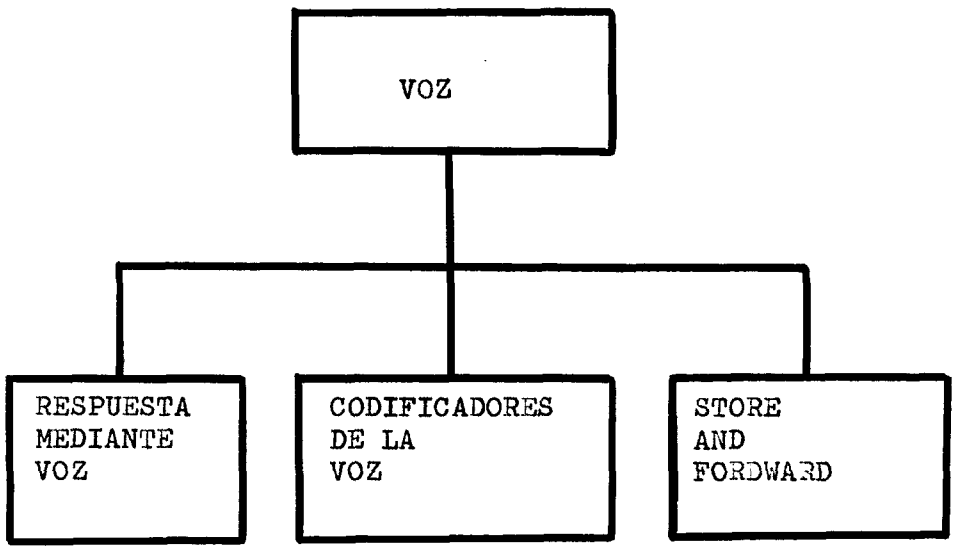
En la actualidad la mitad de los fabricantes de Chips para la síntesis de voz utilizan la tecnica (LPC) para evaluar los parámetros de la señal de voz, y para representarla. Esto se debe al ahorro en utilización de memoria y velocidad de trasmisión de datos. Además esta técnica es integrable en un solo Chip usando tecnologia CSI.

la codificación analógica precisa de velocidades de trasmisión de 100 bytes por segundo o incluso más.

La señal de voz es codificada por muestreo,se comprime para eliminar redundancias de simetria y silencios. Un hecho importante es el ajuste en fase, luego se almacena en memoria. Para regenerar sólo hay que invertir el proceso

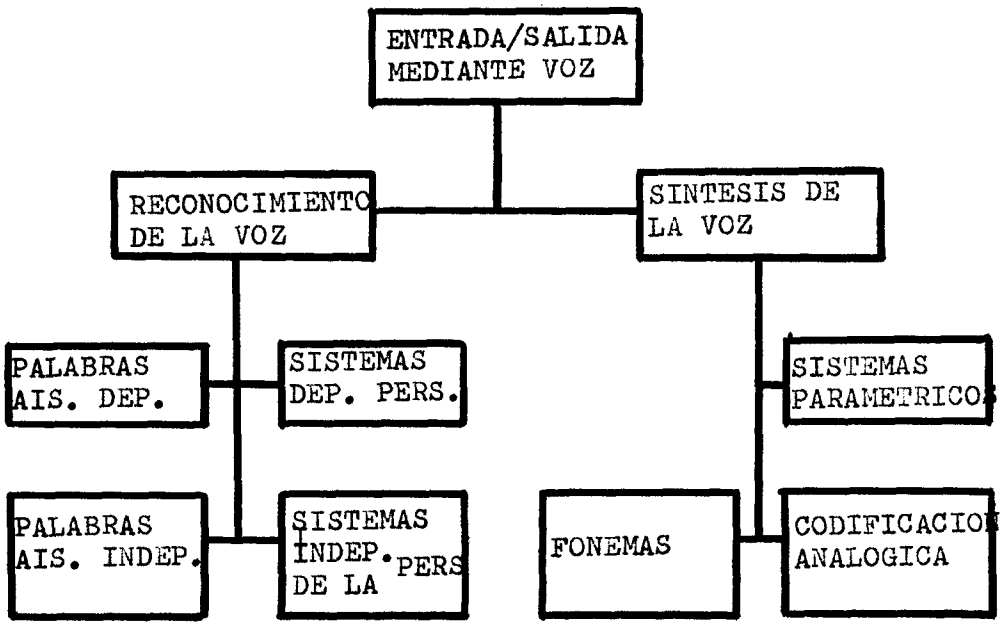
La calidad de voz es muy superior a las anteriores métodos, pero el vocabulario queda limitado a la capacidad de la memoria.

La tecnología de la voz puede ser dividida en tres grandes áreas.



Los sistemas con entrada/salida mediante voz,precisa el reconocimiento de la misma,mediante métodos dependientes o independientes de la persona,bien de palabras aisladas ode frases completas. La otra parte del proceso es la síntesis de la voz que puede realzarse mediante sistemas

que usan fonemas pregrabados, o que hacen una calificación lineal predictiva (LPC) o, bien, mediante una codificación de la señal analógica (modulación por impulsos codificados) PCM, o modulación delta por impulsos codificados o ADM.



Según Stephen E. Levinson y Mark Y. Liberman en su artículo "Reconocimiento del habla por medio de ordenadores" publicado en Cientifican American. Diseñar una máquina que escuche es mucho más difícil que construir una que hable. Sólo con una mejor comprensión de los modelos humanos del habla habrá progresos significativos en su reconocimiento.

automático.

El avance de la ciencia exige ahora la necesidad de comunicación hombre o máquina por medios más naturales. La evolución ha hecho el lenguaje hablado muy acorde con las necesidades de la comunicación humana. Es muy rápido y casi no se requiere esfuerzo, tampoco es necesario el contacto físico o visual y apenas limita la movilidad del cuerpo o el uso de las manos.

Una máquina capaz de reconocer el habla humana podría cambiar todas estas ventajas con los poderes del ordenador. De este modo podríamos acceder a los grandes bancos de datos con solo comunicarnos por teléfono, controlar complejos ordenadores y realizar prodigiosos artilugios proteticos para minusválidos.

Sin embargo, a pesar de más de 40 años de investigación, el reconocimiento automático del habla natural o conversacional sigue siendo un objetivo utópico.

Ya hay comenzados aparatos que reconocen entre 10 y 30 palabras. En los más avanzados laboratorios de investigación hay reconocedores del habla con vocabulario que alcanzan hasta 1.000 palabras, sistemas que reconocen oraciones a partir de un vocabulario limitado con pausas breve entre palabras, e incluso sistemas que reconocen el habla conexa con bastante precisión , si el vocabulario es corto, la sintaxis reducida y el hablante cuidadoso.

¿Por que es tan inexpugnable el reconocimiento?.

La dificultad rodica en los complejísimos y variables medios con que los mensajes lingüisticos se codifican en el habla.

La lengua que hablamos nos permite expresar nuestros pensamientos en forma de sonidos y captar mensajes de otras personas a través de los sonidos que producen. Este entendimiento necesita de un acuerdo mutuo sobre el código a emplear, debe existir por tanto un cierto entramado conceptoral. Hablar y lo que es más importante com-

prender requiere un sistema común que codifique mensajes en sonidos y descodifique sonidos del habla para que esto tengan significados. Es decir, se necesita tener el mismo idioma.

En cierto modo el ordenador es un interlocutor más.El en determinados momentos, tiene ideas, cosas o simples datos que quiere comunicarnos.

El hecho de que existan lenguas como la Inglesa o la Española no nos estraña, tampoco nos extraña que el ordenador-individuo se comuniquen con lenguajes como el BASIC, COBOL, FORTRAN.. las primeras son lenguas naturales, y quedan implicitamente definidos por el uso diarios. Los segundos son lenguajes formales, compuestos por un conjunto explícito de reglas especialmente estipuladas por perítos. Al menos por ahora, los ordenadores solo ejecutan aquello para lo que estan programados . No viven en el mundo de los humanos ni aprenden nada a partir de la experiencia cotidiana. Es por tanto por lo que el conocer una lengua natural por el ordenador debe de hacerse de una forma explícita y precisa de la misma.

En todos los sistemas existentes de reconocimiento del habla actualmente concebidos la descripción formal de una lengua natural solo cubre un fragmento de la lengua en cuestión.

El núcleo del habla humano está constituido por la palabra. Las secuencias de palabras suelen disponerse en frases deacuerdo con unos principios convinatorios conocidos por el nombre de sintaxis, estas secuencias constituyen el deseo de expresar algo. El hecho de que las palabras formen parte normalmente de un discurso, coherente puede tener su utilidad en el reconocimiento de las mismas, por cuanto proporcionan un contexto en el que cierta palabras son más probables que otras. Ahora bien, es extremadamente dificil hacer que un ordenador actue como si fuese capaz de comprender las secuencias de palabras.

El problema no sólo afecta a las relaciones entre las palabras, sino tambien a la tarea de reconocer y razonar acerca de la naturaleza del mundo.

Aunque la capacidad de comprender la lengua puede ser el fín último, la empresa del reconocimiento del habla se funda realmente en la identificación de las palabras. Y la parte de las palabras que se interesa aquí estudiar es su sonido.

El conjunto de todos los sonidos que a pesar de ser diferentes representan (en el contexto de su enunciación) una misma unidad léxica. El problema de lograr reconocer una palabra consiste en hallar un espacio matemático definido en el que queda y pueda delimitarse efectivamente tal conjunto de sonidos. Puesto que el grado de variación dentro del conjunto de sonidos. Puesto que el grado de variación dentro del conjunto de sonidos correspondientes a una palabra dada es muy amplio, muy pequeña la distinción acústica entre palabras diferentes y como, en fin, el hablante adulto normal puede disponer de unas 100.000 palabras o más, el problema resulta verdaderamente peliagudo.

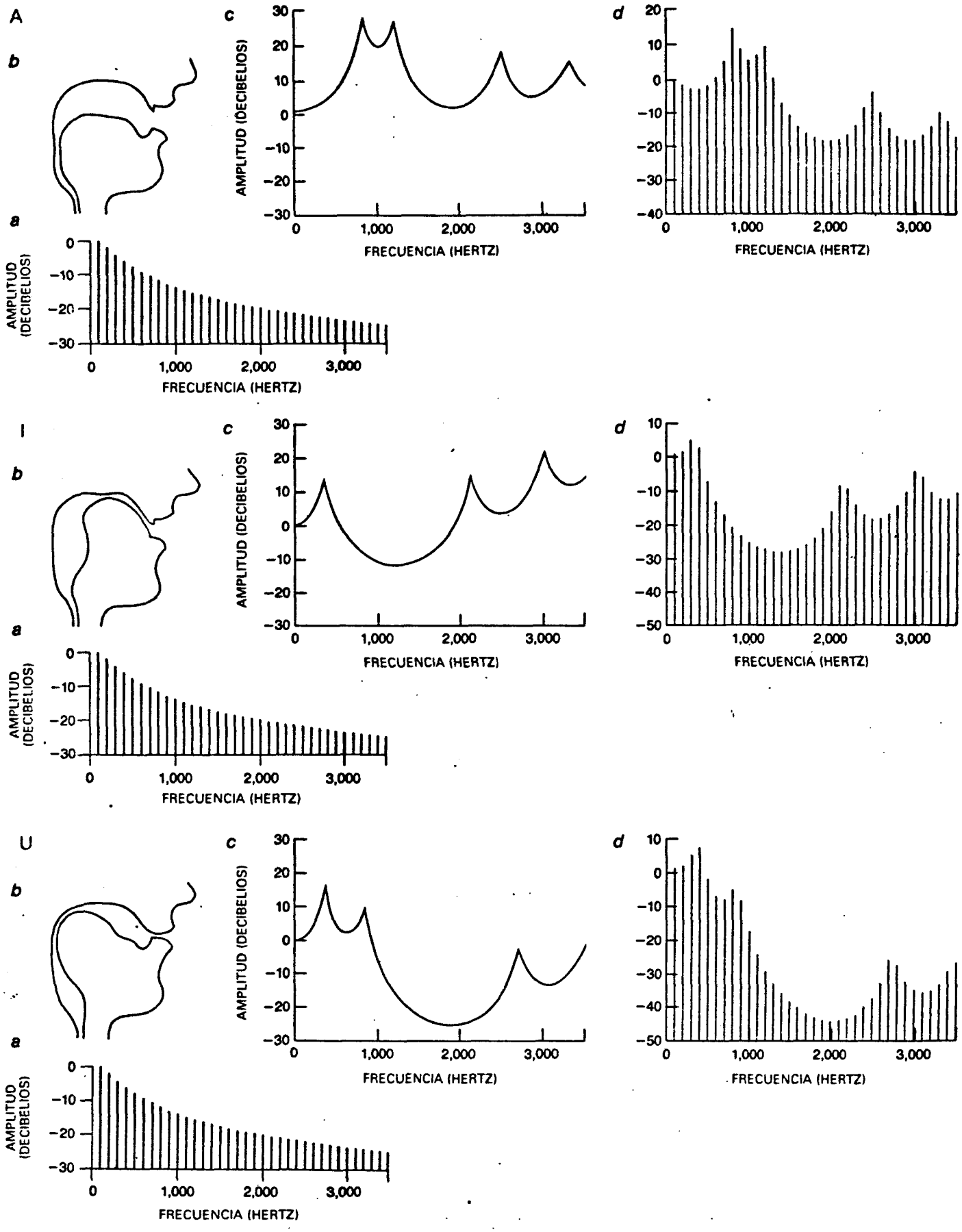
Para llegar, a comprender las razones por las cuales la variación en el sonido de una palabra y la naturaleza de la distinción entre una palabra y otra es necesario que sean explicadas unas cuantas cosas. En un primer lugar tenemos que comprender el medio básico de la comunicación hablada, la forma como el aparato vocal humano puede producir las alteraciones acústicas aéreas y la forma como el sistema auditivo puede percibirlas, En un segundo lugar, tenemos que admitir que los sonidos del habla son elementos de un sistema fonológico peculiar para cada lengua. Todo sistema fonológico limita la manera en que las diversas palabras pueden diferir y controlar, en parte, el modo en que puede variar la pronunciación. Cuando hablamos una cantidad de aire alojado en los pulmones pasa de la laringe a la garganta y ahí sale expedida a través de la boca. Si la úvula (la campanilla que pende al fondo del paladar).

está baja, la corriente del aire sale también por la naríz; si está levantada, los conductos nasales quedan bloqueados. La corriente de aire puede bloquearse al cerrar los labios , apretando la lengua contra el paladar o cerrando la glótis, un órgano que consta de dos pliegues paralelos de tejido blando (las cuerdas vocales) situadas dentro de la laringe.

la corriente de aire puede potenciar el sonido de tres maneras básicas a lo largo del conducto vocal, la primera haciendo vibrar las cuerdas de una forma más o menos igual a la doble lengüeta de un oboe o un bajón.

Cuando las cuerdas vocales se unen pasan al aire procedente de los pulmones, con lo que aumenta la presión debajo de ellas. Esta presión provoca la separación de las cuerdas, pero la velocidad del aire que se precipita hacia el exterior reduce la presión en el espacio intermedio. El descenso de presión y la elasticidad de los tejidos vuelven a unir las cuerdas vocales y con ello se inicia de nuevo la presión. La rapidez con qué se repite este ciclo constituye la frecuencia fundamental de la voz, que se manifiesta acústicamente a base del tono.

¿Que efecto acústico ejerce la forma del conducto vocal sobre el sentido emitido? Cuando se presentan los sonidos a partir de su espectro de amplitud, los efectos son claros (véase la ilustración) pag. 42 Scientifican Américam" . El conducto vocal actua como un filtro sobre el espectro de la fuente fónica, intensificando algunas frecuencias y disminuyendo otras. La filtración selectiva tambien puede describirse a base de una expresión matemática llamada función de transferencia, de modo que cada una de estas funciones se asocia a cada una de las posiciones que adquieren los órganos del conducto vocal.La función de la transferencia suele tener varias cimas frecuenciales, denominadas formantes, en las que se concentra la mayor parte de la energía procedente de la fuente de sonido.



LOS SONIDOS VOCALICOS derivan de diversas configuraciones de la boca, los labios, la lengua y el velo (o paladar blando). La forma resultante del conducto vocal puede concebirse como una serie de cavidades de resonancia que aumentan la energía en ciertas frecuencias y la disminuyen en otras, siguiendo unas pautas predictibles. Estas características de respuesta a unos filtros pueden representarse por una función de transferencia (*c*) para cada posición del modelo del conducto vocal (*b*). Cuando la energía sónica de entrada es periódica, tanto el espectro de entrada (*a*) como el de salida (*d*) son espectros de líneas. En un espectro de líneas, la energía sónica se concentra en armónicos, o múltiplos enteros, de la frecuencia de las cuerdas vocales. Una fuente de sonido aperiódica como la de una vocal cuchicheada no presenta líneas discretas en el espectro, pero la forma del espectro de salida todavía se corresponde con el de la función de transferencia. En la figura aparecen las configuraciones modélicas del conducto vocal para las vocales "i", "a" y "u".

Las interrupciones del espectro, los altibajos de energia de ciertas bandas frecuenciales y otros signos acústicos facilitan claves sobre eventos articulatorios: el cierre o abertura del conducto vocal o el comienzo o final de la vibración laríngea. Esto nos puede sugerir el realizar la segmentación y roturación a partir de las unidades fonológicas básicas de que constan las palabras.

La mezcla y fuminación de información acústica esta más acusadas en las unidades más pequeñas que en las palabras.

Hay razones de peso que apoyan la segmentación en unidades más pequeñas a medida que aumenta el el vocabulario

Hay en Inglés unas 300.000 palabras, unas 20.000 sílabas y unos 40 fonemas (consonantes y vocales). Los cuales a su vez, pueden descomponerse en una docena de rasgos fonológicos que establecen las carasterísticas distintivas de la forma del conducto vocal y el control de la laringe

No obstante a medida que decrece el número de unidades lingüisticas su relación con las pautas de sonido se vuelven más abstractas, más complejas y peor comprendidas. La segmentación de las unidades mediante las técnicas hoy conocidas conduce a altos indices de error. Y sin embargo si las limitaciones que impone el código lingüistico pueden compemsar los errores o si se llegan a encontrar métodos más fiables de análisis, el reducido número de unidades fonológicas básicas brindará una ventaja decisiva en favor de estos elementos fundamentales.

MICROFONEMAS

Desde el punto de vista fonético la unidad más elemental del lenguaje es el fonema. Sin embargo examinando detalladamente la forma de onda correspondiente a un fonema podemos llegar a la definición de una unidad más elemental, el microfonema, apropiada para la síntaxis de palabras aisladas y de comprensión de la información necesaria a dicha síntesis.

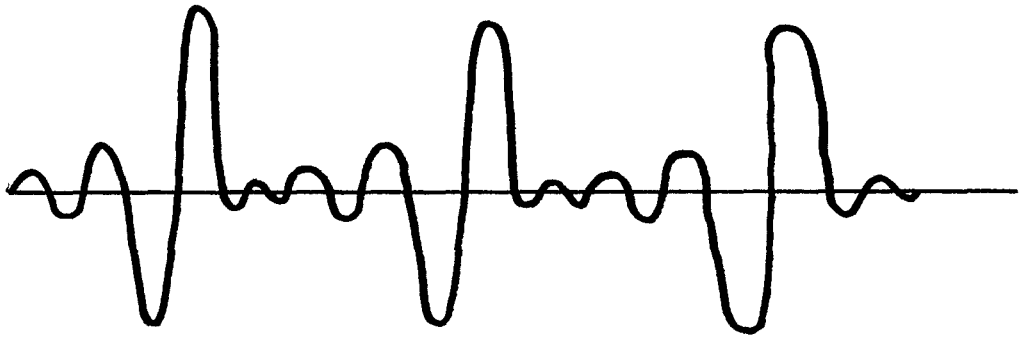
En la definición de microfonema distinguimos tres casos, según el tipo de articulación de fonema correspondiente

a) FONEMAS SONOROS

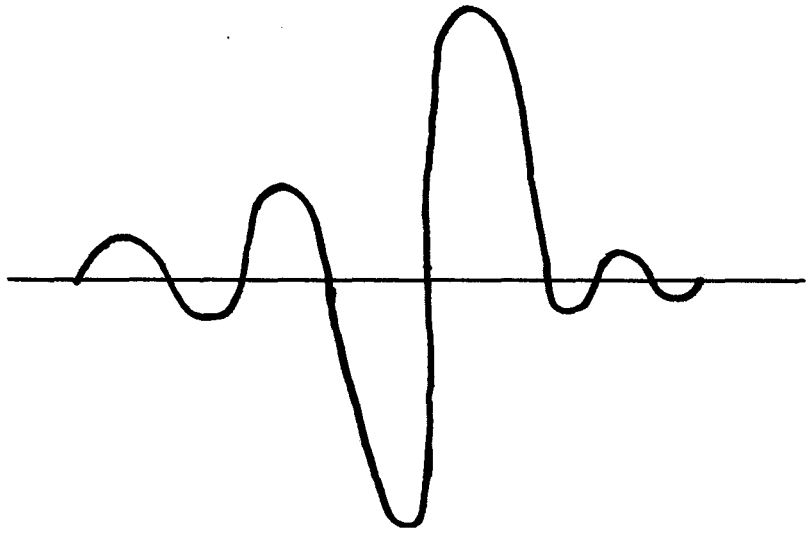
la forma de onda es casi periódica (ver figura 6a) en perfecta consonancia en la vibración de las cuerdas vocales. En este caso el microfonema será un periódo básico, o intervalo que va de dos aperturas consecutivas de las cuerdas vocales (ver figura 6) .

a) Por ejemplo

segmento de la forma de la palabra dos correspondiente /O/.



b) microfonema correspondiente a /O/.



b) FONEMAS OCLUSIVOS

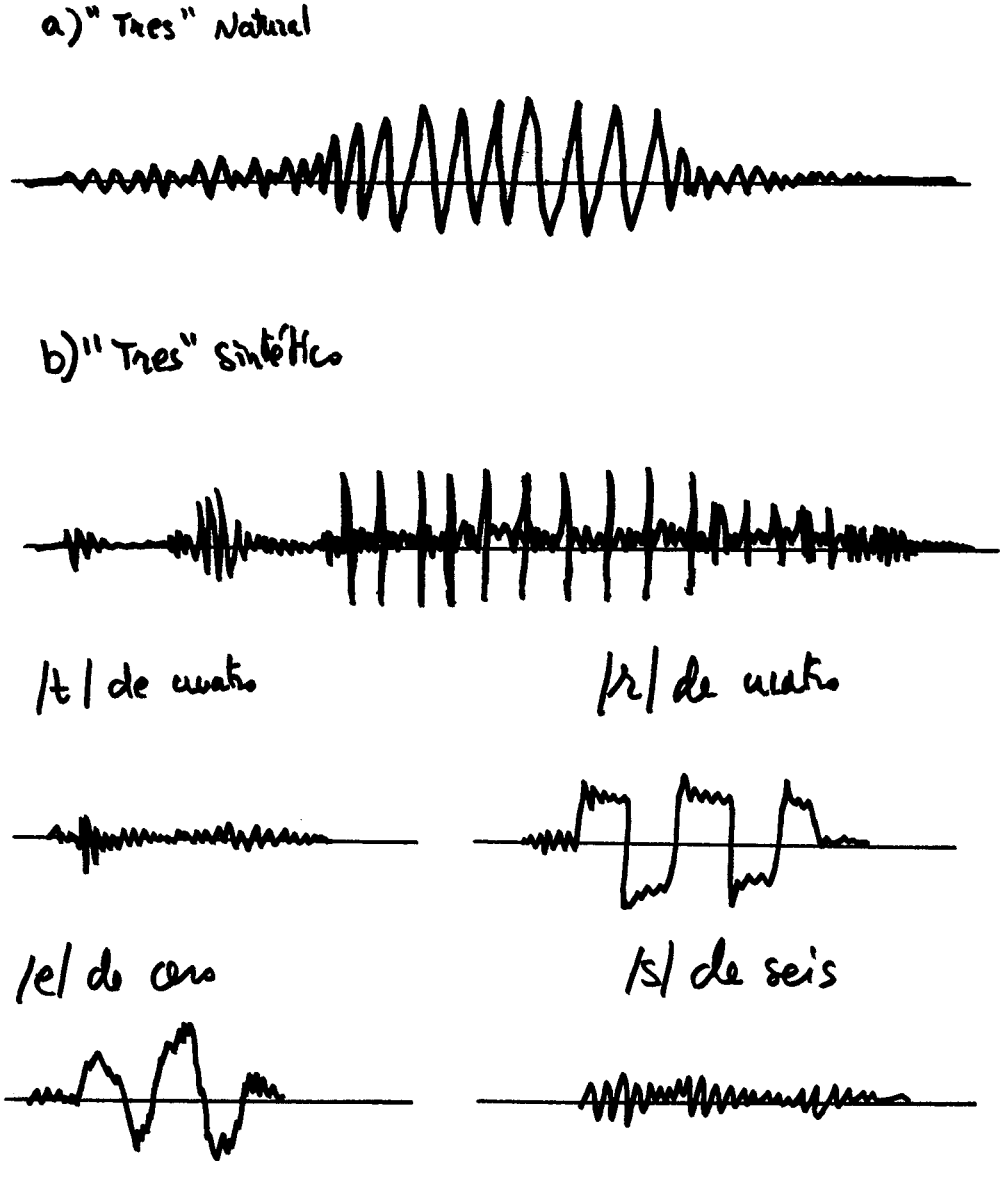
Estos sonidos se originan por una excitación transitoria del aparato fonador y son de corta duración El microfonema y el fonema coinciden (ver /4/ en fig.2a)

c) FONEMAS FRICATIVOS SORDOS

La codificación adecuada de los fonemas es la más compleja. Por un lado son de larga duración y por otro su estructura es muy aleatoria.

La solución adoptada ha sido la de tomar un segmento de la forma de onda como microfonema representativo.

La longitud de este segmento requiere un compromiso entre calidad y cantidad de información a almacenar (ver /S/ en fig. 2a



El número de bits necesario para la codificación PCM de los microfonemas depende de la frecuencia de muestreo, 10KHz y un rango de tonalidades de Sa 10ms son necesarias de 50 a 100 muestras por microfonemas.

Las síntesis de las palabras palabras aisladas mediante microfonemas se efectuan mediante concatenación de éstos con las amplitudes adecuadas. En la fig. 2a se representa la gráfica de la forma de onda de la palabra " TRES" pueden distinguirse los cuatros segmentos correspondientes a los fonemas /t/, /r/, /e/, y /s/ con cierta modulación de amplitud puede esta sintetizarse según y muestra la fig. 2b. En la fig. 2c puede verse las formas de ondas de los microfonemas a partir de los cuales se síntetiza "TRES", así como su palabra de origen.

Resumiendo puedo decir que la síntesis por articulación de microfonemas se efectua a partir de la codificación en PCM de los microfonemas y de una especificación que comprende tipos de fonemas, duración y envolvente de amplitud, para cada palabra de vocabulario a síntetizar. La mayor ventaja como se puede ver es la reducción drástica de la memoria utilizada.

ANALISIS DE SEÑAL DE VOZ

Con el procesado digital de la señal de voz se pretende obtener una representación más conveniente o más fácil y útil de la información transportada por la señal. Representando la señal de voz de otra forma se consiguen facilidades tales como determinar si la voz es sonora o sorda y dicernir entre voz y silencio (ruido).

La suposición básica realizada al análizar la señal de voz consiste en considerarla como una señal cuasiestacionario en segmentos cortos de duración entre 10 y 30 milisegundos. Esto permite aplicar métodos de procesado localizado . Este procesado proporciona una secuencia temporal de números que representan uno o más parámetros de la señal de voz.

Estas representaciones paramètricas ponen de manifiesto mucha información de la voz que en su estado de forma forma de onda no es posible, en general, evidencia.

En el apartado de introduciôn se señaló que la primera tarea de cualquier sistema de reconocimiento de voz en extraer los parámetros de esta para así formar los patrones de test y de referencia de las palabras a ser conocidas . En este apartado se presentan métodos para obtener estos parámetros tanto como para el dominio del tiempo como de la frecuencia.

ANALISIS DEL DOMINIO DEL TIEMPO

Consta de un conjunto de técnicas de procesado denominado métodos temporales. Esto viene a significar que dichos métodos comtemplan directamente la onda de la señal de voz, una de ellas es.

VELOCIDAD DE CRUCES POR CERO, MEDIDA DEL CONTENIDO DE FRECUENCIA DE LA SEÑAL DE VOZ

Para las señales discretas, decimos que ocurre un cruce por cero si muestras sucesivas tienen distintos signos algebraicos. Por tanto la velocidad de cruces por cero da una medida sencilla de contenido de frecuencia de la señal de voz. Una definición matemática de la velocidad de cruces por cero puede ser.

$$VCC = \sum_{m=0}^{N-1} \left| \text{Sign}\{s(m)\} - \text{Sign}\{s(n)\} \right| \omega(n-m)$$

donde $$\text{Sign}\{s(p)\} = \begin{cases} 1 & \text{Si } s(p) \geq 0 \\ -1 & \text{Si } s(p) < 0 \end{cases}$$

y la ventana ω:

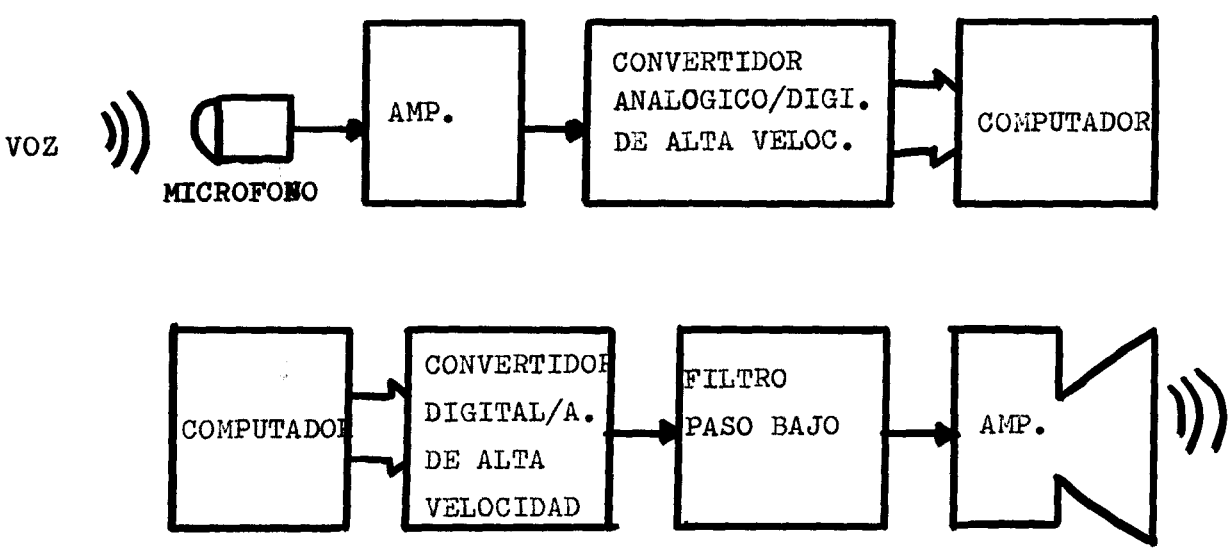
$$\omega(p) = \begin{cases} \frac{1}{2N} & 0 \leq p \leq N-1 \\ 0 & \text{en otro caso.} \end{cases}$$

La longitud promedio N de la ventana está comprendida entre 100 y 300 muestras por una velocidad de muestreo de 8 KHz.

Puede demostrarse que la energia de la voz sonora está concentrada por debajo de los 3KHz a causa de que el espectro introducido por la onda global decae y que la energia de la voz sorda se encuentra a frecuencias más altas.

Ya que frecuencias altas implican alta velocidad de cruces por cero y bajas lo contrario, existe una estrecha relación entre la velocidad de cruces por cero y la distribución de la energia con la frecuencia. Por tanto una razonable generalización es que; si la velocidad de cruces por cero es alta la señal de voz es sorda y si es baja sonora . No obstante esto queda algo impreciso al no indicarse que se entiende por baja y por alta.

Es una primera parte. Voy a intentar digitalizar la señal de voz para poderla almacenar y después sacarla y reproducirla. Un esquema sencillo podra ser.



ONDA DE AMPLITUD DE LA VOZ

En esta parte del proyecto, ahondo en la base matemática que sustenta y hace posible el muestreo de una señal y su posterior recuperación. Nombro muy brevemente un típico ejemplo de aplicación, la técnica MIC. Para una mayor información recomiendo los libros: SISTEMAS DE COMUNICACION de B.P.LATHI, TECNICA DE CODIFICACION MIC del autor de este proyecto.

ONDA DE AMPLITUD DE LA VOZ

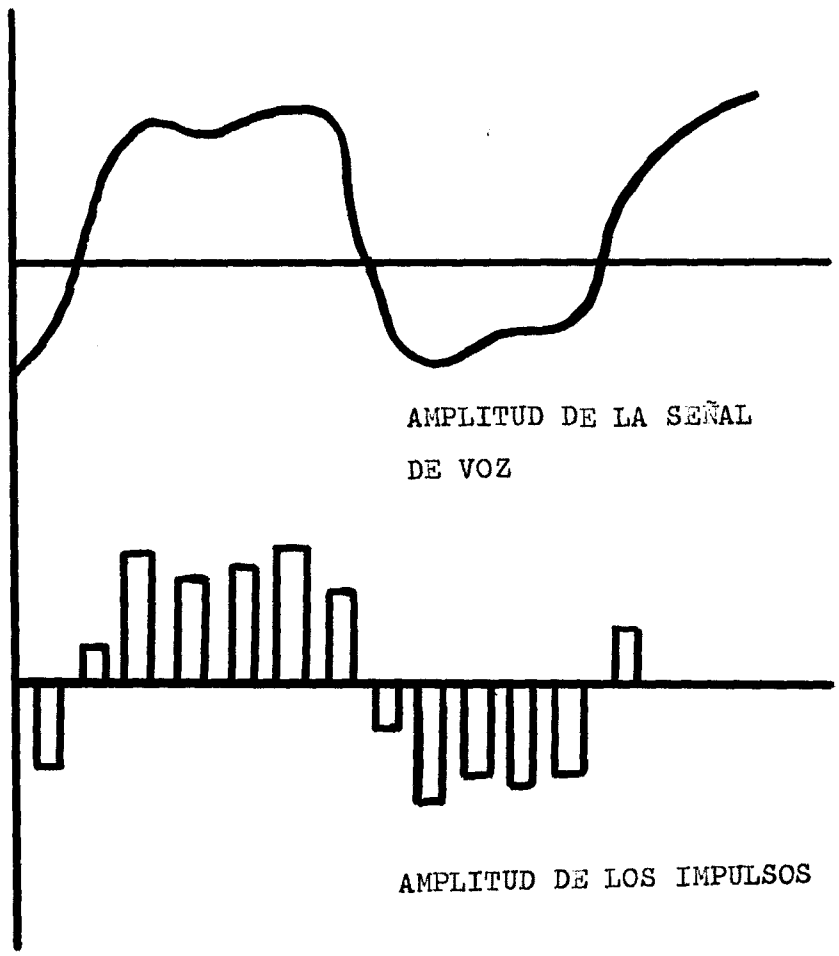
Impulsos equivalentes representativos de esa amplitud. Un paso previo para digitalizar la señal de voz es, tomar muestras discretas de ella, con ello obtenemos un tren de pulsos con amplitud variable y con posibilidad de polaridad positiva y negativa. Si estas muestras han sido tomadas cumpliendo el teorema de Nyquist se puede recuperar la señal original con un simple paso bajo. A groso modo podríamos explicarlo diciendo que al pasr de la señal de voz (de ancho de banda de 4KHz aprósimadamente) a un tren de pulsos (por tratar de pulsos tendrá componentes de alta frecuencia).

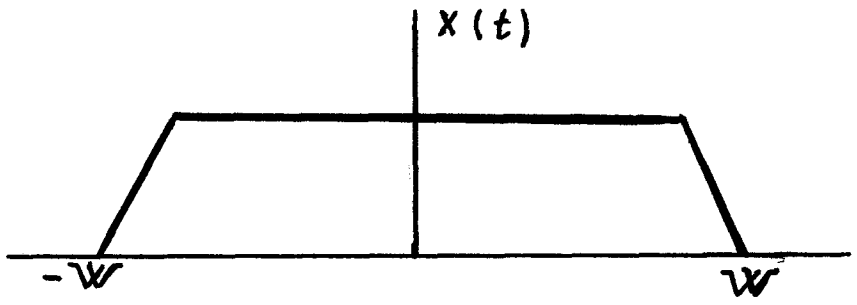
Al utilizar el filtro paso bajo esta señal de pulsos , sucede:

En el dominio de la frecuencia un retorno del ancho de banda de las altas a las bajas frecuencias , en el dominio del tiempo la unión de todos los picos de los pulsos en lo que se produce la recuperación de la señal de voz.

Podra pensarse, a primera vista que el tren de impulsos así obtenido contiene solo apróximadamente la información a transmitir . Sin embargo no puede así, ya que según el teorema de las muestras podremos fijar el número mínimo de impulsos que se nesecitan para presentar con toda esactitud la señal original. Su enunciado dice así.

"Si una información es una magnitud función del tiempo se muestra instantáneamente a intervalos regulares y con una frecuencia que sea al menos dos veces la frecuencia significativa más alta de la información , las muestras obtenidas contienen toda la información original". Consideremos la señalpaso-bajo y de energía finita x (t) cuyo espectro x (f) es el de la figura y cumple X (f)=0 para /f/> W.





Es posible establecer un desarrollo en serie de Fourier de X (f), limitado a /f/< W, del modo siguiente.

$$X(t) = \sum_n \cdot C_n \cdot e^{j2\pi n \frac{t}{2W}} \qquad (f = W)$$

Los coeficientes Cn del desarrollo, vendrán datos por:

$$C_n = \frac{1}{2W} \int_{-W}^{W} X(f) \cdot e^{-j2\pi n \cdot \frac{t}{2W}} df.$$

Ahora bien, si x(t) es la transformad inversa de X(f):

$$X(t) = \int_{-W}^{W} X(f) \cdot e^{j2\pi f t} df.$$

de donde se infiere una relación imediata entre los cn y los valores de Cn y los valores particulares de X(t); concretamente.

$$C_n = \frac{1}{2W} \cdot X\left(-\frac{n}{2W}\right)$$

Asi pues puede escribirse el espectro X(f) de x(t) en términos de las propias muestras x(- n/2W de x(t) sin más que sustituir los valores de Cn dados por:

$$X(f) = \sum_n \frac{1}{2W} x\left(\frac{n}{2W}\right) e^{-j2\pi n \frac{f}{2W}} \qquad [f] = W$$

(hemos cambiado n por -n lo cual es variable por la simetría del sumatorio). Ypara todo f.

$$X(f) = \sum_n x \frac{1}{2W} x\left(\frac{n}{2W}\right) e^{j2\pi n \frac{f}{2W}}$$

Especificada X (f) en términos de las muestras de x(t) es inmediato encontrar la propia x(t); basta hallar la transformada inversa. Resulta así.

$$X(t) = \sum_n x\left(\frac{n}{2W}\right) \operatorname{Sinc} 2W \left(t - \frac{n}{2W}\right)$$

Hemos probado así constructivamente el llamado teorema de las muestras que afirma que toda señal de energia finita y de banda limitada puede expresarse de modo único en función de sus muestras y valores instantaneos tomados a intervalos regulares Ts

El valor de Ts es tal que $1/Ts \geq 2W$, siendo W la máxima frecuencia.

Espectral de la señal. Ala frecuencia fs = 2W se le llama frecuencia de Nyquist y al intervalo de muestreo Ts, intervalos de Nyquist.

Lo que se hace es tomar un periodo T de ella. Si de esa sección de curva tomamos dos puntos la curva que totalmente definida. Dicho de otro modo sólo hay dos valores como máximo que son independientes y fijados estos, todos los demás quedan automáticamente determinados.

Si esta curva queda definida por dos puntos cualesquiera por periodo T, bastará con elegir y trasmitir esos dos y toda la información estará contenida en ellos el elegir o muestrear dos puntos por cada periodo equivale a muestrear con una frecuencia doble de f.

De hecho si la frecuencia máxima de la señal a trasmitir es fm Hz , la frecuencia de muestreo a de ser igual o mayor de dos fm Hz.

Los canales de frecuencia vocal (300-3.400) Hz ocupan una anchura de banda de 4kHz. Según esto, es necesaria que la división de la señal se realice con una frecuencia de 8.000 Hz, a fín de asegurar una reproducción fiel de todas las corrientes de voz.

LAS ETAPAS FUNDAMENTALES DE LA TECNICA MIC

La modulación por impulsos codificados, M.I.C., es una técnica de modulación relativamente reciente, que presenta diferencias fundamentales con los otros sistemas de modulación convencionales.

En los sistemas M.I.C., se emplea la técnica digital de modo que lo que varía con la señal moduladora es la combinación de presencias o ausencias de un grupo de impulsos que constituyen la representación de la señal en un momento determinado.

Este tipo de técnicas se ha venido utilizado en telegrafía , donde se emplean grupos de presencias o ausencias de impulsos para representar los distintos carácteres que constituyen un mensaje. No obstante el uso de esta técnica para señales analógicas, es decir señales como las vocales, es un problema bastante más complejo.

En el año 1937 el ingeniero A. H. Reeves que trabajaba en los laboratorios de la multinacional Norteamericana I. T. T. en París desarrolló un sistema basado en la modulación por impulsos codificados para la trasmisión de señales vocales, por su puesto patentó al año siguiente su invento.

Los primeros sistemas M. I. C., no tuvieron mucho exito debido a las dificultades de orden tecnológico. El progreso posterior de la miniturización y la creciente generalización del uso de componentes de estado sólido permitieron un abaratamiento y perfeccionamiento considerable en los circuitos, dando lugar a la aparición de toda una serie de equipos que utilizan la tecnica digital.

Las diferentes etapas del sistema M. I. C. son:

La señal a trasmitir, onda de baja frecuencia de amplitud variable de forma continua, debe someterse primeramente al proceso de muestreo, común por otra parte a todas las técnicas de modulación por impulsos. Este proceso ya ha sido tratado con anterioridad.

Las muestras obtenidas se someten a proceso de CUANTIFICACION.

El tren de impulsos modulados en amplitud (P. A. M.) que hemos obtenidos en la etapa anterior, tiene el inconveniente con vistas a su posterior tratamiento digital , de uqe las amplitudes de estos impulsos varían de forma analógica , es decir, pueden tomar cualquier valor intermedio. Es preciso disponer las cosas de forma que solo exista un número discreto de valores posibles, y para ello hay que "cuantizar" estas amplitudes.

Para ello, elegido el conjunto de valores discretos se le asigna ala amplitud de cada impulso el valor cuántico más cercano a su valor real.

Este proceso inevitablemente introduce una distorción que se conoce con el nombre de "Ruido de cuantización". Ahora se conprende que se está haciendo una apróximación y por tanto estamos introduciendo un error.

Este error en la práctica se mantiene entre unos estrechos límites para que la señal sea aceptable.

Las muestras asi cuantizadas , cuyas amplitudes no pueden pasar de un número limitado de niveles, previamente determinados como ya hemos dicho. Queda someterlo a la última operación que es la llamada "CODIFICACION". Mediante ella se sustituye cada muestra por un número de impulsos de amplitud física determinado.

La presencia o ausencia de cada uno de los impulsos de este grupo , permite obtener una serie de señales digitales que representan biunívocamente a los valores cuantizados de las muestras correspondientes.

El tren de impulsos así obtenido puede enviarse al medio de trasmisión. Al llegar los impulsos al primer repetidor, este solo necesita reconocer de cada impulso; Si está presente o si no lo está, no importando el grado de deformación. El repetidor puede entonces generar con gran fiabilidad el impulso idéntico al original. Se comprende por que la gran ventaja de este sistema frente a otros. Traslada el emisor al sitio donde se encuentra el repetidor.

Una vez llegado el impulso o el tren de impulsos al extremo receptor, se inician las etapas de reconstrucción de la señal original, que son similares a las ya comentadas , pero en sentido inverso.

Las operaciones básicas en la Técnica M. I. C. son las sitadas. Sin embargo existen unas "complementarias" que completan y perfeccionan los resultados obtenidos por las otras.

La primera de ellas y una de las más importantes es la que tiende a disminuir o compensar el ruido producido.

Sucede, que el ruido producido por la cuantización es función del número de cuantos o escalones elegidos, pero no le da amplitud de la señal. En consecuencia las amplitudes bajas se veran más afectadas que las altas.

Esto se traduce en que se precisa mayor definición, o sea,"escalones" más pequeños, para representar las amplitudes bajas que para las altas.

Es necesario realizar una "COMPRENSION", que partiendo de una ley no lineal, asigne cuantos o escalones más separados a las amplitudes altas, y más juntos a las bajas , mejorando de esa forma la relación señal/ruido del conjunto, si aumentar el número total de estados cuánticos.

Más adelante haremos un estudio más profundo de ese tema.

Por último se le puede añadir otra etapa, para la reducción de la enérgia de la señal. El tren de impulsos que se obtiene es "Unipolar" es decir, todos los impulsos tienen la misma polaridad. Esto representa de que el conjunto tiene gran porcentaje en enérgia en componente continua y de muy baja frecuencia, cuya trasmisión presenta serios problemas.

La solución de este problema se realiza mediante la trasmisión de la señal de forma "Bipolar" esto es, se intervienen alternadamente la polaridad de los impulsos de forma que dos consecutivos no tengan la misma polaridad.

Para trasmitir esta información y poder ser reconocida y en difinitiva para poder trabajar con ella, es preciso añadir una señal de señalización asociada a cada canal. Para trasmitir estas señalizaciones de los canales es preciso añadir los impulsos correspondiente a la señal múltiple. M. I. C. .

Pero, además, precizamos, una sincronización entre el equipo terminal de un extremo y el del otro, ya que cualquier desface entre la secuencia de muestreo , en un punto respecto al otro, hará imposible la reconstrucción de los diferentes canales en la recepción: Este sincronismo se consigue añadiendo nuevos impulsos a la señal múltiple.

En estas condiciones ya tenemos completa la señal M.I.C. y el sistema puede funcionar.

VISION MATEMATICA DEL MUESTREO

En este tipo de modulación se trasmiten pulsos cuya amplitud varía en proporción a los valores muestrales. Las muestras se localizan a intervalos de 1/2fm segundos., según se demostró anteriormente. El espectro de la señal muestreada fs(t) es decir de F(w). Se recupera f(t) al hacer pasar fs (t) por un filtro bajo con frecuencia de corte fm. El espectro de fs(t) es Fs(w).

$$F_s(w) = \frac{1}{T} \sum_{n=-\infty}^{\infty} F(w - nw_o)$$

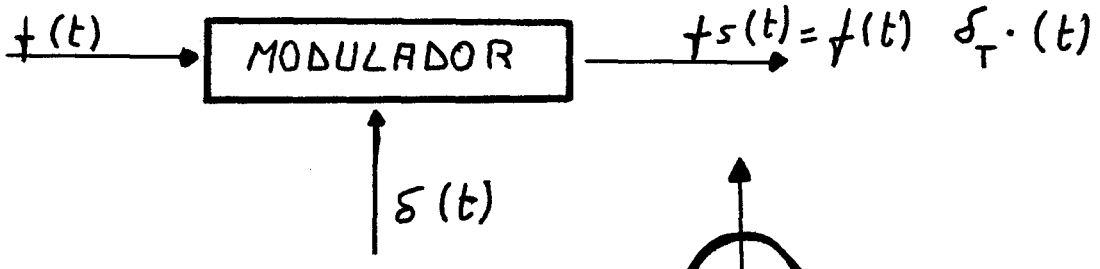
$$w_o = 2w_m$$

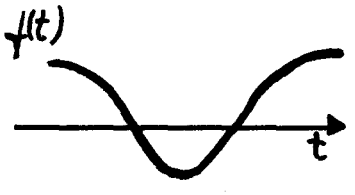
$$w_o = \frac{2\pi}{T}$$

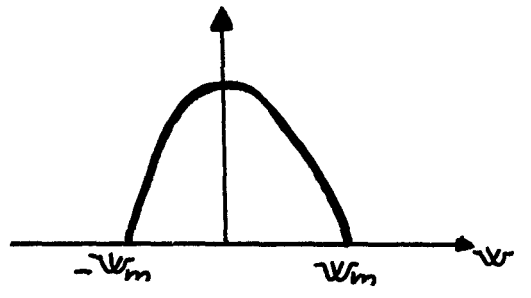
$$T \leq \frac{1}{2f_m}$$ Intervalo de Nyquist

$$F_s(w) = \frac{1}{T} \sum_{n=-\infty}^{\infty} F(w - 2nw_m)$$

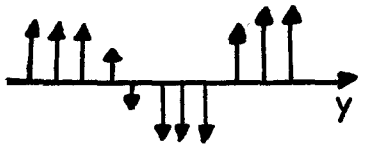
$$T = \frac{\pi}{w_m}$$

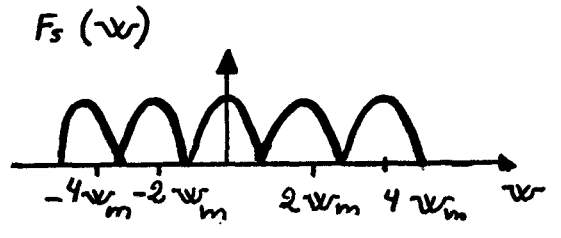




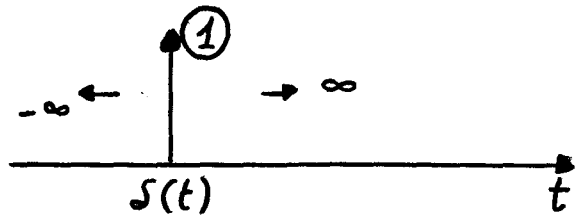


$$f_s(t) = f(t).\delta_T(t)$$

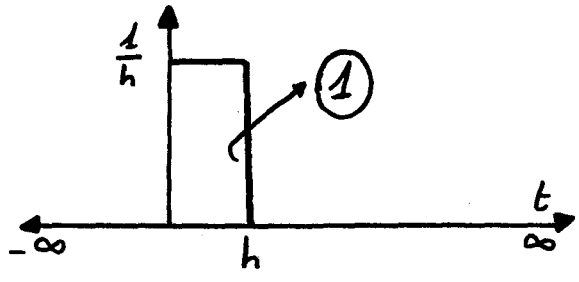




CONCEPTO MATEMATICO DE MUESTRA



$$\delta(t)\begin{cases} -\infty > \delta(t) = 0 \\ 0 = \delta(t) = 1 \\ \infty < \delta(t) = 0 \end{cases}$$



$$h(t)\begin{cases} -\infty < t > 0 \quad h(t) = 0 \\ 0 < + < h \quad h(t) = \frac{1}{h} \\ h < t < \infty \quad h(t) = 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(t)\,dt = \int_{-\infty}^{0} \delta(t)\,dt + \int_{0} \delta(t)\,dt + \int_{0}^{\infty} \delta(t)\,dt = 0+1+0 = 1$$

$$\int_{-\infty}^{\infty} h(t)\,dt = \int_{-\infty}^{0} h(t)\,dt + \int_{0}^{h} \frac{1}{h}\,dt + \int_{h}^{\infty} h(t)\,dt =$$

$$= 0 + \left[h - 0\right]_{0}^{h} \frac{1}{h} + 0 = \underline{\underline{1}}$$

CONVERTIDORES

Esta parte especifica detalladamente las partes electrónicas que van a conformar el circuito que hará posible la digitalización de la señal de voz y luego su reproducción

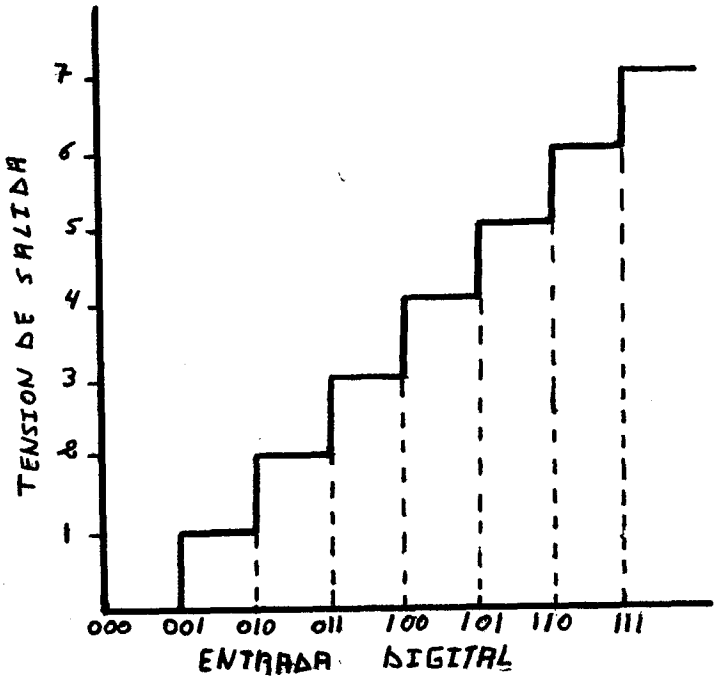
Una mayor información se puede conseguir en:
CIRCUITOS INTEGRADOS de Elias Muñoz Merino.

CONVERTIDOR D a A

La conversión de una señal analógica en digital requiere cuatro procesos: muestreo, retención, cuantificación y codificación. Los dos primeros procesos son los realizados por los amplificadores de muestreo-retención, mientras que los segundos serán conseguidos por los reconversores A/D (analógico-digitales). El paso contrario se tiene con los conversores D/A (digitales-analógicos), detras de los cuales será necesario un filtro en algunos casos que elimine los escalones de la señal de salida.

Convertidores Digital - Analógicos

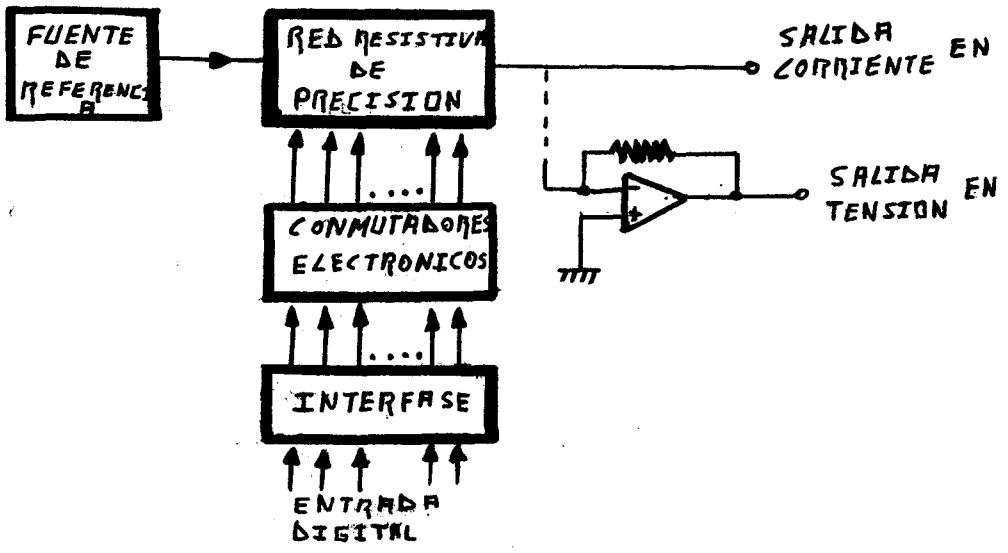
Los resultados entregados por el procesador digital deben ser transformados a fin de que puedan ser utilizados en el mundo analógico. Los convertidores D/A son, pues, circuitos que tienen que entregar una señal analógica proporcional al valor digital aplicado a su entrada tanto el tipo de información digital como el de la señal analógicas pueden ser cambiantes (Binario, BCD, complemento a 2, bipolar).



En el caso de la figura anterior se ha representado una

de las posibilidades: salida analógica en forma de tensión. Sin embargo, también se pueden encontrar disponibles comercialmente convertidores D/A (DAC) con salida en corriente.

Las estructura general de un DAC es la indicada en la figura siguiente. Un circuito de interface adopta los niveles lógicos de entrada a los que necesitan los conmutadores, los cuales, a su vez, actuan sobre una red resistiva de precisión que con ayuda de una fuente de referencia dará una salida analógica en forma de corriente o de tensión.



La diferencia fundamental entre unos conversores y otros reside en el tipo de red resistiva utilizada, y en la fuente de referencia: tensión o corriente. Desde el punto de vista de la red empleada dos son los tipos principales de DAC'S: los que constan de un sumador con resistencia de entrada ponderadas al peso binario de cada bit, y los y los que utilizan una red en escalera del tipo R-2R.

CODIGO BINARIO NATURAL

Este tipo es uno de los más utilizados como formato de entrada de un DAC, y cada palabra concreta representa una fracción del valor definido como Rongo o Fondo de escala, FSR, de la señal de salida del convertidor. En este código, cada bit tiene un cierto peso, denominándose al primero de la izquierda de una palabra binaria: MSB (bit más significativo) y a la derecha: LSB (bit menos significativo), siendo el valor analógico equivalente de este.

$$\text{LSB (valor analógico)} = \frac{FSR}{2N}$$

Donde N representa el número de bit del convertidor.

Según esto, dado un número binario es facil deducir el valor analógico de la tensión de salida equivalente, el cual nunca alcanzará el valor definitivo como FSR. En efecto, para el caso de un convertidor de 8 bits con un:

FSR = S voltios se tendrá:

$$LSB = \frac{S}{2^8} = 0'01953125 \text{ V.}$$

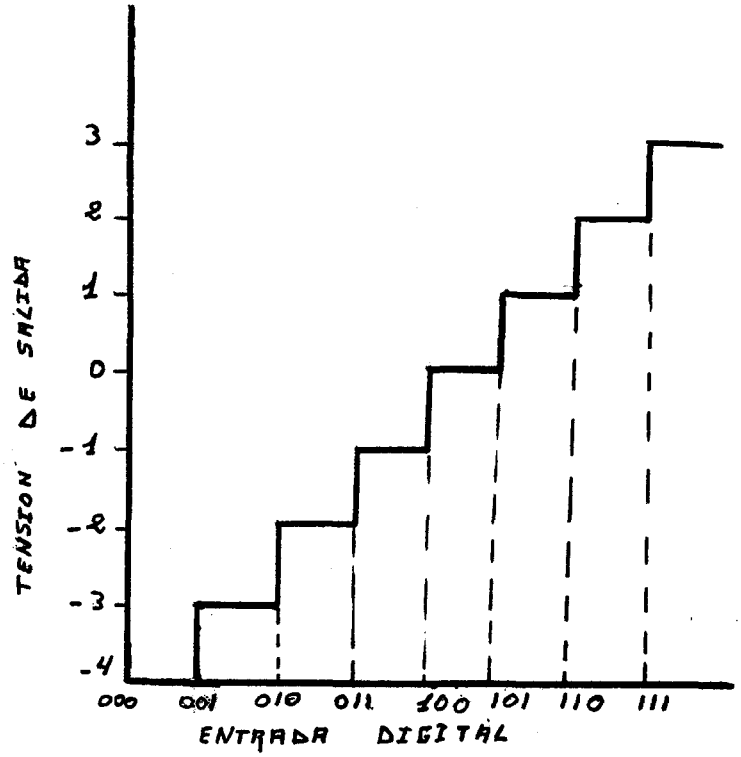
Y el máximo valor analógico de salida

(entrada 11111111) será:

$$FSR\ (1 - 2^{-8}) = 4'98046875 \text{ V.}$$

Si a un convertidor con código de entrada binario natural se le añadiera un " Offset " podría conseguirse que el valor binario central (100...00) le correspondiera una salida de OV, y al resto por encima y debajo de dicho valor tensiones positivas y negativas respectivamente (ver tabla) . Este código recibe el nombre de binario desplazado.

Entrada digital	Vo (V)
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

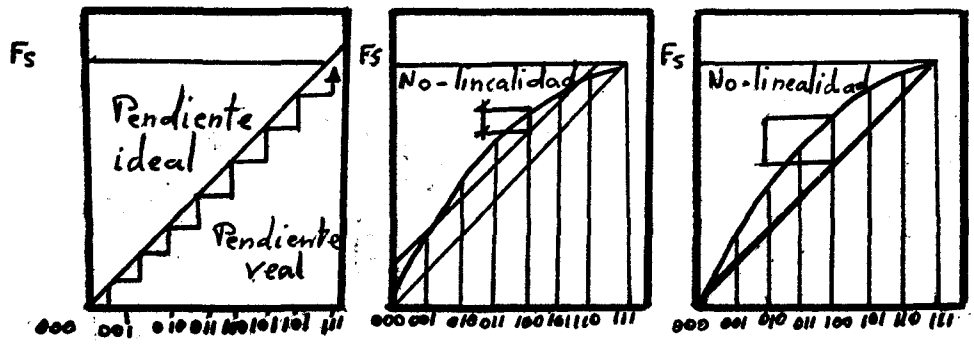


ESPECIFICACIONES DE LOS CONVERSORES D/A

En la caracterización de un producto existen ciertos datos que nos definen el mismo, mientras que otras especificaciones nos permiten evaluar la no identidad de un componente o circuito. En el caso de los conversores D/A los fabricantes facilitan ambas informaciones, La primera incluye: código digital de la señal de entrada, tipo de salida, niveles lógicos de señales de entrada, alimentación , ... y la segunda nos especifica el comportamiento no ideal de la función de tranferencia, así como su evolución con la temperatura y el tiempo. Esta última información es facilitada por los fabricante a través de una serie de especificaciones, de las que las más importantes son las que se resumen a continuación.

Resolución: Es el cambio incremental más pequeño de la tensión de salida, conforme a esto, un convertidor de N bits resolverá una parte en 2^n. Comercialmente existen DAC's de 8, 10, 12 ....bits cuya resolución depende del valor a fondo de escala (FS). Así, si es FS = 10V la resolución de un conversor de 12 bits es de 2,44mV. Análogamente para una salida de corriente (p. ej. 0,625 µA).

Error de escala. Tambien llamado error de ganancia, es la diferencia entre las pendientes de las funciones de trasferencia real e ideal (fig. 1). [illegible] el dato dado por los fabricantes suele ser la diferencia entre ambas rectas para la combinación que da el valor a fondo de escala sin que exista error de Offset, expresada como % del FS.



(a) (b) (c)

Fig 1

CONVERTIDORES DE DATOS

Linealidad (más bien no-linealidad). Es la máxima desviación de la salida real respecto a la teórica que sería una línea recta. Esta recta, dependiendo de fabricante será: o "la mejor" fig. 1,b) , determinada empíricamente para igualar las desviaciones máximas y mínimas, o la trazada entre los puntos extremos de la función de transferencia (fig. 1,c) . Se puede expresar en un % de valor a fondo de escala, o como una fracción de bit menos significativo. Un conversor deberá ser lineal en menos de 1/2 en LSB ya que en caso contrario si para un cierto código de entrada su valor fuera positivo y superior a 1/2 LSB, a la vez que negativo y de modo tambien mayor a 1/2 LSB para el código siguiente, el convertidor no cumplirá una especificación importante como es la de monotonicidad (la salida siempre crece o permanece constante al aumentar el valor de la entrada digital).

Tiempo de asentamiento. Es el tiempo que tarda en alcanzar la salida un nuevo valor estable., dentro de un cierto margen, tras un cambio en la entrada digital. La causa del mismo reside en las capacidades parásitas que dan a lugar a transitorios de una cierta duración, así como al comportamiento ideal de los conmutadores. Esta especificación puede ser dada con el tiempo requerido para esta

bilizar la salida al cambiar la entrada de un código al siguiente, o bien, lo que es más usual y de mayor interés el tiempo que emplea la salida en pasar desde OV hasta el valor próximo al fondo de escala indicado como (% del FS) Parámetros típicos para DAC con salida en corriente: 300 ns a 1 us para alcanzar valores comprendidos entre ± 0,005 y ± 0,2% del FS; para salidas de tensión: 5-30 µs para unos porcentajes del FS parecidos.

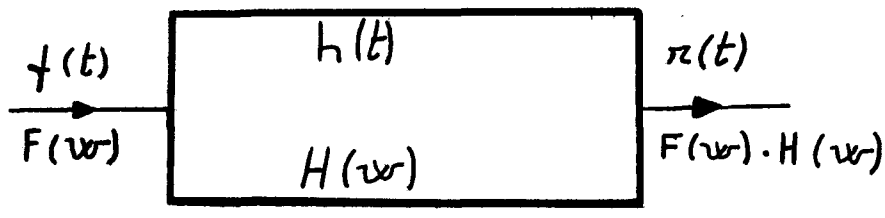
Sensibilidad con la temperatura. Las características mencionadas no son parámetros fijos, si no que, por el contrario cambian al hacerlo la temperatura ambiente.

De este modo, ésta influye sobre el "Offset", la ganancia o linealidad. El cambio de "offset" suele indicarse en µV/ºC o en partes por millón (ppm) por ºC(±15 ppm /ºC); el coeficiente de temperatura de la ganancia, en ppm/ºC (± 35ppm/ºC); y el del error de linealidad en ppm/ºC del fondo de escala (± 2 ppm/ºC del FS).

CARACTERISTICAS DE FILTRO DE LOS SISTEMAS LINEALES

En un sistema dado, una señal f(t) de entrada produce una señal de respuesta r(t) de manera caracteristica del sistema. La función de densidad espectoral de la señal de entrada es F(co) mientras que la función de densidad espectoral de la respuesta es F(w) H(co). Por lo tanto el sistema modifica la función de densidad espectoral de la señal de entrada.

Es evidente que el sistema actúa como una especie de filtro de las diferentes componentes de la frecuencia de intensidad de algunos componentes aumenta, la de otras se atenua y otras más pueden quedar iguales. De esta manera semejante, cada componente sufre un cambio de fase diferente en el proceso de trasmisión. Por lo tanto, el sistema modifica la función de densidad espectoral deacuerdo con sus características de filtro. Esta modificación depende de la función de transferencia M(w), que representa la respuesta del sistema a las diferentes componentes de frecuencia. Así, H(w) actua como una función de ponderación según las diferentes frecuencias. La respuesta resultante tiene densidad espectoral F(w),H(w). La señal de entrada tiene densidad espectoral F(w) y la función de trasferencia está dada por H(co). La densidad espectoral de la respuesta evidentemente es F(w) H(w).



Considérece el circuito R-C de la figura (a) siguiente.

En las terminales de entrada de este circuito,se a - plica un pulso rectangular comoel de la figúra.

La respuesta del circuito es el voltaje Vo(f) que a- parece en las terminales de salida. La función de densi- dad espectoral de la señal de entrada (pulso rectángular) está también representada en la figura (d). La función la trnsferecia H(w) del circuito, que relaciona el volta- je de salida con el de entrada, es obviamente.

$$\frac{1}{(jw + 1)}$$

Por lo tanto:

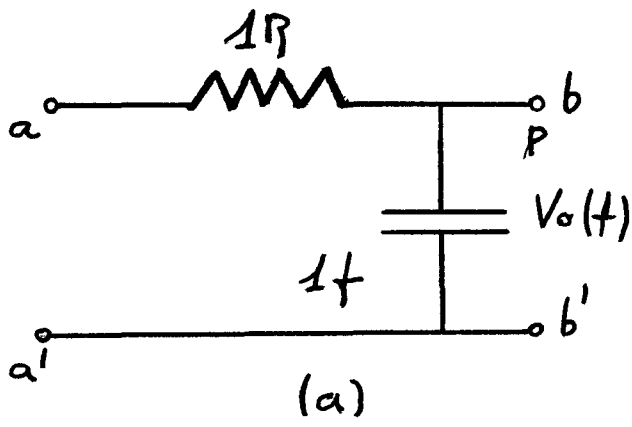
$$H(w) = \frac{1}{(jw + 1)}$$

En la figura (b) se muestra la gráfica de la magnitud /H (w) / correspondiente a las características de filtro de circuito. Por el momento, haremos caso omiso de las ca - racterísticas de fase. Obsérvese que este circuito atenú a las frecuencias altas y permite que las bajas pasen con atenuación relativamente pequeña. Así, este circuito esla forma más simple de un filtro de paso bajo los componente de frecuencia alta del espectro de entrada experimentan u na atenuación severa si la comparamos con la frecuencia baja.

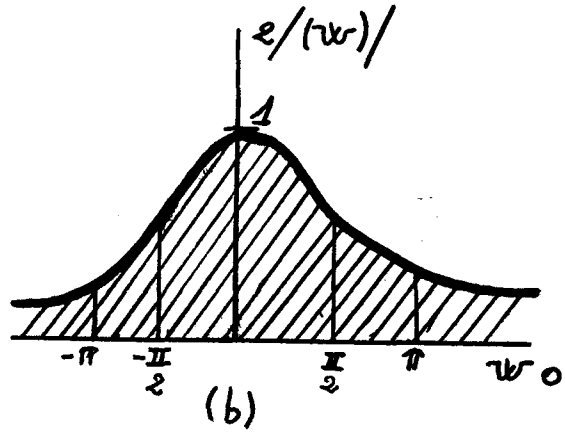
La función de densidad espectoral de la respuesta es el producto F(w), H(w). En la figura(s) se ve la magnitud /F(w), H(w) / de esta función.

La comparación entre las figuras (d) y (f) muestra cla ramente la atenuación constante por el circuito a las fre cuencias altas. La función de respuesta Vo(f) figura (e) es obviamente una réplica distorsionada de la señal de en trada. acceso a la trasmisión de todas las componentes de

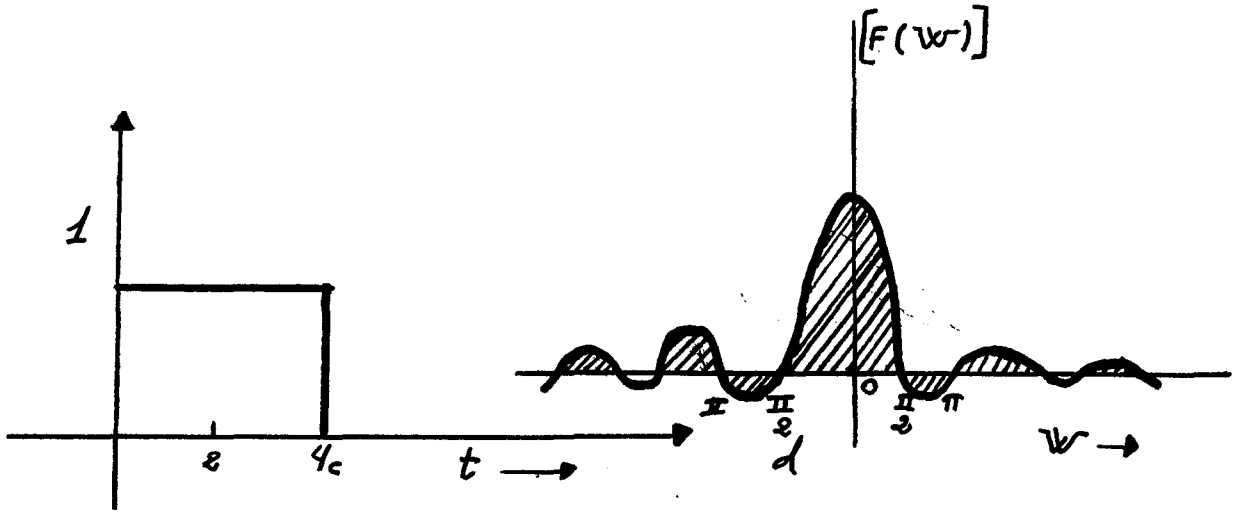
frecuencia de la señal de entrada. En particular, las componentes de frecuencia alta resulta muy afectadas.

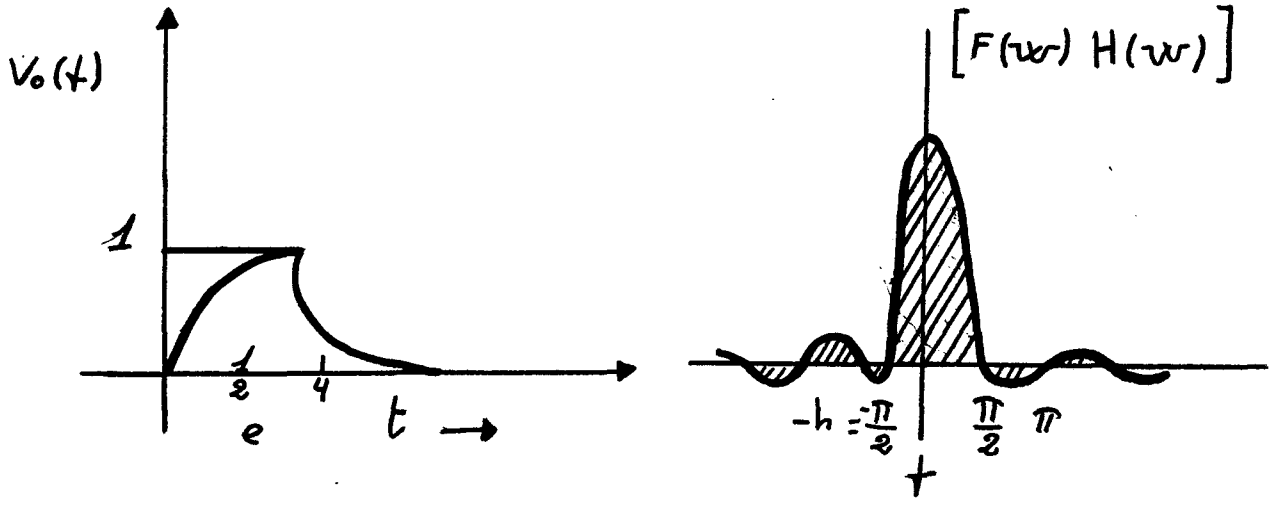


(a)



(b)

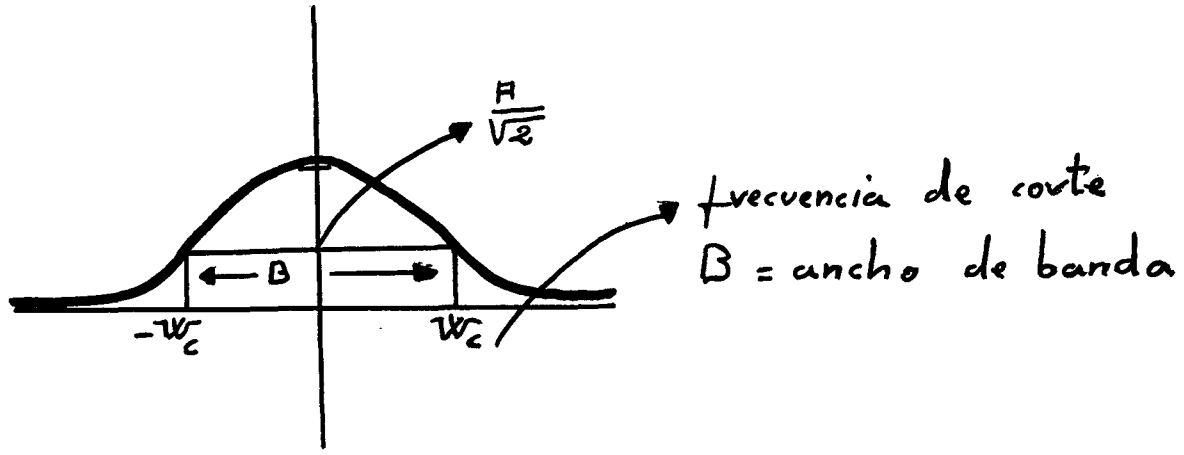




Se manifiesta en las características de elevación y caida del voltaje de respuesta. La señal de entrada sube abruptamente en f=Ø. La elevación súbita, que significa un cambio rápido, implica componentes de frecuencia muy alta. Y

Ya que el circuito no permite el paso a las componente de frecuencia altas, el voltaje de salida no puede cambiar con esa rapidez, por lo que sube y baja menos abruptamente en comparación con la señal de entrada.

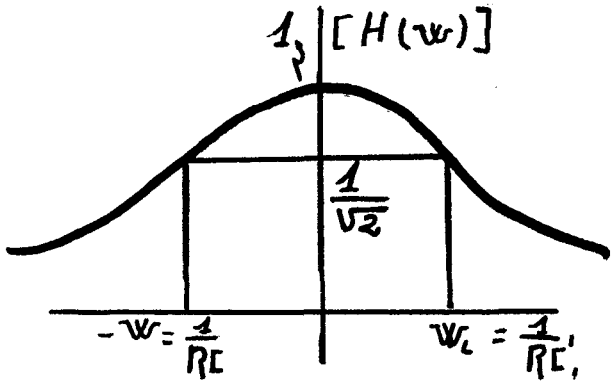
Para filtrar una señal se necesita un criterio. Este nos da perfectamente el ancho de banda de la función de transferencia.



$$\left[H(w) \right] > \frac{1}{\sqrt{2}} \left[H(w) \right]$$

Valor en $w = \emptyset$ Centro del intervalo o Valor máximo según sea el filtro.

Normalmente se presenta por 20log. /H(w)/.
Se habla entonces de decibelios y es así por razones de escala.



$B = 2w_c = \frac{2}{RC}$ ancho de banda

$20 \log |H(w)| = |H(w)|$. dB
↳ expresado en dcc.

Como $w_c = \pm \frac{1}{RC}$

$\frac{1}{w_c} = \pm RC$

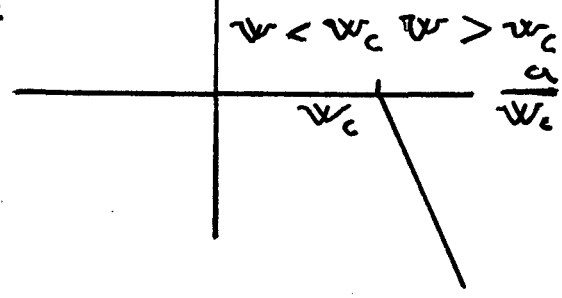
$$20 \log |H(w)| = 20 \log \frac{1}{\sqrt{\left(\frac{w}{w_c}\right)^2 + 1}} =$$

$$= 20 \log 1 - 20 \log \sqrt{\left(\frac{w}{w_c}\right)^2 + 1}$$

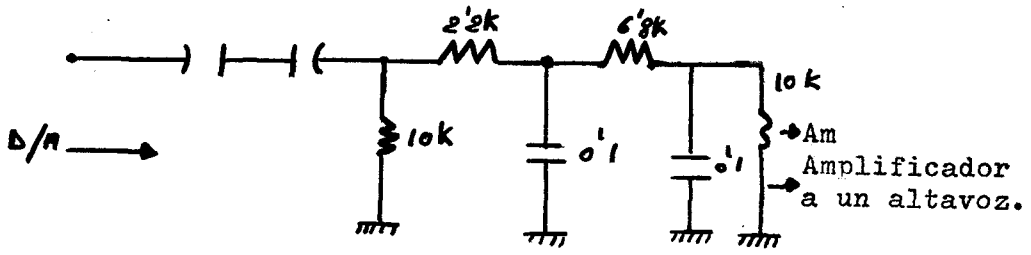
Aproximadamente para frecuencias mayores de las de corte

$w > w_c \Rightarrow -20 \log \frac{w}{w_c}$ atenuante en decibelios

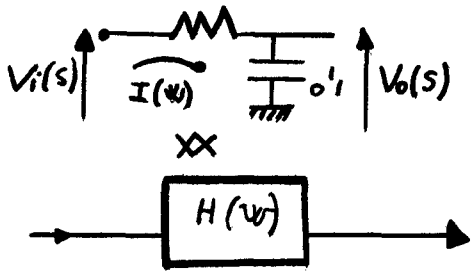
Para $w < w_c = \phi$



A la salida del decodificador D/A colocaremos un filtro paso bajo.



Calculo de la función de transferencia.

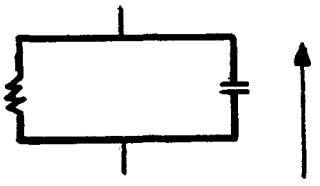


La tensión de entrada Vi (S) y de salida Vo (S) están en función de S = jw.

H(W) = relación entre entrada y salida.

Se puede demostrar que $H(W) = \frac{Vo(S)}{Vi(S)}$ por tanto: Vo(S) = $= Z_e \cdot I(w)$.

Para calcular Z (Impedancia del sistema lineal) cortocircuito de entrada Vi(S) entonces:



Vo (s)

$$Zeq = \frac{R\,\frac{1}{cs}}{\frac{1}{R\,\frac{1}{cs}}}$$

Por lo que :

$$Vo(S) = \frac{R \cdot I(w)\frac{1}{cs}}{R + \frac{1}{cs}} = \frac{Vi(S) \cdot \frac{1}{cs}}{R + \frac{1}{cs}}$$

$$H(W) = \frac{Vo(S)}{Vi(S)} = \frac{\frac{1}{cs}}{R \cdot \frac{1}{cs}} = \frac{1}{RCS + 1}$$

$S \longrightarrow \infty \Rightarrow H(w) = \emptyset$

$S \longrightarrow \infty \Rightarrow H(w) = 1$

Demostrar:

Para W = ∅ tenemos $\left|H(W)\right| \pm 1$

$$|H(w)| = \frac{1}{\sqrt{(RCjw)^2 + 1}} = \frac{1}{1} = 1$$

Para w = ∞ tenemos $|H(w)| = \emptyset$

$$|H(w)| = \frac{1}{\infty} = \emptyset$$

Cálculo práctico de la frecuencia de corte:

$$\frac{1}{\sqrt{(RCjw)^2 + 1}} = \frac{1}{\sqrt{2}}$$

$$(RCjw_c)^2 + 1 = 2$$

$$(RCjw_c)^2 = 1$$

$$-R^2C^2w_c^2 = 1$$

$$-w_c^2 = \frac{1}{R^2C^2}$$

$$-w_c = \pm\sqrt{\frac{1}{R^2C^2}} \Longrightarrow w_c = \mp\frac{1}{RC}$$

Para R_2 = 6'8K y C_2 = 0'1 μF

$$w_c 2 = \pm\frac{1}{(6'8 10^3 . 0'1 . 10^{-6})} = \pm 1.470'58 \text{ Rad/seg}$$

Para R_1 = 2'2K y C_1 = 0'1 μF

$$w_{c1} = \pm \frac{1}{(2'2 10^3 \cdot 0'1 10^{-6})} = \pm 4'545'45 \ Rad/Seg$$

Como $w = 2\pi f$

$$f_2 = \frac{w_{c2}}{2\pi} = \frac{1470'58}{2.\pi} = 233'95 \ Hz$$

$$f_1 = \frac{w_{c1}}{2\pi} = \frac{4.545'45}{2\pi} = 723'43 \ Hz$$

LOS CONVERTIDORES ANALOGICOS DIGITALES

Estos conversores denominados tambien ADC's en la nomenclatura anglosajona, tiene como misión la obtención de una representación digital de la magnitud analógica que se presenta a su entrada. Los procesos que deben llevar a cabo son los de cuantificación, por el que la señal analógica continua de entrada se transforma en un conjunto discreto de estados de salida, y codificación, el cual asigna un conjunto de bits (palabra o código digital) a cada una de dichos estados. La función de transferencia de un conversor A/D es la indicada en la (Fig. 2,a) para el caso de cuantificar en ocho estados de salida una señal analógica comprendida entre 0 y +10V. La escalera es la mejor aproximación que se puede obtener de una función de transferencia rectílinea ideal, por eso se introduce un error de cuantificación (fig. 2,b).

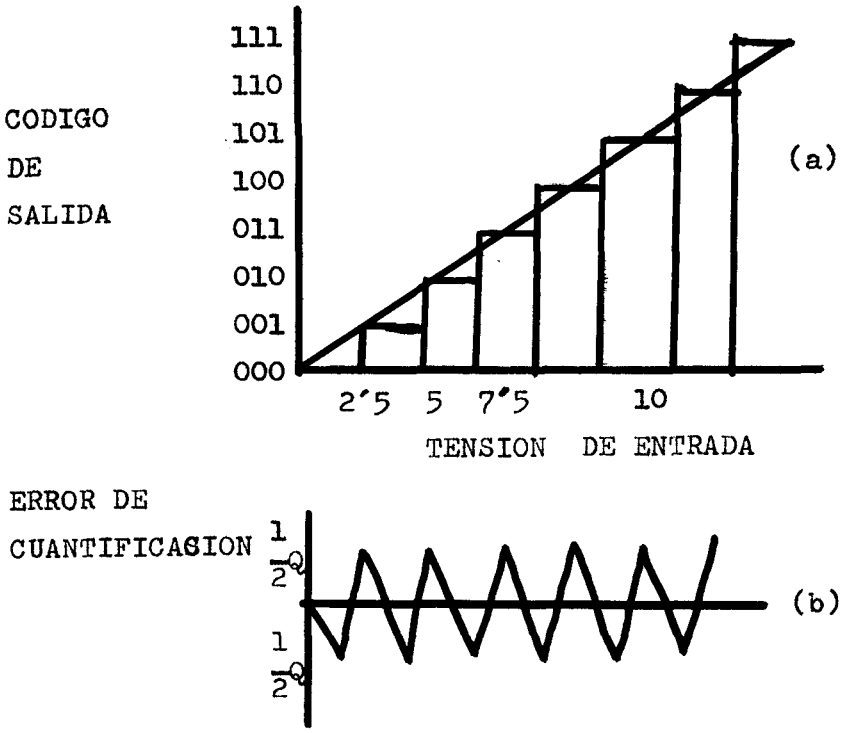
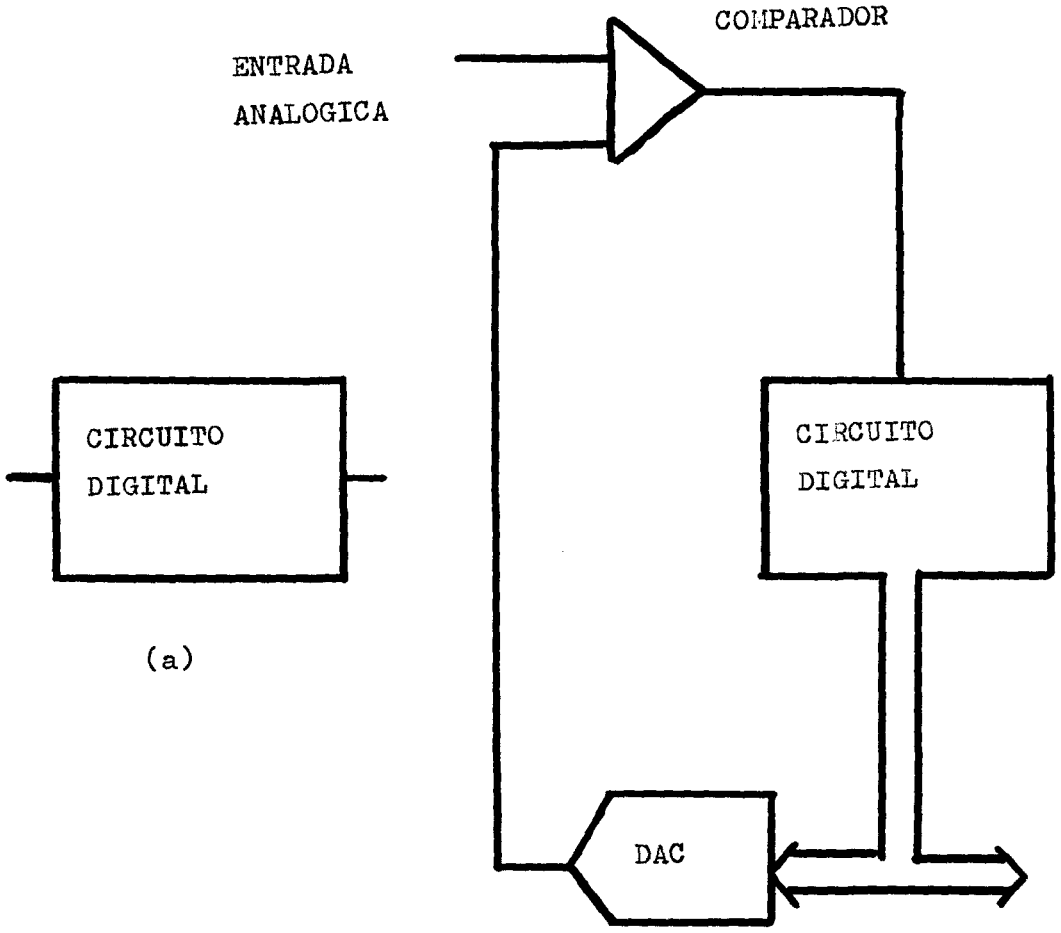


Fig 2.

Existen muy diversas técnicas para realizar esta transformación analógica-digital. Estas pueden ser agrupadas en dos tipos básicos: ADC's de bucle abierto y bucle cerrado. En los primeros no existe realimentación interna, obteniendose la información digital de forma directa (fig. 3,A). Entre ellos están los que convierten la señal

en frecuencia o en impulsos y los que emplean comparadores. Un segundo tipo de conversores D/A son los que poseen un lazo de realimentación del que forma parte un conversor D/A En ellos los procesos de cuantificación y codificación son simultáneos, obteniendose una seduencia de números digitales que son reconvertidos a un valor analógico, el cual es comparado con la entrada. La salida digital es el valor mas próximo. Veamos a continuación la estructura de algunos conversores de ambos tipos.



(a)

(b)

Fig 3

CONVERTIDOR A/D CON APROXIMACION SUCESIVAS

Este es el conversor más popular para la mayoria de las aplicaciones donde requiere medias y hasta altas velocidades de conversión. Con una estructura en bucle cerrado como en los casos anteriores, pero con un circuito de control algo más complejo, el ACD trata de acercarse al valor final mediante aproximaciones sucesivas. La operación es análoga a la que realiza para conocer el peso de un objeto utilizando unos pesos estandar de valor 1.½, ¼....1/N Kg. El proceso lógico es comenzar poniendo en la balanza el peso mayor, quitándolo y/o y añadiendo más de valores menores en sentido descendente según sea el resultado de la comparación con el peso desconocido. Una vez empleado el peso más pequeño(1/N Kg.) la operación habrá finalizado.

En nuestro caso utilizamos un registro de desplazamiento que almacenará la convinación digital de un circuito secuencial de control (fig.4,a), cuyo conjunto suele recibir el nombre de aproximaciones sucesivas. El registro es inicializado con la combinación 100....00, lo cual supone explorar si el valor analógico de entrada está por encima o debajo del valor medio del rango. Si por ejemplo V_A es mayor que dicho valor, se investigará dentro de la mitad superior añadiendo un 1 en la siguiente posición y se pasa a la siguiente (10100..00). Si aquí se obtiene que V_A es menor que el aquivalente analógico de dicho valor digital se quita el 1 de la segunda posición y se pasa a la siguiente (10100..00)/. Este proceso finalizará cuando se haya estudiado el comportamiento del bit menos significativo, lo cual es indicado a través de la linea de fin de conversión.En el caso de que un conversor de tres bits, el diagrama de transiciones de la fig. 4,b) del circuito secuencial de control representa todas las posibilidades. Como se puede comprobar solo se necesitan tres impulsos de reloj para alcanzar el valor final(en general un número de impulsos igual al de bits de resolución) siendo, además, el tiempo de conversión con independencia del valor analógico de entrada. Es por estas

simplicidad y rapidez, por la que este conversor es de los más utilizados en la mayoria de las aplicaciones, y las velocidades que puedan alcanzar serán mayores cuanto más se reduzcan los tiempos de propagación y respuesta de los circuitos internos D/A comparador.

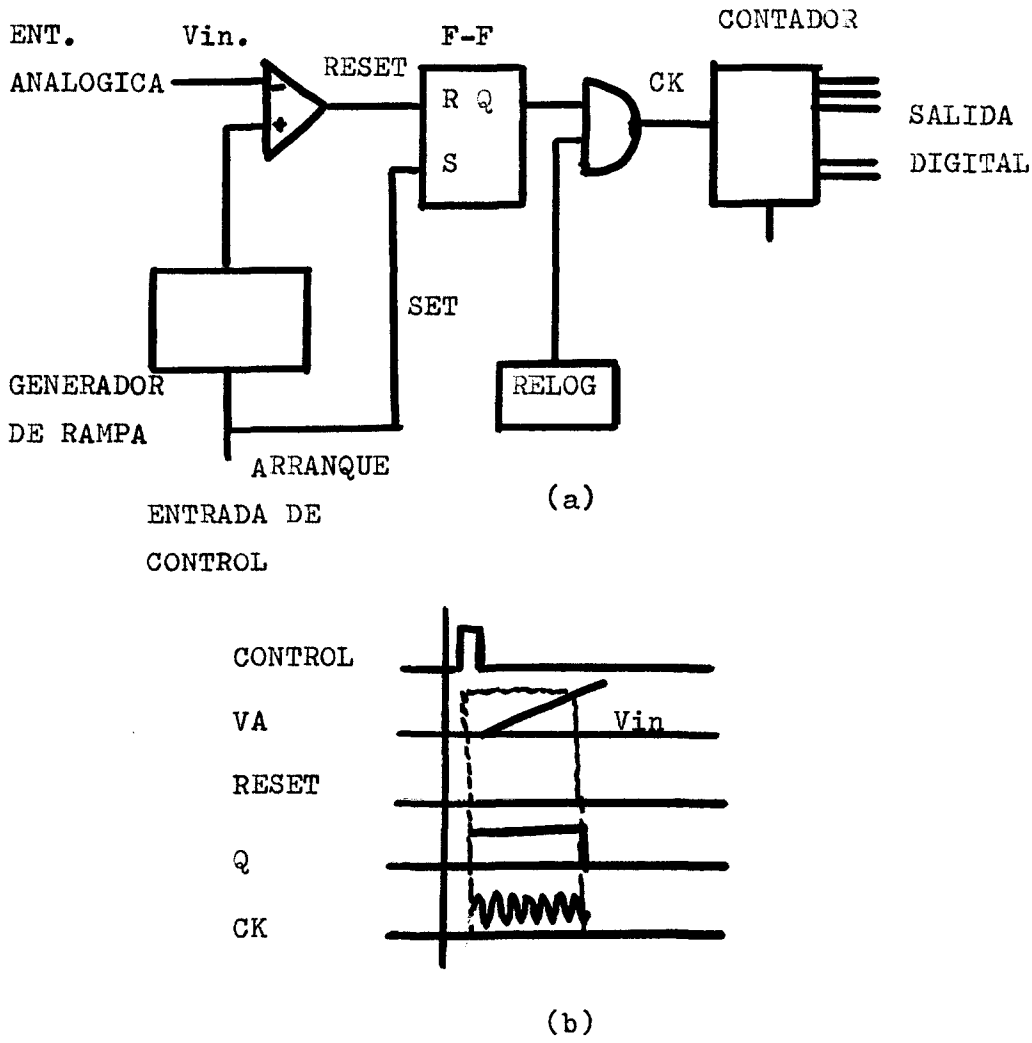


FIG 4

ESPECIFICACIONES DE LOS CONVERSORES A/D

Muchas de las especificaciones que suministran los fabricantes para los convertidores A/D ya han sido citadas al referirse a los DAC, con la única diferencia de estar referidas ahora a la entrada analógica y una salida digital Estas son : resolución, Linealidad, Presición, Impedancia, y sensibilidad de dichos parámetros con la temperatura. Sin embargo existen otras especificaciones propias de estos conversores tales como:error de cuantificación y tiempo de conversión.

En el primer conjunto de parámetros, al estar intercanbiadas las funciones de los terminales de entrada y salida respecto a los DAC., pueden tenerse valores muy distintos Así , por ejemplo, la impedancia será de entrada y comprendida entre pocos kiloohmios y decenas de megaohmios en los ADC, mientras que en los ADC era de salida muy baja.

Los ADC comerciales pueden admitir entradas analógicas unipolores o bipolares con rangos típicos de ±10V,0-10V con códigos de salida: binario natural, binario desplazado, BCD, complemento a 1 o complemento a 2. Muchas veces los circuitos permiten programar el rango de entrada y el código de salida entre varias opciones.

Error de cuantificación este tipo de error ya fué mencionado al introducir los convertidores A/D y aparece como consecuencia de que un ADC de N bits la continuidad de la señal analógica es dividida en 2^N rangos. De esta forma todos los valores analógicos dentro de un rango estan representados por un único código digital, normalmente asignado al valor medio del mismo. Existe, pues, siempre un error de cuantificación de ± 1/2 LSB, que junto con el ruido del sistema digital, "offset"... engloba dentro de la especificación de presición.

Tiempo de conversión. Es el tiempo requerido por el conversor para entregar la palabra digital equivalente a la entrada analógica. Como ya se vió, este tiempo puede variar

mucho de unos convertidores a otros, siendo un valor típico el de 50μs para velocidades medias, pudiendo llegar a varios milisegundos los de baja velocidad o reducirse a 50ns o menos en los ultra-rápidos.

ANALISIS DE LA SEÑAL DE VOZ.

Este tema ha sido desarrollado en su totalidad por los apuntes que sobre voz ha editado en esta escuela el profesor Santos, correspondientes a unos estudios que sobre la misma ha desarrollado en Madrid.

Para una mayor información en el particular remitirse a dichos apuntes.

ANALISIS DE LA SEÑAL DE VOZ

Con el procesado digital de la señal de voz se pretende obtener una representación más conveniente o más util de la información transportada por la señal. Representando la señal de voz de otra forma se consiguen facilidades tales como determinar si la voz es sonora o sorda y discennir entre voz y silencio (ruido).

La supusición básica realizada al analizar la señal de voz consiste en considerarla como una señal cuasiestacionaria en segmentos cortos de duración entre 10 y 30 milisegundos. En estos segmentos las propiedades de la señal se consideran fijas. Esto permite aplicar métodos de procesado localizado. Este procesado proporciona una secuencia temporal de números que representan uno o más parámetros de la señal de voz. Estas representaciones paramétricas ponen de manifiesto de la voz que en su estado de forma de onda no es posible, en general, evidenciar.

En el apartado de introducción se señaló que la primera tarea de cualquier sistema de reconocimiento de voz es extraer los parámetros de esta para así formar de test y de referencia de las palabras a ser reconocidas. En este apartado se presentan métodos para obtener estos parámetros tanto en el dominio del tiempo como de la frecuencia.

ANALISIS EN EL DOMINIO DEL TIEMPO

Consta de un conjunto de técnicas de procesado denominadas métodos temporales. Esto significa que los métodos de procesado comtemplan directamente la onda de la señal de la voz.

Las representaciones de la señal de voz que se representan en esta sección son la velocidad de cruces por cero, la energia y la función de autocorrelación. La estimación de estas medidas requiere un procesado digital mùy sencillo Sin embargo los resultados obtenidos proporcionan una base util para estimar características de la voz tales como el p

periodo de tono (frecuencia fundamental), comienzos y finales de palabras, discriminación voz-silencio y distinción sonoro-sordo.

ENERGIA DE LA SEÑAL DE VOZ. DISTINCION DE SEGMENTOS DE VOZ SONORA Y DE SEGMENTOS DE VOZ SORDA.

La energia localizada de la señal se define como:

$$E_n = \sum_{m=0}^{N-1} s^2(m) W(n-m) \qquad 3.1$$

donde W(p) es una ventana del tipo de la sección 1.3.. La longitud adecuada de esta ventana para una velocidad de muestreo de 8 Khz está comprendida entre 100 y 300 muestras (duración de los segmentos de voz entre 12.5 y 37.5 milisegundos).

En la figura 1.13 se observa que la amplitud de la señal de voz varia apreciablemente con el tiempo. Los segmentos sordos presentan una amplitud mucho menor que los sonoros. Según la ecuación 3.1 esto significa que los sonoros presentan mayor energia que los sordos. Basandose en este principio se puede construir un algoritmo para calcular la energia de la voz en segmento sucesivos de este tipo se ha elaborado el programa VOZ-ENE que permite representar graficamente la energia de la señal de voz. [La figura 1.38 muestra la energia en función del tiempo (milisegundos) de los segmentos de la palabra LAS]

Se observa que los segmentos con menor energía se corresponden con voz sorda. Esto permite realizar de forma sencilla una primera aproximación estimativa de los segmentos de voz en cuanto a la sonoridad.

VELOCIDAD DE CRUCES POR CERO. MEDIDA DEL CONTENIDO DE LA SEÑAL DE VOZ

Para señales discretas se dice que ocurre un cruce por cero si muestras sucesivas tienen distintos signos algebraicos. Por tanto, la velocidad de cruces por cero da una medida sencilla del contenido de frecuencia de la señal de voz. Una definición de la velocidad de cruces por cero puede ser

3.2

$$VCC = \sum_{m=0}^{N+1} / \text{sig.}\{s(m)\} - \text{sig.}\{s(n)\} / W(n-m)$$

donde:

$$\text{sig.}\{s(p)\} = \begin{cases} 1 & \text{si } s(p) \geq 0 \\ -1 & \text{si } s(p) < 0 \end{cases}$$

y la ventana W :

$$W(p) = \begin{cases} 1/2N & 0 \leq p \leq N-1 \\ 0 & \text{en otro caso} \end{cases}$$

La longitud promedio N de la ventana está comprendida entre 100 y 300 muestras para una velocidad de muestreo de 8 Khz El modelo canónico de producción de voz presentado, es la sección 2. 1. , sugiere que la energia de la voz sonora está concentrada por debajo de los 3 Khz a causa de que el espectro introducido por la onda glotal decae y que la energia de la voz sorda se encuentra a frecuencias más altas. Ya que a frecuencias altas implican alta velocidad de cruces por ceros y bajas lo contrario, estrecha relación entre la velocidad de cruces por ceros y la distribución de la energia con la frecuencia. Por tanto una razanable generalización es que si la velocidad de cruces por ceros es alta la señal de voz es sorda y si es baja sonora. No obstante, esto queda algo impreciso al no indicarse que se entiende por baja y por alta.

El algoritmo descrito por la ecuación 3.2. se a incrementado mediante el programa VOZ -CCERO. Este permite representar graficamente la velocidad de cruces por cero de segmentos de voz. La figura 1.39 muestra de VVC para los segmentos de la palabra LAS.

Se observa que segmentos de la consonante.

VCC mayores que segmentos de la vocal A.

S presentan valores de la consonante L.

DESCRIMINACION VOZ vs SILENCIO USANDO LOS PARAMETROS DE ENERGIA Y CRUCES POR CERO

Trata problemas de la localización de los principios y finales de las palabras. Esto es esencial en el reconocimiento de las palabras aisladas. Este problema no es trivial salvo para los casos en que el parámetro SNR (relación señal-ruido) sea alto. En general el problema presenta dificultad si al principio o al final de palabra hay:

1. Picos fricativos (/f/, /S/, /x/,)
2. Picos de ruptura oclusivos (/p/, /t / , /k/).
3. Nasales en los finales de palabras.
4. Fricativas sonoras que se convierten en sordas al final.
5. Desvanecimientos del sonido vocalicos al final de palabras. En otros términos, las dificultades se presentan cuando la energia de la voz al final o al principio de la palabra es comparable a la energia del ruido.

A pesar de estas dificultades, los parámetros de energia y cruces por cero se pueden utilizar de forma conjunta para obtener un algoritmo que permita estimar principios y finales de palabras. La descripción cualitativa de un algoritmo de este tipo es: Los patrones de las palabras a reconocer se forman utilizando un locutor que pronuncie estas en intervalos de grabación preestablecidos. Al principio y al final de cada palabra grabada se situa un intervalo de tiempo, por ejemplo del orden de 100 milisegundos, que no contiene voz.

En estos intervalos , se calcula la media y la desviación típica de los dos parámetros para obtener una caracterización del ruido permite establecer unos umbrales que son utilizado para estimar los principios y finales de los patrones de tes de las palabras a reconocer.

FUNCION DE AUTOCORRELACION. ESTIMACION DEL TONO PERIODO DEL TONO O FRECUENCIA FUNDAMENTAL DE LA SEÑAL DE VOZ.

La estima directa del periodo de tono sobre la señal de voz es practicamente imposible debido a la complejidad de esta señal. Esto es así a causa de la dificultad de establecer los requisitos necesarios para estimar un periodo: Ante esta dificultad se recurre a transformaciones de la señal de voz. La función de autocorrelación es una transformación que proporciona facilmente la estima del periodo de una señal.

La función de autocorrelación localizada para una ventana de longitud N se define como:

$$\hat{R}_n(k) = \sum_{m=0}^{N-1-K} [s(n+m)\,W(m)]\,[s(n+m+k)\,W(m+k)]$$

donde k es el orden de autocorrelación y w la ventana de análisis utilizada. De las propiedades de esta función se deduce que la auto correlación de una señal periódica es tambien periodica y con el mismo periodo.

Además la autocorrelación de estas señales periodicas alcanza sus máximos en la muestras 0+- p, +- 2p, ... Esto indica, independientemente del origen del tiempo de la señal, que el periodo se puede estimar fijando el origen de la media en R(0) y encontrando el primer máximo de la autocorrelación.

La elección de la langitud N de la ventana de análisis es

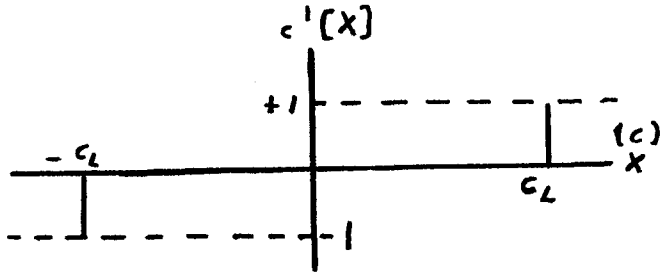
importante para tener una buena indicación de periocidad de la voz. Al igual que en la estima de casi todos los parámetros de esta, aquí tambien se enfrentan requerimientos con conflictivos. Debido al cambio de propiedades de la voz, el valor de N debería ser tan pequeño como fuera posible. Por autocorrelación el valor de N debería ser suficientemente grande. Esto es así a causa de la longitud finita del segundo segmento enventado utilizado en el cálculo y que a medida que aumenta el orden de la autocorrelación hay menos datos comprendidos en dicho cálculo. Hay que señalar que tambien que ventanas muy largas posibilitan promediados indeseados cuando el periodo del tono es corto. Para satisfacer ambos requisitos se puede tomar un conservador para N de almenos dos periodos del tono que se espera estimar. En un caso general se recurre a métodos adaptivos.

El número de terminos de la autocorrelación, es otro parámetro que concretar. Para una frecuencia de muestreo de 8Khz varia entre 15 y 30.

[El programa de VOZ-AUTOCOR permite representar la autocorrelación de segmentos de una señal y estimar el periodo del tono. Las figuras 1.41 y 1.42 muestran la autocorrelación de los segmentos de la palabra LAS y los periodos del tono estimados.]

Como punto final hay que indicar que la autocorrelación de la señal de voz presenta muchos picos que pueden ser atribuidos a las oscilaciones amortiguadas de la respuesta del tracto vocal y que son responsables de la configuración de cada periódo de la onda de voz. Estos picos se acentúan al usar ventanas demasiado cortas con respecto al periodo del tono a estimar y por cambios rápidos de las frecuencias formantes. Por tanto, en los casos en que estos picos sean, del orden de los picos debidos a la peridicidad de la exitación vocal el procedimiento de tomar el pico mayor de la función de autocorrelación para estimar el periodo del tono no es válido.

Para resolver el problema este la señal de voz es sometida a un procesado previo. En el dominio del tiempo las transformaciones más usadas son los correctores centrales. La figura 1.43 muestra un recortador de tres niveles de este tipo.



Fgura 1.43. Recortador central de tres niveles.

Según esta figura los valores de la señal que no superen un umbral fijado se les iguala a cero, y a los que si -1. Al resultado de esta transformación se le indica la autocorrelación.

[Este problema de procesado previo se aborda con más eficacia en el dominio de la frecuencia. La sección 3.2.1. presenta un procedimiento mucho más potente basado en el error residual de predicción lineal.]

3.2. ANALISIS EN EL DOMINIO DE LA FRECUENCIA

Las técnicas del PDS utilizadas para anlizar la señal de voz en el dominio de la frecuencia contienen de forma implicita o explícita alguna forma de representación espectoral.

En esta sección se utiliza el análisis de predicción lineal para estimar el periodo del tono, las frecuencias formantes y los anchos de bandas de estas.

3.2.1. DETECCION DEL PERIODO DEL TONO USANDO LOS PARAMETROS LPC DISTINCION SONORO-SORDO.

La utilización en el dominio de la frecuencia de la función de autocorrelación para estimar el periodo del tono no es sencillo debido a la presencia conjunta de los espectros de la respuesta del tracto vocal (formantes) y el de la exitación de los pulsos glotales, cuya periodicidad se trata de estimar. Esta dificultad es abordada sometiendo la señal de voz a diferentes transformaciones.

Markel propuso el algoritmo denominado SIFT (Simplified Inverse Filter Tracking) para estimar el periodo del tono y realizar la desición sonoro-sordo. La figura 1.44 muestra el diagrama de bloques de este algoritmo.

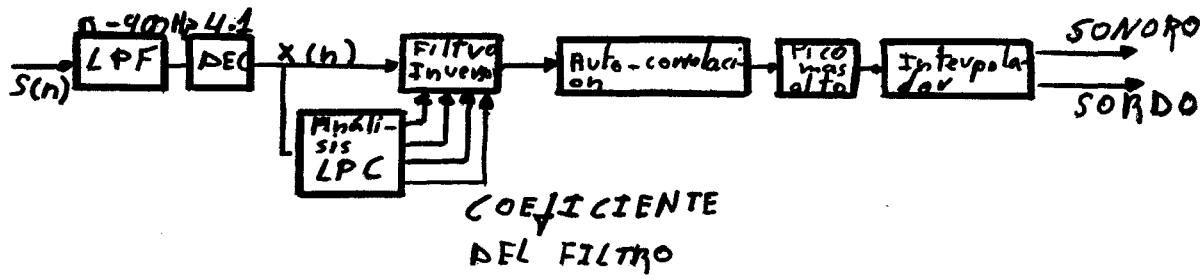


Figura 1.44 Algoritmo SIFT.

La descripción cualitativa de esta figura es como sigue: El segmento de voz se filtra paso bajo (fc=800 Hz.) para limitarla en banda. A continuación se decima (4:1), es decir, se toma 1 de cada 4 muestras. Esto tiene como finalidad reducir la velocidad de muestreo de 8 Khz. a 2Khz. para aumentar la eficiencia del procesado de predicción lineal y eliminar el aliasing a la salida del filtro paso bajo. El análisis LPC se realiza por el método de autocorrelación con un predictor de orden de 5. La señal decimada es filtrada mediante este filtro inverso. El orden del filtro es suficiente para modelar el espectro de la señal en el rango de ffecuencias 0-1 Khz, ya que habrá a lo sumo uno o 2 formantes en este rango.

El resultado de este filtrado inverso es una señal (el error residual de predicción lineal) con espectro apróximadamente plano al haberse eliminado los efectos de la envolte devida a los formantes. Por tanto, el propósito del análisis de predicción lineal es planar espectralmente la señal de entrada de forma análoga al método de recorte central indicado en la sección 3.1.4. Ala señal filtrada inversa se le calcula la auto correlación.

Entonces el pico más alto en el rango comprendido entre 4 y 16 milisegundos se elige como el periodo delo tono . La elección de este rango de tiempo es debida a que los casos extremos de voz muy grave y muy aguda se corresponden con estos dos valores extremos.

Hay que señalar que para locutores con alto periodo del tono es innecesario el aplanamiento espectoral. Esto es debido a la ausencia de mas de un armónico (periodo de tono) en la banda de 0 - 0,8 Khz.

La clasificación sonoro-sordo se realiza cuando el nivel de pico de la autocorrelación, normalizada adecuadamente, sobrepasa o cae por debajo de un umbral establecido. Por ejemplo, estableciendo un umbral a 0,4 se prueba si el valor del pico es mayor o menor de esta cantidad. Si es mayor el segmento es sonoro y el periodo del tono es equivalente en tiempo al número de muestras a que sucede el pico. Si es menor el segmento es sordo. Sin embarga si el pico es mayor que 0.3 y los dos segmentos anteriores son sonoros, tambien se considera sonoro.

El programa VOZ-TONO implementa este proceso y permite representar la auto correlacion del error residual de predicción lineal , el valor del periodo del tono y la decición sonoro-sordo. [Las figuras 1.45 y 1.46 muestran esto para diferentes segmentos de la palabra Las.]

3.2.2. ANALISIS DE FORMANTES UTILIZANDO LOS PARAMETROS LPC ESTIMACION DE LOS ANCHOS DE BANDA/

3.2.2.0. INTRODUCCION

La función de transferencia que modela la conformación espectoral del tracto vocal es :

$$H(z) = 1/(1 - \sum_{K=1}^{p} a_k z^{-k}) \qquad (3.\$)$$

Los efectos de este modelo terminal son equivalentes a los del modelo de tracto vocal desarrollado por la teoría acústica de la voz. Las resonancias (formantes) se corresponden con los polos de cada ecuación.

3.3 Este modelo todo polos es una buena representación de los efectos del tracto vocal para la mayoria de los sonidos de la voz, Sin embargo, la teoría acústica indica que los sonidos nasales y fricativos requieren resonancias y antiresonancias (polos y ceros). No obstantes, como ya se indico,a tal demostró que los efectos de los ceros se pueden simular incluyendo más polos en la función de trnsferencia. Las raices del denominador de la ecuación 3.3. son reales o pares complejos conjugados tienen por tanto la expresión:

$$Z_k, Z_k = \exp(-T_k T) \exp(-j2 F_k T)$$

$$= \exp(-T_k T) - j\exp(-T_k T) \operatorname{sen}(2 F_k T)$$

El ancho de banda de la resonancia del tracto vocal es aproximadamente $2T_k$ y la frecuencia central es $2F_X$ (frecuencia formante)

En la representación polar en el plano Z se tendria:

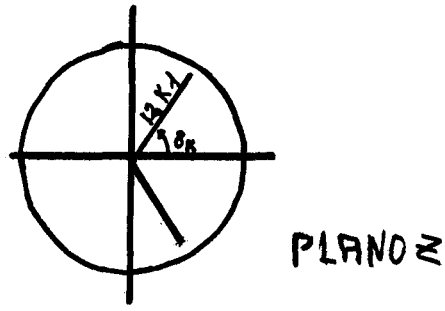


Figura 1.50. Representación polar en el plano Z de los formantes y anchos de banda

En la figura 1.50 $/Z_k/ = \exp(T_k T)$ es la distancia desde el origen al polo. Esta distancia determina el ancho de banda. El ángulo del polo en el plano Z es:

$$Q_k = 2 F_K T$$

Este ángulo estima los formantes. Para la estabilidad del modelo los polos tienen que estar dentro del circulo unidad.

Una vez presentado el significado de las frecuencias formantes y sus anchos de banda se describe a continuación la forma usual de estimar estos dos parámetros.

3.2.2.1. FORMANTES Y ANCHOS DE BANDA

Estos dos parámetros se pueden estimar usando distintas vías. Una de ellas consiste en factorizar el polinomio predictor. Como el orden del predictor se fija a priori hay que analizar p/2 polos conjugados. Así, el problema se reduce a decidir que polos son formantes y cuales no. La dificultad dificultad se encuentra en la complejidad del programa que calcula las raices del polinomio. Otra vía consiste en representar el envolvente espectoral y estimar los picos y los anchos de éstos, Esto permite realizar una primera estima de estos parámetros. Sin embargo, para tomar una decisión defi-

nitivo hay que usar bastante información lateral. Tambien hay que tener presente la desventaja inherente del método LPC al ser un modelo todo polos. Esto significa que no es claro decidir si un polo candidato a formante corresponde a un cero nasal o fricativo simulado por un polo o no. Por ello la asignación del formante se realiza mediante diversas técnicas.

Entre estas se incluyen técnicas tales como la trayectoria de formantes, cerénfacis espectoral para minimizar la posibilidad de mezcla de formantes cercanos y el uso de un contorno fuera del circulo unidad para evaluar el espectro LPC y de ese modo configurar los picos más altos del espectro.

[El programa FORMANTES implementa esta última vía] Este programa permite dibujar la envolvente espectoral y estimar a partir de ella los formatos y la anchura de estos. Las figuras 1.47 a 1.49 muestran la envolvente espectoral de diferentes segmentos de la palabra LAS y lo correspondientés valores valores de los formantes y anchos de bandas de estos. El polo F5 no es un formante su ancho de banda es muy grande 43 1 Khz.

EL ADC 0808 es el que empleo en el prototipo. Este A/D conversor es compatible con microprocesador de 8 bits.

El ADC 0808 es un componente CMOS' monolítico con un convertidor analógico digital de 8 bits, 8 canales multiplesado y control lógico compatible con microprocesador.

El conversor A/D usa el método de sucesivas aproximaciones como técnica, con 256 divisiones de voltaje.

Cuando la conversión se completa. Se carga la palabra binaria correspondiente en el latch de subida y aparece en nivel lógico en EOCí que así lo indica.

El latch de subida mantiene este dato hasta otra conversión termine y el nuevo dato sea cargado a este.

El circuito para operar en el modo libre debe estar conectado el start conversión a el EOCi (End of conversión).

Como siempre para asegurar la conversión bajo todas las posibles condiciones, es nece sario dar un pulso en el START para comenzar la conversión.

La tansferencia de datos ocurre cada 100 µseg aproximadamente para una frecuencia de reloj de 6'40 KHz aproximadamente. Este tiempo de conversión equivales a una frecuencia de muestra de 10KHz.

$$F_{muestreo} = \frac{1}{100\ \mu seg} = \frac{1}{100 \cdot 10^{-6}} = \frac{10^{6}}{100} = 10.000 \quad 10KHz$$

Esta será más que suficiente, puesto que el ancho de banda de la señal de voz es de 4KHz aproximadamente.

Esto daría una frecuencia mínima de muestreo, de 8Khz, según el teorema de Nyquist, por tanto 10KHz es más que suficiente.

Siempre tenemos que tener presente el compromiso entre número de muestras por segundo para que podamos recuperar la señal de voz, y que esta cantidad de información sea lo menor posible, para su empaquetado en memoria.

Para f muestreo de 10KHz, son diez mil muestras de señal por seg, es decir diez mil palabras de 8 bits cada una que tendremos que guardar en memoria.

$$10.000 \times F = 80.000 \text{ bits} \Rightarrow 10 \text{ Kbytes}$$

Ocuparemos 10 Kbytes de una memoria para tan solo un segundo de voz.

IDEA DEL PROYECTO.

PALABRAS DEL AUTOR.

Tengo que reconocer que la idea para construir este prototipo, me la dió el artículo de STEVE CIARCIA en la revista Norteamericana BYTE.

Después de haberlo leído atentamente, reconocí que elccircuito en sí se había quedado obsoleto.

Si bien la parte correspondiente al conversor digital/analógico estaba vigente aún, al igual que los filtros paso bajo. No ocurría así con el convertidor analógico/digital,puesto que, ya había en el mercado chips que desempeñaban todo el trabajo que a el le había costado hacer con tres chips.

Empleaba el MC 14.559 como el registro de aproximaciones sucesivas y el 74.100 como latch de salida.

Con el ADC 0808 conseguiría todo esto en un solo, y aún tenía ya incorporado el comparador y una entrada analógica multiplexada con posubilidades para 8 diferentes entradas.

Si bien yo no las uso se pueden dejar para posibles mejoras en la reducción de información, utilizando diferentes voltajes de la señal de voz y tratados adecuadamente.

El circuito se podía simplificar,no cabía duda.

La idea era la misma;tomar la señal de voz digitalizada almacenarla en ordenador y luego reproducirla.

Una descripción por bloques a grandes rasgos sería:

FIGURA 1

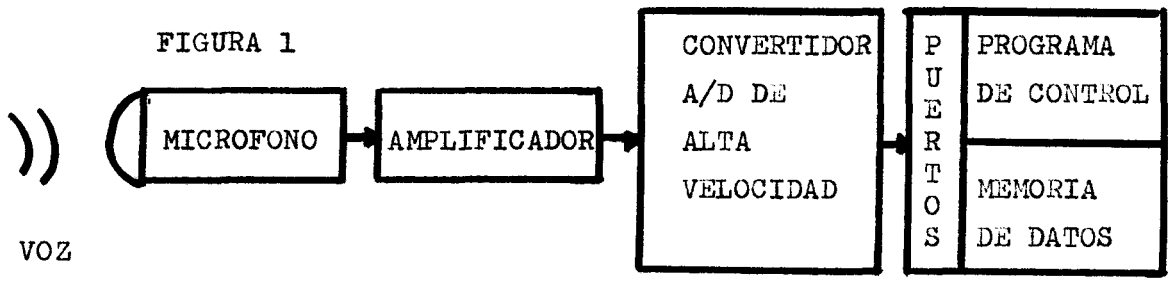


FIGURA 2

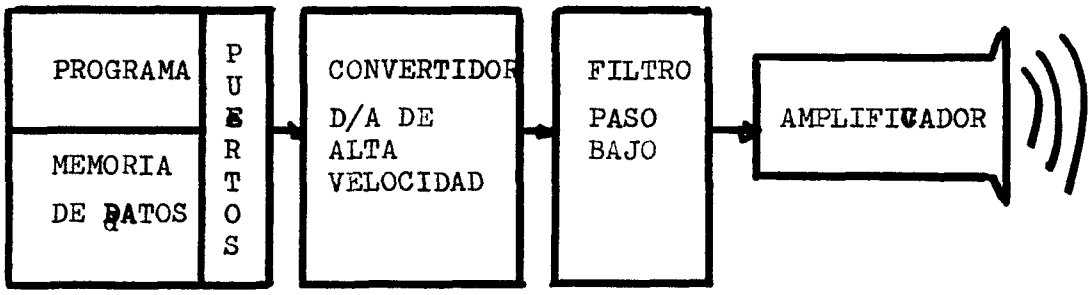


Figura 1.- SISTEMA DE GRABACION DE LA VOZ DIGITALIZADA.

La voz es tomada mediante el micrófono, esta señal es amplificada y procesada a través de un conversor A/D de alta velocidad. La onda analógica del sonido es muestreada a una frecuencia de miles de muestras por seg. estas muestras son almacenadas en la memoria programada del computador.

Figura 2.- DIAGRAMA DEL SISTEMA DE REPRODUCCION DE VOZ DIGITALIZADA.

Las muestras almacenadas en memoria son sacadas y procesadas a través de un conversor D/A de alta velocidad para redondear la señal emplearemos un filtro paso bajo, es amplificada y reproducida por un altavoz.

DIGITALIZACION DE LA VOZ

La digitalización de la señal de la voz es una sencilla tecnica de adquisición de datos con esta nueva definición.

Durante años los cientificos han estado usando las computadoras para muestrear señales analógicas digitales y almacenarlas en memoria, Frecuentemente en aplicaciones de alta velocidad como por ejemplo túneles de viento para comprobar maquetas de aviones o en experimentos nucleares , la velocidad de muestreo puede exceder de miles de muestras por segundo. En casos de sucesos de corta duración estas miles de muestras se almacenan en memoria.

Cada muestra es sucesivamente procesada a travéz de un conversor digital/ analógico en un rango más lento, con lo que el estudio del suceso se facilita bastante.

La voz, cuando es traducida por un micrófono pasa a ser una señal analógica variable en amplitud y frecuencia. Al introducir esta señal analógica en un conversor analogico digital y luego almacenada en la memoria programable del ordenador, este no sabe reconocer si estas muestras son sucesos nucleares, experimentos o de la señal de voz.

A la hora de reproducir la voz basta con un programa que se encarge de ir sacando por los puestos elegidos para salida, las muestras consecutivas, con la regla de primera en entrar primera en salir. Como vemos debemos programar la memoria como si de una FIFO se tratase (First imput First output

Un sistema de voz digitalizada crea una onda analógica usando el conversor digital/analógico, que si hemos tomado el número de muestras adecuadas podremos reproducir enteramente la señal de voz. anteriormente tomada, como ocurrirá con la voz sintetizada

Un dato importante muy influyente en la síntesis de la voz, va a ser la frecuencia de muestreo, es decir, la cantidad de muestras de una palabra que tengamos que almacenar para después reproducir la palabra aceptadamente .

ESCOGIENDO EL CORRECTO TIEMPO DE MUESTREO

La voz humana normalmente ocupa un ancho de banda de unos 4000 Hz.

Se puede demostrar que este ancho de banda se reduce a 1' 5 KHz por encima de esta frecuencia se encuentran los sonidos graves fuertes, y la entonación y el acento.
Hay una ley que limita esta velocidad de muestreo para una correcta reproducción de la señal digitalizada, es criterio de Nyquist. Esta señal nos dice que la frecuencia máxima, como, mínimo, de la señal a muestrear.

F muestreo = 2 fmáx.

Si la voz humana se extiende hasta los cuatro KHz, la frecuencia mínima de muestreo será de 8 KHz. Esto supone que la salida habrá un filtro ideal, cosa que es imposible ya que se podría demostrar matematicamente que un filtro ideal tiene respuesta aunque no haya exitación a la entrada, cosa, que es irrealizable en la práctica. En realidad la frecuencia de muestreo tendrá que ser tres o cuatro veces la máxima frecuencia de la señal a muestrear, es decir, 12 KHz ó 16 KHz.

En la figura siguiente, muestra como una señal analógica variable en frecuencia y amplitud es muestreada a diferentes rangos. Como anteriormente había dicho la mayoria de los sonidos de la voz están alojados en un ancho de banda de 1.500 Hz, y por encima de esta frecuencia se encuentra la entonación y el acento. Esto hace que la voz de una persona sea diferente a las otras.

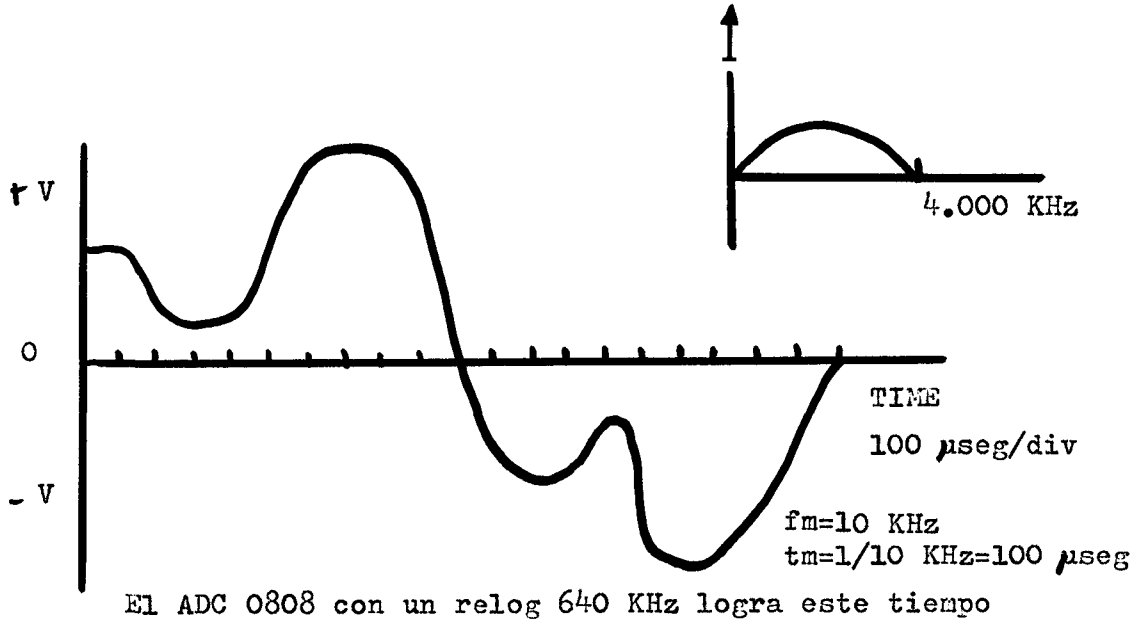
En la fig. 1, está la señal analógica a muestrear, con un frecuencia fundamental de aproximadamente 500Hz y algunos componentes de alta frecuencia.

En la fig. 2, la señal es muestreada a un rango de 5.000 muestras por segundo y el resultado de la muestra es mostrado.

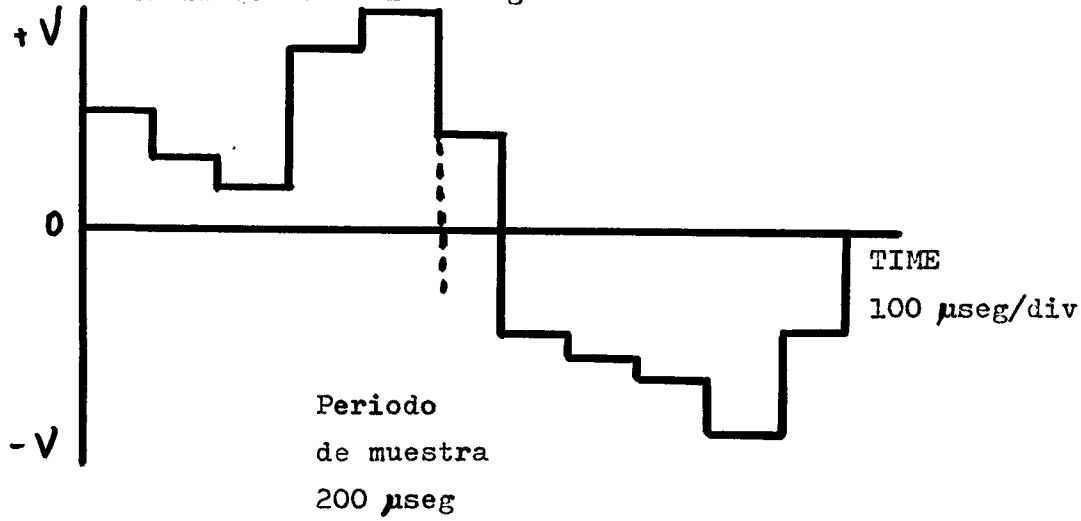
Se ve a simple vista, que recuerda vagamente a la señal original.

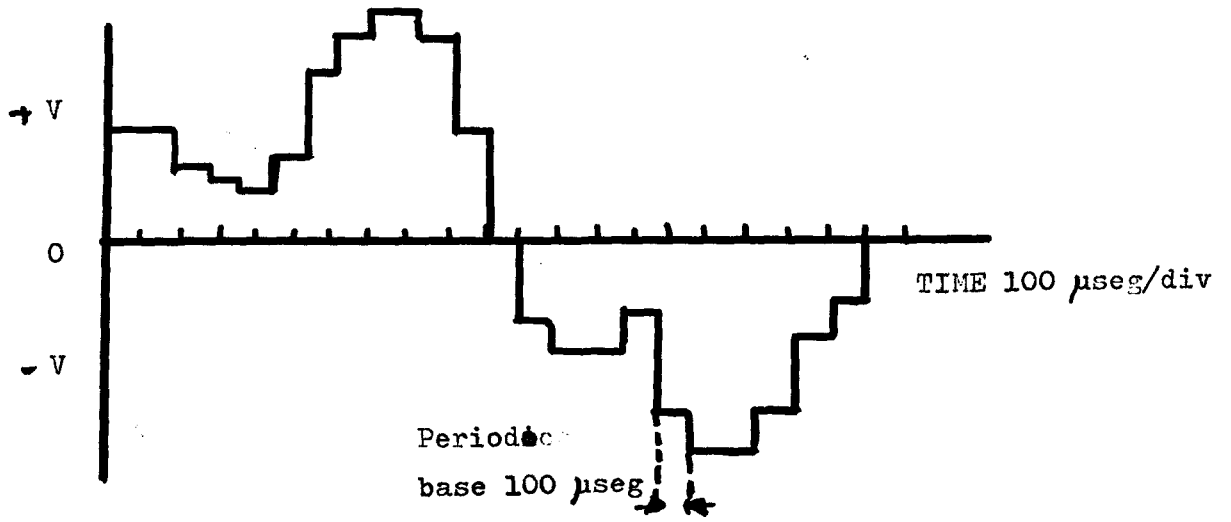
Si reproducimos esta señal digitalizada la señal análoga que obtendremos a la salida del filtro paso bajo va a ser bastante pobre, y no reproducirá exactamente a la señal original.

Fig. 3.- Incrementando el rango de muestreo a 10.000 muestras por segundo, notamos que recuerda perfectamente a la señal original y al reproducirse, basta pasarla por un filtro paso bajo para que alise los picos de la señal y con ello obtendremos la señal original bastante aproximada.



El ADC 0808 con un relog 640 KHz logra este tiempo de muestreo de 100 useg





PROBLEMAS A CONSIDERAR

Los beneficios asociados con el reducido costo del circuito de entrada y salida de la voz digitalizadan están en oposición con la gran cantidad de memória que vamos a necesitar En el ejemplo previo si alamcenamos las muestras del resultado de muestrear a un rango de 10.000 muestras por segundo. En la tabla de memoria nesecitamos para cada segundo de voz unos 10.000 bytes (Suponiendo que el convertidor sea de 8 bits por palabra). Si incrementamos este rango para aumentar la calidad de la voz, para una frecuencia de 16KHz dará por cada segundo de voz una capacidad de memoria de 16 Kbytes.

En todo caso mi prototipo es experimental y el trabajo con el que será realizado con el sistema de desarrollo MDS 221, con una memoria programable de 64 Kbytes podré conseguir una frecuencia de muestras de 10KHz.

$$\frac{64\text{kbytes}}{10\text{KHz}} = 6'4 \text{ seg de voz}$$

Ampliables en caso de que utilizar un diskette esclusivamente para almacenar datos, mediante la creación de un fichero esclusivo.

En todo caso, el trabajo con la parte Software del prototipo, dará la medida de poder reducir la cantidad de información sin con ello perder calidad en la reproducción de esta, con ello complicando el Sofware podremos reducir la memoria utilizable,nos interesa reducir la memoria no solo por ser una de las partes más caras del prototipo, sino para la simplificación y mayor capacidad de nuestra biblioteca de palabras en este caso, palabroteca.

Con una mayor complicación del Software y del Hardware, podremos trabajar con fonemas tendríamos una, fonomoteca.

Pero sólo con esta primera y más sencilla parte del prototipo nos podremos asombrarnos de los resultados.

CONSTRUYENDO UN DIGITALIZADOR DE VOZ

Para experimentar con la voz digitalizada es necesario tener un conversor analógico digital de alta velocidad una memoria programable para almacenar, los datos y un conversor digital analógico para reconstruir la entrada analógica.

El prototipo necesita un sistema mínimo con microprocesador el 8085 de 8 bits compatible con los conversores con palabras de 8 bits. Una memoria ROM donde vayan alojados los programas de control y la RAM necesaria para los datos. Este es a grandes rasgos el sistema mínimo utilizado.

Yo particularmente lo he realizado en la tarjeta SDK 85 que cumple mis especificaciones anteriores y además tiene los puertos necesarios para la entrada y salida de un teclado, desde el cual puedo introducir el Software del prototipo. En su fase de desarrollo, he empleado el ICE, el circuito emulador bajo el control del sistema de desarrollo MDS de la casa Intel. El ICE me resuelve el problema de añadirle la memoria suplementaria al SDK, ya que gracias al ICE me es suficiente para las pruebas y realización de mejoras.

En la fase de desarrollo y el de la corrección de errores, así como el de desarrollo del Software el MDS ha tenido un papel indiscutiblemente importante.

VENTAJAS DE LA UTILIZACION DE MICROPROCESADORES

Los microprocesadores simplifican todas las fases de desarrollo de un proceso. El primer paso, como en cualquier programa de desarrollo, es identificar las distintas funciones que necesita el sistema final. En lugar de implementar estas funciones con mallas de puertas y flip-flops en el microprocesador están ya implementadas mediante secuencias de instrucciones (programas) en los elementos de memoria. Los datos y ciertos tipos de programas se almacenan en memoria RAM, mientras que el programa básico puede almacenarse en ROM .

El mócroprocesador realiza todas las funciones del sistema buscando las intrucciones en la memoria, ejecutando las y comunicando los resultados a través de las puertas de entrada/salida.

Los beneficios del diseño de sistemas mediante microprocesadores, por tanto, están a la vista. La ventaja más ostensible es el descanso de coste del harware. Un circuito integrado sustituye docenas de elementos de lógica aleatoria, reduciendo tanto el costo como el tamaño del sistema. Además el coste de producción disminuye enormemente, al existir menos elementos y reducirse el número de circuitos impresos (dificiles de verificar y corregir). Pero quizá la ventaja más grande de los microprocesadores sea su flexibilidad, modificar su sistema consiste simplemente en modificar un programa, sin tener que rediseñar el sistema completo. Asi mismo, es importante su fiabilidad.

Al disminuir el número de componentes disminuye asi mismo la posibilidad de avería, puesto que todas las funciones de lógica de control se utilizan numerosos componentes, están ahora implementados en un simple elemento ROM, que no es volátil, es decir cuyo contenido no desaparece al bajar o desaparecer la potencia pero si modificable.

CONFIGURACION DE UN SISTEMA

La placa con los conversores A/D Y D/A es conectada a un sistema mínimo;microprocesador, memoria y puertos, en la práctica es conectado a una tarjeta como puede ser la SDK de INTEL o la ALECOP de la universidad de DEUSTO Ambas utilizan el microprocesador 8085 por ello me permito hacer un estudio semiprofundo de sus posibilidades para que luego se comprenda la programación de control del prototipo.

CONFIGURACION DE UN SISTEMA

Un computador digital consta básicamente de los siguientes bloques:

a) Unidad central de Proceso (CPU)
b) Unidad de Memoria
c) Puertas de Entrada/Salida

La unidad de memoria sirve como almacenamiento de las instrucciones, palabras de información codificadas que definen las operaciones a realizar por la CPU, y los datos, información asi mismo codificada que se procesa por la CPU

Un programa es un conjunto de instruccipnes colocadas coherentemente y almacenadas en la memoria. CPU "lee" cada instrucción de la memoria en una secuencia lógica y determinada utilizando esta información para iniciar un proceso Si la secuencia de programa es correcta, el proceso del mismo originará resultados inteligentes y útiles.

La memoria se utiliza también para almacenar los datos a manipular por las instrucciones del programa. Este debe estar colocado de tal forma que la CPU no tome una palabra que no corresponda a una instrcción cuando está esperando una instrcción.

La CPU puede acceder rápidamente a cualquier dato almacenado en la memoria, pero en ocasiones ésta no es lo suficientemente grande para almacenar el banco de datos completo requerido en una aplicación determinada. Este problema se puede resolver dotando al computador de una o más puertas de entrada de datos. La CPU puede direccionar estas puertas, permitiendo al computador recibir información de equipos externos al mismo (como pueden ser una lectora de cinta de papel, un floppy disk, ect.) a través de las mismas a una gran velocidad y en gran cantidad.

Aparte de puertas de entrada, un computador precisa de puertas de salida que permitan a la CPU comunicar los resultados de su procesos al exterior. El dispisitivo de salida, por ejemplo, puede ser una pantalla, para uso de un operador humano, o un dispositivo que origine una copia de

la información como una impresora, o un elemento de almacenamiento exterior como un floppy disk, o incluso puede consistir en las señales de control de un proceso que origine las operaciones de otro sistema., como una línea de montaje automático. Al igual que las puertas de entrada, las de salida son direccionales. El conjunto de ambas permiten al ordenador comunicarse con el mundo exterior.

La unidad central de proceso es el elemento regidor del sistema, controlando las operaciones realizadas por otros componentes. La CPU debe permitir buscar e identificar sus instrucciones en la memoria, decodificar su contenido binario y ejecutarlas. Debe también ser capaz de referenciar la memoria y las puertas de entrada y salidas necesarias en la ejecución de las instrucciones, teniendo además que reconocer y responder a ciertas señales de control externas, como son las demandas de INTERRUPCION. Las unidades funcionales que conforman la CPU y que le permiten la realización de estas funciones se describen a continuación.

ARQUITECTURA DE CPU

Una unidad central de proceso (CPU) convencional consta de las siguientes unidades funcionales interconectadas:

- Registros
- Unidad Lógica/Aritmética (ALU)
- Lógica de Control.

Los registros son unidades de almacenamiento temporal en el interior de la CPU. Algunos de ellos, como el contador de programa o el registro de instrucción, tienen una función específica. Otras, como el acumulador, tienen una , tienen un campo de ampliación más amplio.

ACUMULADOR

Generalmente, el acumulador almacena un operando a ser manipulado por la ALU. Una instrucción típica a realizar por esta unidad funcional puede ser la de sumar el contenido de cualquier otro registro de la CPU al acumulador y guardar el resultado en este último, lo que da idea de la

versatilidad del mismo, pudiendo ser, por regla general, a la vez un registro fuente (operando) y destino (resultado) La CPU incleye asi mismo otros registros que pueden usarse como almacenamiento temporal de operandos o datos inmediatos. Con ello se evita el tener que efectuar desplazamientos de datos constantes entre el acumulador y la memoria (típico lugar de desplazamiento de todo tipo de datos), dando lugar a un proceso mucho más rápido y eficiente.

CONTADOR DE PROGRAMA PC

Un programa consta de una serie de instrucciones que están almacenadas secuencialmente en la memoria del sistema El procesador central, por tanto, va consultando el contenido de la misma a fin de determinar cuál es la operación a realizar. Para ello, se debe tener una referencia respecto a cuál es la posición de memoria en que está contenida la próxima instrucción a ejecutar.

Cada una de estas posiciones de memoria tiene asociado un número, a fin de distinguirlas de las demás. Al número asociado que identifica una posición de memoria se le denomina DIRECCION.

El contador que contiene la dirección de la siguiente posición a ejecutar se denomina CONTADOR DE PROGRAMA. El Procesador Central va incrementado unidad al contenido de dicho contador cada vez que se ejecuta una instrucción, de tal manera que siempre contiene la dirección de la instrucción siguiente. Esto hace que el programador deba almacenar las instrucciones de un programa en direcciones numéricamente adyacentes colocando en las direcciones más altas, las últimas.

Esta regla de ejecución secuencial se cumple siempre, excepto cuando la instrucción contenida en una posición de memoria es la de salto a cualquier otra posición de memoria es la de salto a cualquier otra posición determinada. Una instrucción de este tipo contiene la dirección de la próxima instrcción a ejecutar, que puede estar

almacenada en cualquier posición de la memoria, siempre y cuando no sobrepase los límites de la misma.

Cuando tiene lugar una operación de salto., el procesador Central sustituye el contenido de su contador de su programa por la dirección especificada en la instrucción, por lo que la ejecución continúa en esta dirección.

Existe, no obstante, un tipo especial de salto, que es llamada a una subrutina. En este caso el procesador necesita por las razones que más tarde se expondrán, "recordar" el contenido del contador de programa en el momento inmediatamente anterior a cuando tiene lugar el salto.

Una SUBRUTINA es un programa dentro de un programa. Generalmente consiste en un grupo de instrucciones que realizan una operación determinada que se repite varias veces a lo largo del programa total. Por ejemplo, unas aplicaciones típicas de subrutinas pueden ser las de calcular el cuadrado, el seno o el logaritmo de una constante, o extraer o introducir datos en algún periférico.

Para asegurar que una vez ejecutada la subrutina, la ejecución del programa retorna al lugar adecuado en el programa principal, cuando el procesador reconoce una instrucción de llamada, incrementada al contador de programa y almacena su contenido en un memoria reservada, denominada STACK,

El stack contiene, por tanto, la dirección de la intrucción siguiente a la que realiza la llamada. Es entonces cuando se carga en el contador de programa la dirección especifica por la instrucción de llamada, que será la primera a ejecutar por la subrutina.

La última instrcción de la subrutina es una de RETORNO. Esta instrucción no especifica ninguna dirección, puesto que cuando una instrucción de este tipo es reconocida por el procesador, simplemente sustituye el contenido actual del contador del Programa por la dirección colocada en la parte superior del stack, lo cual origina que la ejecución

del programa se realice corectamente a partir de la instrucción siguiente a la de llamada.

En muchas ocasiones la subrutina se encadenan. Es decir, una primera subrutina puede llamar a una segunda, la segunda a una tercera y así sucesivamente. Esto es totalmente correcto, siempre y cuando se disponga de la capacidad de almacenamiento suficiente para guardar todas las direcciones sucesivas de retorno. Es decir, el nivel máximo de llamada a subrutina viene determinado por el propio tamaño del stack. Si este tiene capacidad para almacenar tres direcciones de retorno, unicamente puede ejecutarse tres niveles de llamada a subrutina, pues si pasamos de este límite se perderian algunas direcciones de retorno y la ejecución del programa no se realizaria correctamente,

Según el tipo de procesador, existen varias formas de confeccionar los stack. Algunos tienen la posibilidad de guardar las direcciones de retorno en el propio procesador. Otros, en cambio, utilizan una determinada zona de la memoria externa como stack, disponiendo de un registro de PUNTERO en el cual está contenida la dirección que ha entrado en último lugar en el stack, que es la primera dirección de retorno que tendrá lugar. La ventaja de utilizar un stack externo estriba en que practicamente permite una concatenación ilimitada de subrutinas. Por otra parte, si el procesador dispone de instrucciones que permitan colocar o sacar del stack el contenido del acumulador o de cualquier otro registro a través de la dirección almacenada en el puntero de stack, es posible realizar interrupciones de varios niveles. De esta forma el estado actual del procesador (por ejm. , el contenido de todo su registro) puede preservarse en el stack cuando se acepta una interrupción, y restaurarse una vez que la misma ha sido procesada. Esta facultad de conservar los estados del procesador es asi mismo factible si la rutina de servicio de una interrupci

ón es asu vez interrunpida.

REGISTRO Y DECODIFICADOR DE INSTRUCCION

Todo computador tiene una LONGITUD DE PALABRA que es característica de la máquina. La longitud de la palabra de un computador biene generalmente determinada por el tamaño de los elementos de almacenamientos internos y los buses de interconección, Por ejem, un computador cuyos registro y buses puedan almacenar y transmitir respectivamente 8 bits simultaneamente, tiene una longitud de palabra característica de 8 bits, y se definirá como un procesador de 8bits paralelo.

Un procesador de este tipo será más eficiente si está distribuido en campos binarios de 8 bits, y su memoria asociada está así mismo dispuesta en unidades direccionable de una capacidad unitaria de almacenamiento de 8bits.

Los datos e instrcciones por tanto, estarán contituidos por números binarios de 8bits, o bien multiplos de estos es decir, 16bits, 24bits, etc. Este campo de 8 bits caracterítica se denomina BYTE.

Cada una de las operaciones que pueda realizar el Procesador se identifica por un único byte de datos, conocido como CODIGO DE INSTRUCCION, o CODIGO DE OPERACION. Como sea que con 8 bits pueden realizarse 256 combinaciones binarias, obtenemos para este procesador un nº máximo de codigos de instrucción distintos de 256, que generalmente es más que suficiente para cualquier procesador.

El primer proceso que realiza el computador en la ejecución de una instrucción es el de búsqueda e identificación de la misma. Este proceso generalmente se realiza mediante dos operaciones: en primer lugar, el procesador envia el contenido de sú contador de programa al bloque de memoria a lo que esta retorna al procesador el byte contenido en la posición direccionada. La CPU almacena este byte de instrucción en un registro denomimado REGISTRO DE INSTRUCCION

usándolo para definir cuál es la operación a realizar por la ejecución de la instrucción.

Para ver con más claridad la forma en que el procesador pasa de un código de instrucción a la ejecución de una operación determinada podemos imaginar que el registro de instrucción se decodifica de tal manera que activa una entre un determinado nº de lineas de salida, que en el caso del procesador de 8bits serían 256.Cada una de estas lineas representa la puesta en marcha de una operación asociada a un código determinado. La linea actuada, combinada con una serie de impulsos temporizados da lugar a la iniciación de la secuencia de ejecución de la operación . Este paso de un código a una acción determinada es la función que realiza el DECODIFICADOR DE INSTRUCCION y su lógica asociada.

Normalmente, y siempre basandonos en un procesador de 8 bits, un código de instrucción de 8bits es suficiente para definir una operación determinada. Sin embargo, en ocasiones la ejecución de la instrucción requiere más información que la que puede proporcionar un sólo byte. Tal es el caso, por ejem. de aquellas instrcciones que hacen referencia a una posición de memoria. En esta se precisa, aparte del código de la operación a ejecutar, la dirección de dicha posición de memoria. Para ello son necesarios como mínimo dos bytes para definir dicha dirección, lo que origina que existan instrucciones que requieran dos o tres bytes para ser definida en su totalidad.

Cuando esto ocurre, los bytes que las componen están colocados en posiciones contiguas de memoria, realizando el procesador dos o tres operaciones de busqueda a fin de ejecutar completamente la instrucción. El primer byte encontrado en la memoria se colocan el registro de instrucción del procesador, colocandose los siguientes en registros de almacenamiento temporal.

REGISTRO (S) DE DIRECCIONAMIENTO

En la CPU se utiliza un registro o un par de ellos para guardar la dirección de una posición de memoria a la que se va a acceder para tomar datos (no confundir con el registro del contador de programas).

Si el registro de dirección es programable (por ejem., si hay instrucciones que permiten al programador modificar su contenido), el programa puede generar una dirección en este registro antes de ejecutar una instrucción de refencia a memoria, como ocurre con una instrucción que lea datos de la memoria, escriba datos en la misma u opere con datos contenidos en ella,.

UNIDAD LOGICA/ARITMETICA (ALU)

Todos los procesadores tienen una unidad lógica/aritmética, a la que se denomina más comunmente como ALU. La ALU como su nombre indica, es la parte de la CPU donde se realiza operaciones lógicas y aritméticas con datos binarios.

La ALU contiene un sumador capaz de convinar los contenidos de dos registros deacuerdo con la lógica de la aritmética binaria. Ello permite realizar manipulaciones aritméticas con datos obtenidos de la ,memoria u otros dispositivos de entrada.

Mediante el uso del sumador básico, un programador capacitado puede realizar rutinas que efectuen operaciones de restas, multiplicación y división, proveyendo a la máquina de un potencial operativo aritmético complejo. No obstante, la gran mayoria de las ALUs ya que obtienen algunas de estas funciones, tales como la resta, operaciones de álgebra boolean y posibilidad de desplazamiento de registro.

A fin de potenciar su funcionamiento, la ALU contiene bits de condición, que definen ciertas condiciones en el proceso de operaciones lógicas y aritméticas. Los bits de condición generalmente utilizados son los de arrastre, cero, signo y paridad. La existencia de los mismos hace posible programar saltos condicionados al estado de uno o más de ellos.

Así, por ejem., pueda haber un programa en el que se salte a una determinada rutina si el bits de arrastre está exitado después de una operación de sumas.

LOGICA DE CONTROL

La lógica de control regula el funcionamiento de una Unidad Central de Proceso. Utilizando entradas de reloj, la lógica de control proporciona la secuencia de ejecución de un proceso. Una vez que una instrucción ha sido decodificada, la lógica de control genera las señales necesaria (ya sea para circuitos internos o externos a la CPU) para iniciar el proceso de ejecución.

En muchas ocasiones la lógica de control tiene la posibilidad de responder las señales externas, tales como una demanda de interrupción o de espera. Una demanda de interrupción hace que se interrumpa temporalmente las ejecución del programa en curso, saltando una rutina de servicio del dispositivo de interrupción, cuya ejecución una vez finalizada provoca de nuevo un retorno al programa que se estaba ejecutando.

Por contra cuando existe una demanda de espera(WAIT) se debe frecuentemente a que un elemento de memoria o entrada/salida opera más lento que la CPU . La lógica de control para la CPU en espera de que la memoria o la puerta de entrada /salida esten preparadas para operar con datos.

OPERACIONES DEL MICROPROCESADOR

Existen ciertas operaciones que son básicas para casi todos los microprocesadores, por lo que para examinar las operaciones específicas de un determinado microprocesador es necesario, en primer lugar tener una amplia visión de las mismas.

BASE DE TIEMPO

El trabajo que ejcuta un procesador es cíclico. Este busca e identifica una instrucción, realiza las operaciones necesarias, busca la siguiente instrucción y así sucesivamente. Esta secuencia ordenada de operaciones requiere una temporización por lo que la CPU necesita un reloj que sirva como referencia a todas sus acciones.

Al conjunto de identificación y ejecución de una instrucción simple se le llama CICLO DE INSTRUCCION. La parte de un ciclo que define claramente una actividad se llama ESTADO, y el intervalo entre los pulsos de reloj se denomina PERIODO. Por regla general, son necesarios unos o más periodos para configurar el estado, mientras que existen varios estados en un ciclo.

BUSQUEDA DE INSTRUCCION (FETCH)

El primer estado de cualquier ciclo está dedicado a la busqueda de la instrucción. La CPU, acompañada de una señal de lectura, envia el contenido del contador del programa a la memoria, la cual envia a la CPU el byte contenido en dicha posición . El primer byte de una instrucción se coloca en el registro de instrucción . Si la intrucción consta de más de un byte, se precizan más estado, a fin de buscar cada uno de ellos. Cuando la instrucción está en su totalidad en la CPU, el contador de programa se incrementa(preparando se para la busqueda de la proxima instrucción) y la instrucción se decodifica.Las operaciones definidas por ellas se ejecutan en los restantes estados del ciclo.

LECTURA DE MEMORIA

En realidad, el proceso de busqueda de una instrucción es una operación de lectura de memoria especial, colocando los datos en el registro de instrucción de la CPU.

No obstante la instrucción identificada puede hacer que se deseen leer datos desde una determinada posición de la memoria. Para ello , la CPU envia una señal de lectura y la di

rección apropiada; la memoria le responderá enviando el byte contenido en la posición correspondiente a la dirección enviada. Estos datos se colocan en el acumulador o en cualquier otro de los registros de trabajo de la CPU.

ESCRITURA EN MEMORIA

Una operación de escritura en memoria es muy similar a una de lectura, con la diferencia en la dirección de transmisión de los datos. En tanto que en la operación de lectura los datos proseguían de una posición de memoria, en una operación de escritura, la CPU , enviando una señal de escritura, envia así mismo la dirección adecuada y los datos que deben escribirse en dicha dirección, destruyendo los que existieran anteriormente en la misma.

Como se ha dicho antes, las actividades de un procesador están temporizadas mediante un reloj, cuyo periodo termina la velocidad del proceso. Esta velocidad no obstante, está limitada por el TIEMPO DE ACCESO a la memoria, una vez que el procesador ha enviado una dirección de lectura de memoria, no puede realizar ninguna otra operación hasta que la memoria le responda. Existen muchas memorias cuyo tiempo de acceso, o de respuesta, es más rápido que el propio procesador, pero existen otras, sin embargo, que no pueden responder en el espacio de tiempo definido por el reloj del procesador Para evitar que la ejecución de las operaciones resulte errónea, y como sea que el procesador necesita tener una sincronización con la memoria es preciso que exista un estado de espera de la CPU. Cuando la memoria recibe una señal de lectura o escritura envía a su vez una señal a la CPU, que lo detiene temporalmente. Una vez que la memoria ha tenido tiempo de responder, desbloquea la linea continuando el proceso del ciclo de instrucción.

ENTRADA/SALIDA

Las operaciones de entrada y de salida son muy similares a las operaciones de memoria, con la salvedad de que en lugar de direccionarse una posición de memoria, se direcciona un dispositivo de entrada o de salida. Pra ello, la CPU envia una señal de control de entrada o de salida, aconpañada de la dirección del dispositivo correspondiente, enviando los datos al mismo, cuando se trata de una operación de salida, o recibiendolos si la operación es de entrada.

Estos datos pueden trasmitirse en serie o paralelo, aunque siempre el código binario. Cuando la trasmisión se realiza en paralelo, la transferencia de los bits que componen la palabra, que pueden estar a cero o a uno, se realiza simultaneamente con un bit por cada línea. Por lo contrario en la transmisión de serie, los bist se transfieren sucecivamente por la misma línea. Como es natural, la transmisión en serie es más lenta, pero precisa de un harware más simple que la transmisión en paralelo.

INTERRUPCIONES

La mayoría de los procesadores incluyen la posibilidad de realización de interrupciones. Imaginemos el caso de un computador que está procesando una gran cantidad de datos, parte de los cuales deben enviarse hacia una impresora. La CPU puede enviar un byte de datos durante cada ciclo de máquina , pero la impresora puede tardar varios ciclos en imprimir el carácter. Podría hacerse que la CPU estuviera en un estado de espera en tanto que la impresora no le da la señal de que puede recibir un nuevo carácter, pero sería un tiempo perdido por la máquina. Por el contrario, si la máquina está preparada para aceptar interrupciones, puede enviar un caracter y proseguir con la ejecución del programa. Cuando la impresora esta lista para recibir un nuevo carácter, efectua una demanda de interrupción, que cuando es aceptada por la CPU, suspende la ejecución del programa, yendose automáticamente hacia la subrutina que envia el siguiente byte de datos.

Una vez que este ha sido transmitido continua la ejecución del programa, y así sucesivamente.
Esta operación es muy similar a una llamada de subrutina excepto en el que el salto se produce debido a una causa externa al propio programa que se está ejecutando.

HOLD

Otra facultad que frecuentemente tienen los procesadores es el Hold, lo cual actúa las operaciones de Acceso Directo a Memoria.

Normalmente en las operaciones de entrada/salida, el proceso por si mismo controla la transferencia de datos en su totalidad. La información que debe colocarse en la memoria se transfiere desde el dispositivo de entrada al procesador y este lo trasmite a la posición de memoria deseada. De forma similar, una información que va desde la memoria a un dispositivo de salida, lo hace así mismo a través del procesador.

No obstante, existen dispositivos capaces de transmitir información de o hacia la memoria a mucha mayor velocidad de la que tiene el procesador al realizar la transferencia. En estos casos, la transferencia puede realizarse directamente, accediendo el dispositivo trasmisor a la memoria, prescindiendo del procesador.

Para evitar que haya acceso simultáneo a la memoria, por parte del dispositivo y del procesador, este suspende temporalmente sus operaciones de transferencia, diciendose en este caso que entra en el estado de Hold.

ORGANIZACION GENERAL DE UN MICROPROCESADOR (8085)

Desde el punto de vista del programador, el computador puede presentarse como que consta de las partes siguientes.

(1) Siete registros de trabajo en los que tienen lugar todas las operaciones con datos, y que constituyen el medio para direccionamiento de la memoria.

(2) Memoria, en la que almacenan instrucciones de un programa de datos, y a la que pueda accederse a cada una de sus posiciones independientemente, a fin de tomar la información deseada.

(3) El contador de programa, cuyo contenido indica la próxima instrucción a ejecutar.

(4) El puntero de pila, un registro que dispone parte de la zona de la memoria para ser utilizada como pila.
Ello facilita la ejecución de subrutinas de maniobras de interrupción, tal como se describe posteriormente.

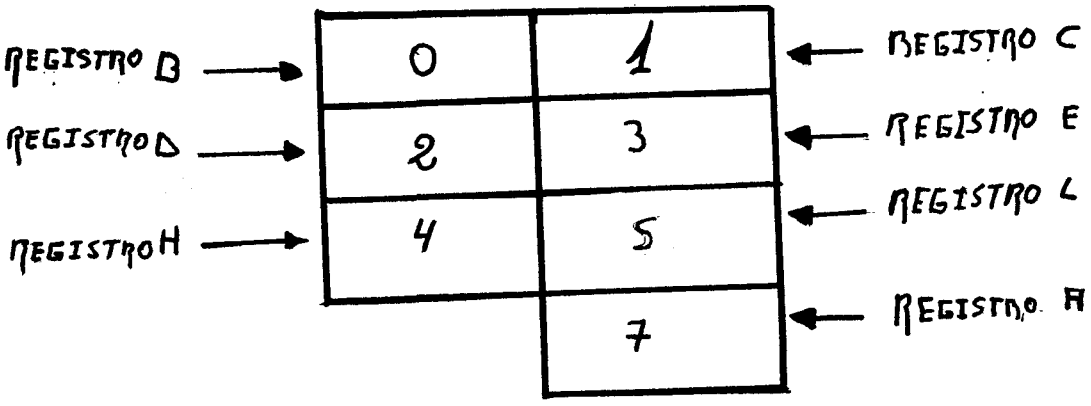
(5) Entrada/salida, que es la unión entre un programa y el mundo exterior.

REGISTROS DE TRABAJO

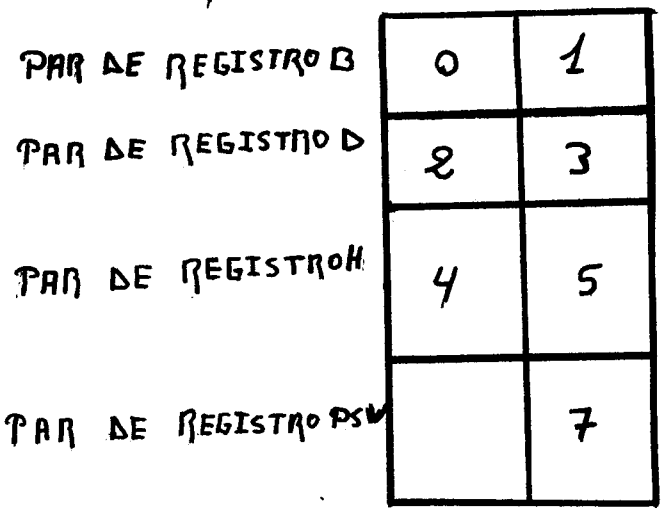
El 8085 pone a disposición del programador un registro acumulador de 8 bits, y seis registros numerados como 0, 1, 2,3,4 y 5, pudiendo tambien accederse a los mismos mediante las letras B, C, D, E, H, L y A (para el acumulador) respectivamente.

Algunas operaciones del 8085 hacen referencia a los registros de trabajo por pares, utilizándose entonces las letras B, D, H, y PSW.

REFERENCIA INDIVIDUAL DE REGISTROS



REFERENCIA DE PARES DE REGISTRO



NOTA: Cuando se especifica el par de registros PSW, los 8bits más significativos a los que se le hace referencia, constituyen un byte especial cuyo contenido se detalla posteriormente al hablar de instrucciones con pares de registros.

MEMORIA

El microprocesador puede utilizarse con memoria de solo lectura, memoria programable de solo lectura y memoria de lectura/escritura. Durante la ejecución de un programa pueden leerse datos desde cualquiera de los tres tipos de memo

ria, pero únicamente puede escribirse en la memoria de lectura/escritura.

Desde el punto de vista del programador,la memoria es una secuencia de bytes, cada uno de los cuales puede almacenar 8 bits (dos dîgitos hexadecimales). El número máximo de bytes de memoria direccionables es de 65.536 (0000H hasta FFFFH), que es el mayor número representable con 16 bits.

CONTADOR DE PROGRAMA.

El contador de programa es un registro de 16 bits,accesible para el programador, y cuyo contenido indica la dirección de la próxima instrucción a ejecutar, tal como se describe en esta parte en el apartado de Representación en Memoria de un Programa.

PUNTERO DE PILA (STACK POINTER).

El stack es un área de la memoria, definida por el programador, y en la que se almacenana datos o direcciones, modificables por las operaciones de stack. Estas operaciones tienen lugar a causa de determinadas instrucciones del 8085 y ayudan en la ejecución de subrutinas y maniobras de las interrupciones. El programador define hasta qué dirección debe de ocupar el stack mediante un registro de 16 bits,llamado puntero de stack.

ENTRADA/SALIDA

La comunicación de 8085 se realiza mediante 256 dispositivos de 256 de salida, cada uno de los cuales recibe o envía, datos respectivamente desde el acumulador, Cada uno de estas dispositivos tiene asignado un número de 0 a 225, controlado por el programador. A las instrucciones que dan lugar a que se realicen transmisiones se las denomina de entrada/salida.

REPRESENTACION DE UN PROGRAMA DE MEMORIA

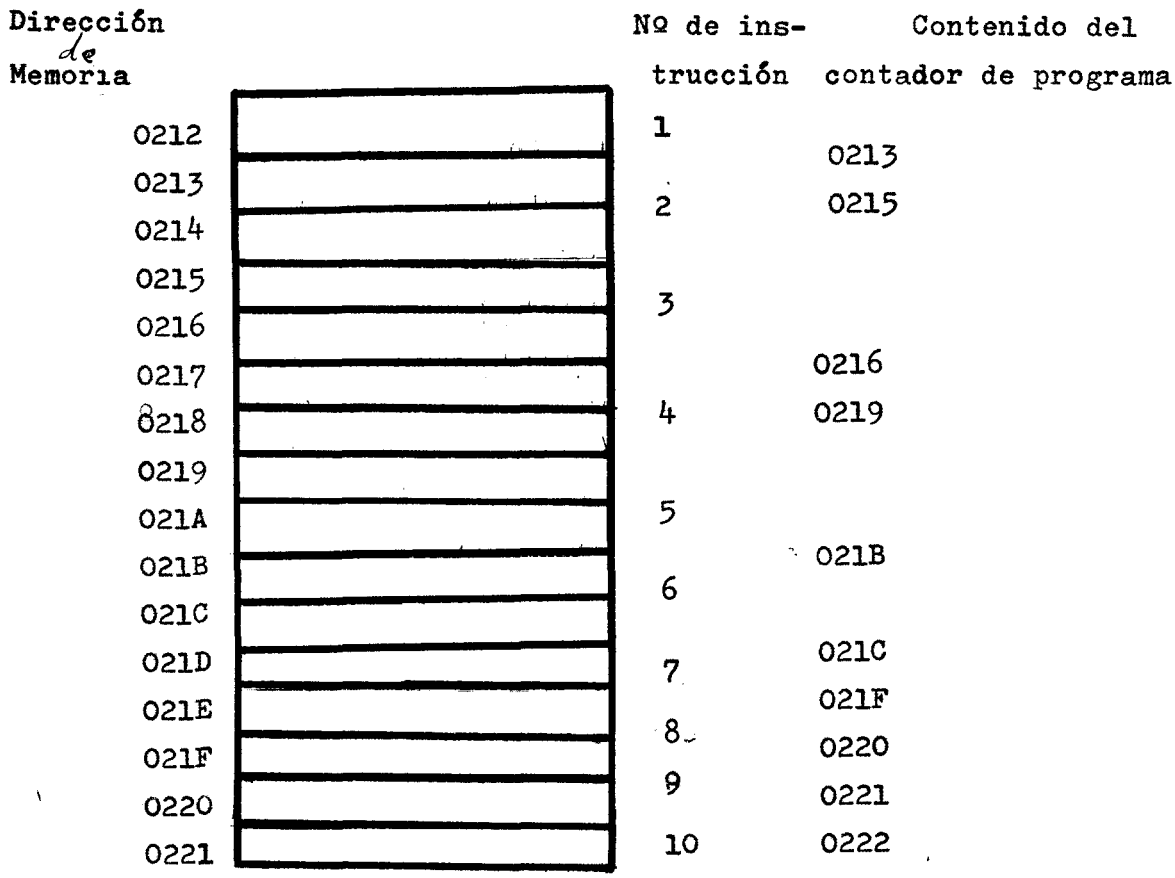
Un programa consiste en una secuencia de instrucciones . Cada instrucción da lugar a una operación básica, tal como el movimiento de un byte de datos, una operación lógica o aritmética, o bien un cambio en la secuencia de ejecución de las instrucciones.

Esta sucesión de instrucciones se representa mediante dígitos hexadecimales.

En el momentos en que comienza a ejecutarse una instrucción, el contador de programa avanza hasta la próxima instrucción a ejecutar. El programa ejecuta secuencialmente , salvo cuando tiene lugar una instrucción de llamada, sálto o retorno, en las que el contador de programa, se coloca en una posición determinada. Cuando esto ocurre, el programa continúa ejecutandose a partir de esta dirección de memoria.

Examinando individualmente el contenido de un byte de memoria, es posible determinar si se trata del código de una instrucción o bien de un dato hexadecimal. Por ejemplo, el código 1FH define la instrucción RAR. por lo que el hecho de encontrar este valor en byte de memoria puede suponer que se trata de esta instrucción, o bien un dato de valor 1FH. La forma en que se determina si se trata de datos o instrucciones se realiza sencillamente como sigue:

Todo programa tiene una dirección de memoria de inicio, donde está almacenada la primera instrucción a ejecutar. Justo antes de esto tenga lugar, el contador de programa, automáticamente, avanzada a la próxima dirección a ejecutar procedimiento que se repite para todas las instruciones del programa. Las instrucciones del microprocesador 8085 pueden requerir 1, 2, ó 3 bytes para codificarlas en su totalidad. En cada caso, el contador de programa avanza automáticamente hasta la posición de inicio de la próxima instrucción, tal como se ilustra a continuación de la figura 1. 1.
Para evitar errores, el programador debe asegurarse de que no hay ningún byte de datos a continuación de una instrucción cuando se debe realizar alguna otra a continuación.



Fgu. 1. 1.- Avance automático del contador de programa a medida que se van ejecutando las instrucciones.

Por ejem. , observando la figi. 1. 1. vemos que hay una instrucción en el byte 021FH que se ejecutará una vez concluida la instrucción 7.

Si 021FH contuviera un byte de datos, el programa no se ejecutará correctamente. Por todo ello, al escribir un programa debe tenerse en cuenta no almacenar datos en entre instrucciónes adyacentes que han de ser ejecutadas consecutivamente.

Asi mismo en el caso de las instrucciones que produce un salto hacia cualquier lugar de la memoria (salto, llamada, retorno,) hay que tener en cuenta que la dirección especifica debe ser de la otra instrucción, pues, de la misma manera que anteriormente, si contuviera datos, no se realiza el programa correctamente. Por ejem. y haciendo de nuevo referencia a la fig. 1.1. supongamos que la instrucción 4 es la de un salto a la posición 021FH, y la

instrucciones 5, 6, 7, se reemplazan por datos: una vez terminada la instrucción 4, el programa se realizaría correctamente. Ahora bien, si por error la instrucción 4 define un salto a la posición 021EH daría lugar a un error pues este byte ahora contiene datos. Asi mismo, aunque la instrucciones 5, 6, y 7 no hubieran sido sustituidas por datos, este salto provocaría un error, puesto que la posición 021EH no contiene el primer byte de una instrucción, sino el segundo.

DIRECCIONAMIENTO DE MEMORIA

Esta operación puede tener lugar en varias formas, que se describen a continuación:

DIRECCIONAMIENTO DIRECTO

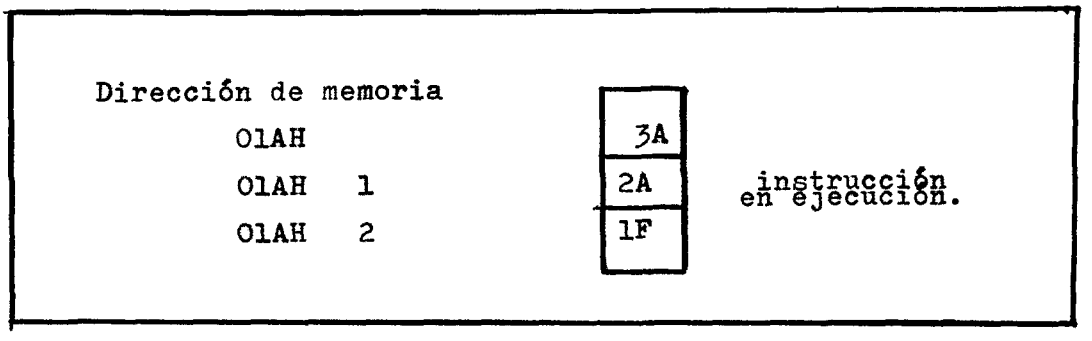
Mediante el direccionamiento directo, una instrucción proporciona una dirección de memoria.

LA INSTRUCCION

"Cargar el contenido de la dirección de memoria 1F2AH en el acumulador".

es un ejemplo de instrucción que utiliza direccionamiento directo, siendo esta dirección 1F2AH.

Esta instrucción aparecería en la memoria como:



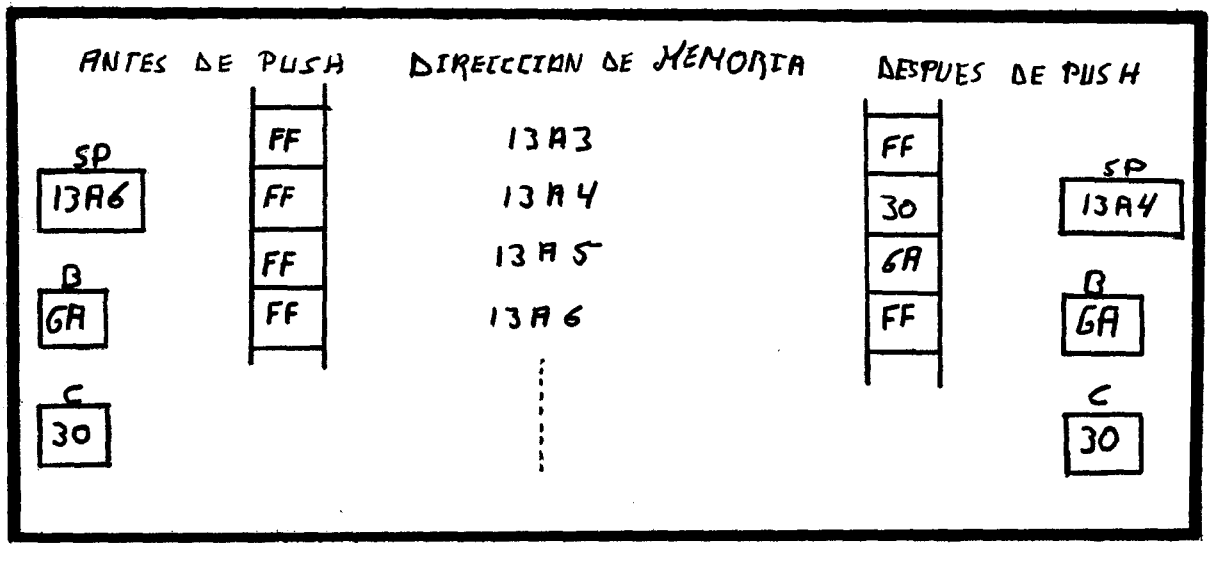
Esta instrucción ocupa tres bytes de memoria, de los que el segundo y tercero guardan la dirección.

(1) Los 8 bits más significativos de la palabra de datos se almacenan en la posición de memoria inmediatamente inferior a la definida por el puntero de stack.

(2) Los 8 bits menos significativos se almacenan en la posición de memoria 2 unidades inferior a la que define el puntero del stack.

(3) El puntero del stack, automáticamente, se decrementa dos unidades.

Por ejem. supongamos que el puntero de stack contiene la dirección de memoria 13a6H, el registro B contiene (HA,y el registro C, 30H. Cuando se realiza la operación de colocar en el stack el contenido del par de registros B, tiene lugar lo siguiente:



DIRECCIONAMIENTO CON UN PAR DE REGISTRO

Una dirección de memoria puede especificarse mediante un par de registros, usándose para la mayoría de instrucciones del 8085 los registros H y L. El registro H contiene los 8 bits más significativos de la dirección a la que se hace referencia, mientras que los restantes 8 bits están contenidos en el registro L.

Por ejem., en el caso de la instrucción de un byte que se utilice para cargar el acumulador con el contenido de la posición de memoria 2106H, el registro H debería contener 21H , mientras que el registro L contendría 06H. Existen además dos instrucciones que usan los registros B y C por una parte, y D y E por otra, como dirección de memoria.

En este caso, el primero de los dos registros contiene los 8 bits más significativos, y el segundo registro los restantes 8 bits. Estas instrucciones, que se describen posteriormente se denominan STAX y LDAX.

DIRECCIONAMIENTO MEDIANTE EL PUNTERO DE PILA

Otra forma de definir una dirección de memoria es a través del registro de 16 bits del puntero de stack. Existen únicamente dos instrucciones de stack que puedan ejecutarse: colocar datos en un stack, denominar PUSH, y sacar datos desde un stack, denominada POP.

Hay que tener en cuenta que para realizar la operación de PUSH es necesario que el stack esté en una zona de memoria de lectura/escritura.

OPERACION DE PONER DATOS EN LA PILA (stack)

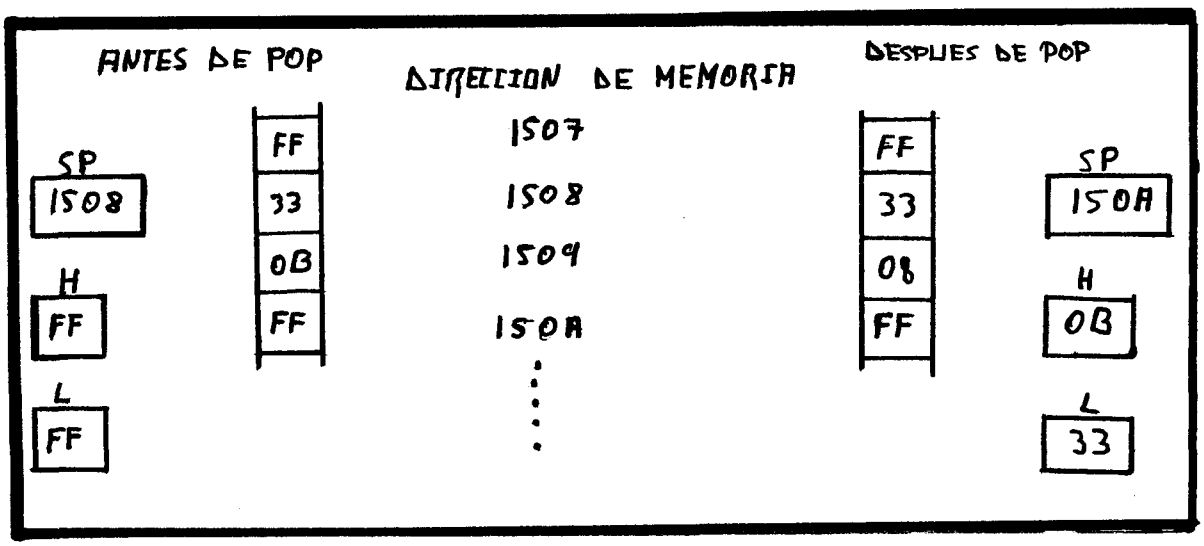
Durante una operación de este tipo, se transfieren a una zona de la memoria (denominada stack) 16 bits de datos contenidos en un par de registros o los 16 bits del programa. Las direcciones de esta zona de memoria se determinan usando el puntero de stack como sigue:

(1) El segundo registro del par de los 8 bits menos significativos del registro del contador de programa, se cargan con el contenido de la posición de memoria cuya dirección guarda el puntero de stack.

(2) El primer registro del par o los 8 bits más significativos del contador de programa, se cargan con el contenido de la posición de la memoria inmediatamente superior a la del puntero de stack.

(3) El puntero de stack, automaticamente, se incrementa en dos unidades.

Por ejemplo, supongamos que el puntero de stack contiene la dirección 1508h, la posición de memoria 1508H contiene 33H, y la posición de memoria 1509H contiene OBH . Al sacar del stack la dirección y guardarla en el par de registros H, se sucede la siguiente operación.



Para cargar el puntero del stack con cualquier valor deseado, se usa la instrucción LXI, que se describe más adelante.

El programador a de tener en cuenta que debe inicializar el puntero de stack antes de realizar una operación con el mismo, pues de no hacerlo así, los resultados obtenido serían erroneos.

DIRECCIONAMIENTO IMEDIATO

Se denomina instrucción inmediata a aquella que contiene datos, como por ejem.:

"Cargar el acumulador con el valor 2AH"

que estaría representada en la memoria de la forma siguiente:

Memoria

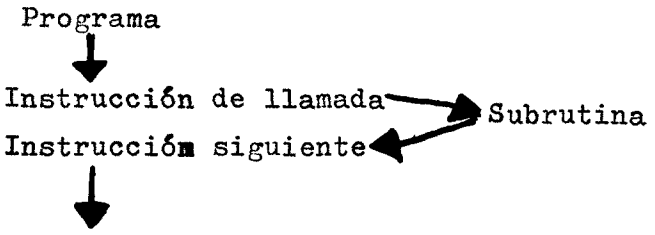
3E Cargar el acumulador con un valor inmediato.

2A Valor a cargar en el acumulador.

SUBRUTINAS Y EL USO DE LA PILA PARA SU DIRECCIONAMIENTO

Antes de ver el proposito y efectividad del stack, es necesario conocer en primer lugar el concepto de subbrutina.

Imaginemos una operación que se utilice muy frecuentemente, como por ejem. una multiplicación. El 8085 tiene ciertas instrucciones de operaciones arítmeticas básica, tales como sumar los bytes de datos, pero cuando se trata de realizar una operación más compleja, como en este caso, se necesitan una serie de instrucciones en secuencia. Por otro lado, es posible que esta operación se realice varias veces durante un programa, con utilización de gran cantidad de memoria.



El nivel de llamadas a subrutinas viene determinado por el tamaño de la zona de la memoria destinada al Stack. una subrutina puede llamar a otra, ésta a otra más y así sucesivamente. Si el tamaño del stack es lo suficientemente grande para guardar todas las direcciones de retorno, las operaciones se realizarán adecuadamente. Si por lo contrario, no se es capaz de contenerlas, se perderán las últimas direcciones, y el programa no funcionará en la forma adecuada.

BITS DE CONDICION (FLAGS)

El microprocesador 8085 dispone de 5 bits de condición que reflejan los resultados de operaciones con datos, la totalidad de los cuales, excepto uno (bit de arrastre auxiliar) puede ser comprobados mediantes determinadas instrucciones.

Posteriormente se verá en que forma se ven afectadas estos bits por la ejecución de las instrucciones, y también en que forman algunas instrucciones están condicionadas en su ejecución, al estado de los mismos.

Hay que tener en cuenta que posteriormente, al hablar de "activar" uno de los bits, nos estamos refiriendo a que su valor es "1", en tanto que el "borrar", se pone a "0".

BIT DE ARRASTRE (carry)

Las operaciones que afectan al bit de arrastre son las sumas, resta, rotación y operaciones lógicas. Por ejemplo, al sumar dos registros de un byte, puede ocurir que

el bit de mayor orden de lugar a un arrastre.

Número de bit	7	6	5	4	3	2	1	0
AE	0	0	1	0	1	1	1	0
74	0	1	1	1	0	1	0	0
122	0	0	1	0	0	0	1	0

Hay arrastre = bit de arrastre = 1

Esta operación activará el bit de arrastre que una operación similar en la que el bit de mayor orden no da lugar a un arrastre lo borrará.

BIT DE ARRASTRE AUXILIAR (auxiliary carry)

El bit de arrastre auxiliar indica cuando hay arrastre del bit. Su estado no puede comprobarse mediante ninguna instrucción, y únicamente se utiliza para la ejecución de una instrucción DAA. En la siguiente operación de suma, se borra el bit de arrastre y se activa el bit de arrastre auxiliar.

Número de bit	7	6	5	4	3	2	1	0
2E =	0	0	1	0	1	1	1	0
74 =	0	1	1	1	0	1	0	0
A2	1	0	1	0	0	0	1	0

Arrastre = 0 arrastre auxiliar.

El estado del bit de arraste auxiliar se ve afectado por las instrucciones de suma, restas, incremento y decremento y comparación.

BIT DE SIGNO

Tal como se describe posteriormente, al hablar de la Representación en Completo de Dos, es posible tratar un nº de una gama entre -128 a 127 en byte de datos. En este caso, por convenio el bit mayor de peso representa el signo del nº.

Cuando el bit está a 1. el número está en la gama de -128 a -1. Si el bit de 7 está a 0 y 127.
Al finalizar ciertas instrucciones, el bit de signo toma el mismo estado que el bit más significativo del resultado.

BIT DE CERO

Este bit se activa cuando el resultado a que da lugar la ejecución de la instrucción es cero, y se pone a 0 cuando el resultado no lo es.

En la siguiente operación hay arrastre, pero el resultado es caro, activando el bit de cero:

Número de bit	7	6	5	4	3	2	1	0
	1	0	1	0	0	1	1	1
	0	1	0	1	1	0	0	1
1	0	0	0	0	0	0	0	0

Arrastre del bit 7

Resultado cero

Bit de cero = 1

BIT PARIDAD

Despues de determinadas operaciones se comprueba la paridad de un byte, contando en nº de bits cuyo valor es 1 y si el total es impar se dice que tiene paridad impar, mientras que si el total es par, se habla de paridad par.

El bit de paridad se activa cuando hay paridad par, y se pone a cero cuando la paridad es impar.

MICROPROCESADORES 8085

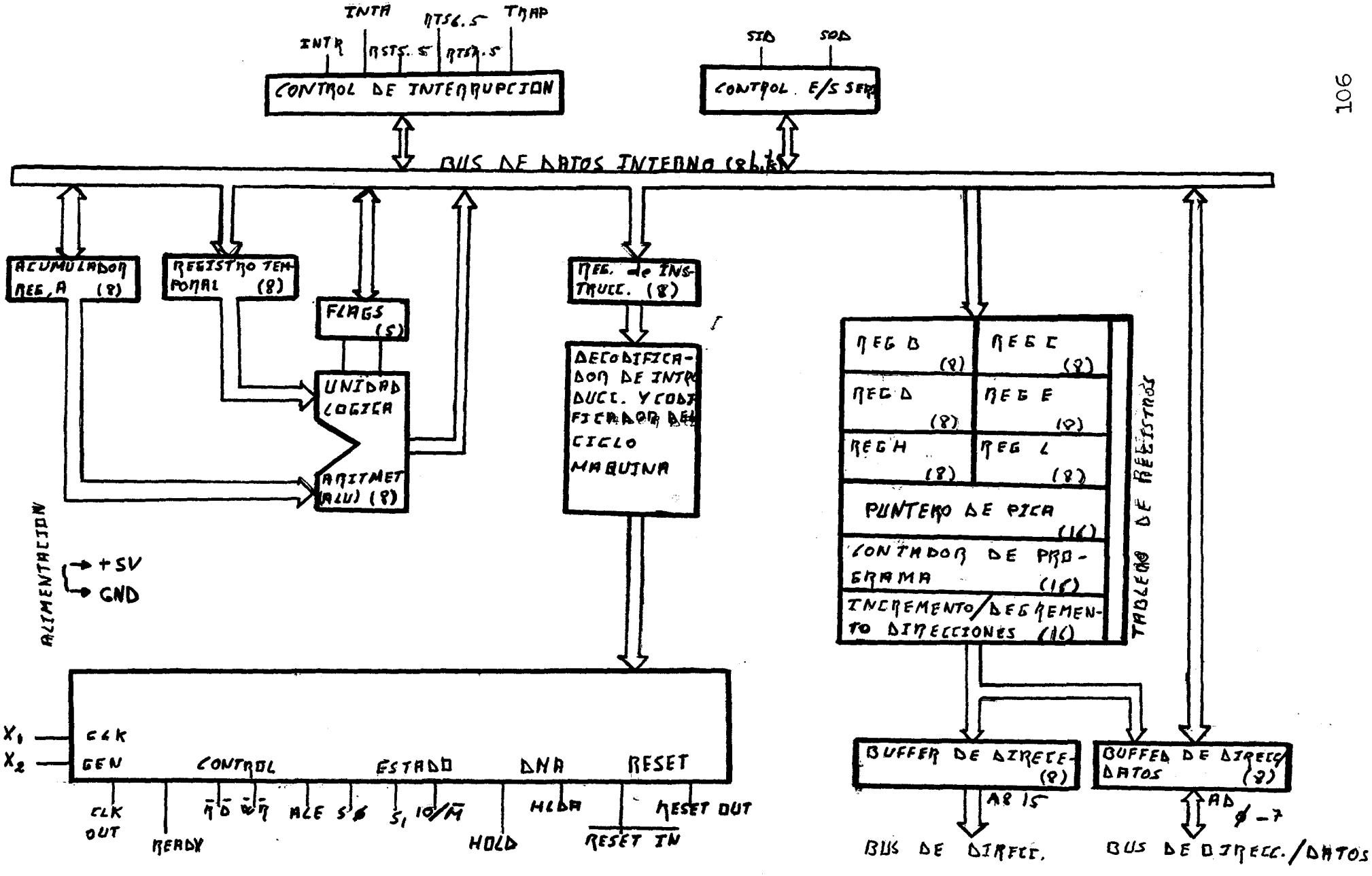
Generalidades:

El 8085 del Intel es un microprocesador de 8 bit Sus principales características son las siguientes:

- Una sola alimentación (+5V).
- Sofware totalmente compatible con el microprocesador 8080.
- El ciclo de la instrucción es de 1.3 microsegundos.
- Tiene el reloj dentro de la pastilla con cristal de cuarzo a red R-C extremas).
- Incluye tambien controlador de sistema.
- 4 vectores de interrupción (uno de ellos no enmascarable).
- Puertas de entrada/salida serie.
- Operaciones aritméticas binarias.
- Capacidad de direccionamiento directo hasta 64 K- Bytes de memoria.
- Su alto nivel tiene integración permite realizar un sistema mínimo de tres circuitos integrados.

.- 8085 (CPU)
.- 8155 (RAM)
.- 8355/8755 (ROM/PROM)

--Se utiliza un bus de datos multiplexado. Es decir que los 16 bits de dirección están formados por los 8 bits del bus de direcciones y los 8 bits del bus de datos.



ARQUITECTURA DE 8085

En el 8085 están contenidas las funciones de: generación de ciclos de reloj, control del bus del sistema, selección de paridades de interrupción, además de la ejecución de las instrucciones ver fig. 1. 1. . El 8085 transfiere datos a través de un bus triestado bidireccional de 8 bit (AD_8 -AD_7) multiplexado. A través de este bus de datos envía también los 8 bits de menor peso de direccionamiento. Otras 8 líneas adicionales (A_8 - A_{15}) permiten una capacidad de direccionamiento de 16 bits, es decir 64K de memoria pueden ser accedidos directamente por la CPU.

La CPU 8085 genera señales de control que se pueden utilizar para seleccionar determinados elementos y funciones exteriores, con el fin de efectuar operaciones de lectura/escritura, y tambien para selecionar direcciones de memoria o puertas I/O.

El 8085 puede direccionar hasta 256 posiciones I/O. estas direcciones tienen el mismo valor numérico (OO a FFH) que las primeras 256 posiciones de memoria, la distinción se hace por medio de la salida IO/M de la CPU. Se pueden tambien direccionar las puertas I/O como direcciones normales.

REGISTROS

El 8085 tiene internadamente registros de 8 a 16 bits. Tiene registros de 8 bits accesibles al programador. De datos 8 registros, 6 pueden ser utilizados como registros simples o dobles (registros de 8 ó 16 bits). Además de estos tres registros pares, hay otros dos registros de 16 bits.

Los registros 8085 son:

EL ACUMULADOR

(ACC o registro A) es el centro de todas las instrucciones del acumulador (tabla 4.1), los cuales realizan o-

operaciones aritméticas, lógicas, carga entrada/salida. Es un registro de 8 bits.

EL CONTADOR DE PROGRAMA

Siempre indica la posición de memoria donde se encuentran la próxima instrucción que debe ser ejecutada. Es un registro de 16 bits.

REGISTROS DE USO GENERAL

(b, C, D, E, H, L) pueden ser utilizados como registros de 8 o 16 bits (B-C, D-E, H-L).

El registro de HL funciona como un indicador de datos (data pointer) para referenciar posiciones de memoria que son la fuente o el destino en algunas instrucciones. Unas pocas instrucciones permiten utilizar los registros B-C y D-E para direccionamiento directo.

EL PUNTO DE PILA

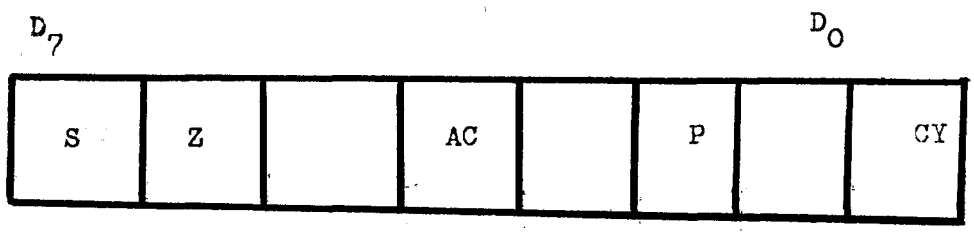
Es un puntero de datos especial que señala la parte superior del stack (pila). Es un registro de 16 bits indivisible.

EL REGISTRO DE FLAGS

Contiene 5 flags de 1 bit, cada uno de los cuales da alguna información sobre el estado del microprocesador y pueden también controlar el funcionamiento del microprocesador.

FLAGS

Los cinco flags del 8085 son:



EL FLAG DE ARRASTRE

El flag de arrastre o carry flag (CY) es puesto a 1 ó 0 por las operaciones aritméticas. Su estado se puede testar directamente desde un programa. Por ejemplo, la adición de dos números de 1 byte puede producir un resultado que no entra en un byte.

Hexadecimal	Binario
AEH	1 0 1 0 1 1 1 0
+ 74H	+0 1 1 1 0 1 0 0
122H	1 0 0 1 0 0 0 1 0
	bit elevada

El bit de llevada pone al 1 flag de llevada. Una operación de adición que produzca un desbordamiento (overflow) en el resultado en el (acumulador) pone al 1 flag de llevada.

EL FLAG DE ARRASTRE AUXILIAR (AC)

Indica el desbordamiento del tercer bit del acumulador de la misma forma que el flag de llevada (CY) indica desbordamiento del septimo bit del acumulador. Este flag se utiliza normalmente en la BCD (Binary Coded Decimal).

EL FLAG DE SIGNO (S)

Depende del bit de mayor peso del acumulador después de una operación aritmética lógica. Estas instrucciones utilizan el 7 bit de un dato para representar el signo de un número contenido en el acumulador. Esto permite la manipulación de números comprendidos entre -128 y +127.

FLAG DE CERO (Z)

Se pone a 1 si el resultado de ciertas instrucciones en cero. Si el resultado no es cero este flag se pone a Ø.

ejemplo

Hexagecimal	binario
A7H	1 0 1 0 0 1 1 1
+59H	+0 1 0 1 1 0 0 1
100H	1 0 0 0 0 0 0 0 0

Bit de llevada

Los 8 bit a cero ponen a 1 el flag de cero.

Incrementando y decrementando ciertos registros de la CPU el flag de cero se pondrá a 1 si el resultado es cero.

EL FLAG DE PARIDAD (P)

Se pone a 1 si en nº de bits del acumulador que están a 1 es par. Si no, se pone a Ø.

LA PILA (stack)

El puntero de la pila (stack pointer) contiene la dirección del último octeto cargado en la pila. Se puede inicializar el puntero de la pila a un cierto valor con la finalidad de utilizar cualquier porción de memoria RAM como pila.

El puntero se decrementa cada vez que un dato entra a la pila, y se incrementa cada vez que el dato sale de la pila.

Nótese que el puntero se incrementa o decrementa de dos en dos bytes, puesto que la información que se guarda en la pila ocupa 2 bytes.

UNIDAD LOGICA ARITMETICA (ALU)

Contiene el acumulador, el registro de flags y algunos registros temporales que son inaccesibles al programador. El ALU realiza operaciones lógicas, aritméticas y de rotación-.El resultado de dichas operaciones se deposita en el acumulador o en el bus de datos interno.

REGISTRO DE DECODIFICADOR DE INSTRUCCIONES

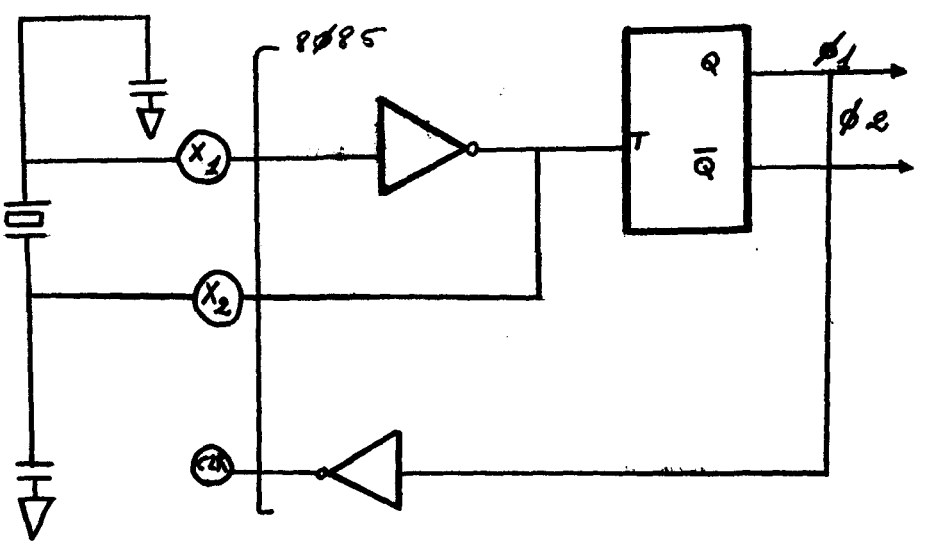
Ya hemos dichos que el primer octeto de la instrcción se transfiere desde el bus interno a un registro de 8 bits llamado registro de instrucciones. El contenido de este registro está a disposición del decodificador de instrucciones. La salida del decodificador controla los registros, e ALU y los buffers de datos y direcciones.

Las salidas del decodificador y el genedor interno del reloj generan las señales ciclo máquina y ciclo estado.

GENERADOR DE RELOJ INTERNO

El 8085 lleva un generador de reloj, de forma que solo se necesita un cristal de cuarzo exterior. Tambien acepta un reloj externo aplicado a la pata X_1.

Para el 8085A se utiliza un cristal de 6,25 MHz. o menos . forma que su frecuencia de resonancia sea el doble de la frecuencia de reloj interna deseada.
Para el 8085 - 2 se puede utilizar un cuarzo de hasta 10 MHz



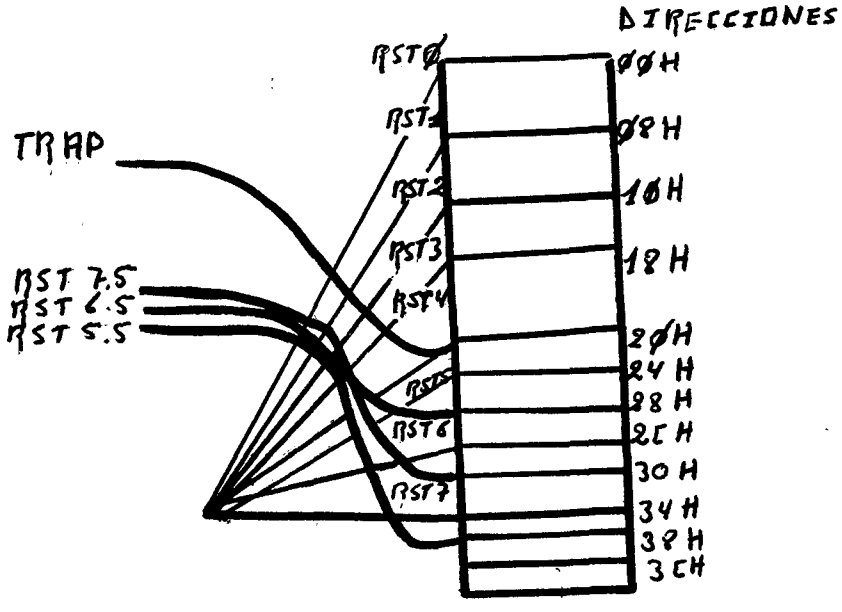
Las figuraciones del generador interno de reloj se muestran en figura 1.2.

El trigger de schmitt se utiliza como occilador o como condicionador de entrada dependiendo de si se utiliza un cuarto de reloj extremo.

La circuiteria de reloj genera dos señales de reloj interna ϕ_1 y θ_2, las cuales controlan el timing de 8085. Estas dos señales no son directamente disponibles desde el exterior. Sin embargo, la pata CLK es equivalente a la señal O_1 invertida . La frecuencia de la señal CLK es la mitad de la frecuencia del cristal y se puede utilizar como reloj para otros elementos del sistema.

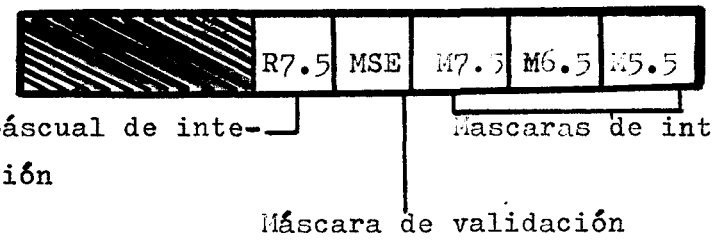
INTERRUPCIONES

En el 8085 hay previstas 5 interrupciones HARWARE, que pueden ser de tres tipos:.- INTR es una interrupción enmascarable es decir que puede ser validad o invalidada por algunas instrucciones Sofware, EI y DI. Esta interrupción hace que la CPU busque una instrucción RST_n sobre el bus de datos. Esta instrucción provocará un salto a una de las ocho posiciones de memoria fijadas



Las interrupciones harware RST 5.5, RST 6.5 y RST 7.5 se diferencian en que son enmascarables por medio de la instrucción SIM. Esta instrucción válida o invalida las instrucciones poniendo a 0 ó 1 bit máscaras basados sobre datos en el acumulador.

Contenido del acumulador antes de la ejecución de sim

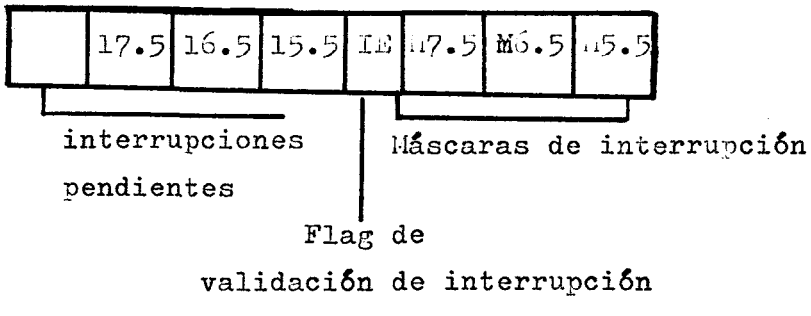


Se puede leer el estado de la máscara de interrupciones previamente establecida, por medio de una interrpción RIM. Su ejecución carga el acumulador la información siguiente:

- Estado actual de la máscara de interrupción, de RST 5.5, RST 6.5 y RST 7.5, -
- Estado actual del flag de interrupciones,
- Las interrupciones RST 5.5 , RST 6.5 y RST 7.5 pendiente.

Rim- lectura máscara de interrupción
(código de operación igual 20)
contenido del acumulador después de la ejecución RIM:

Contenido del acumulador después de la acumulación RIM



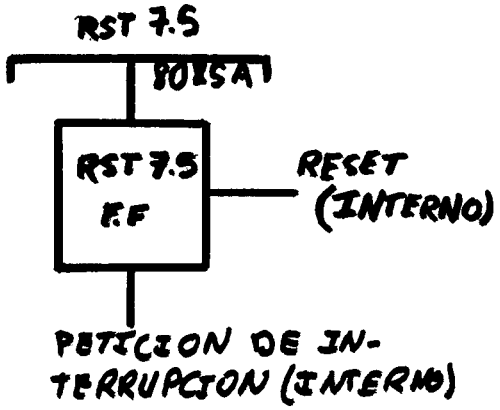
RST 5.5, RST6.5 y RST 7.5, se puede invalidar o validar(en grupo) con las instrucciones EI y DI.
NTR, RST 5.5, RST 6.5 son sencibles al nivel, es decir que serán reconocidas por el procesador cuando se mantienen a nivel alto.

Rst 7.5 es sensible al cambio, es decir, que un flip-flop interno registra la petición de una interrupción en el instante en que una subida aparece sobre la pata RST 7.5 . Esta pata no necesita ser mantenida a nivel alto, el flip - flop permanecerá a 1, hasta que se produzca una de las tres acciones siguientes :

.- El 8085 responde a la interrupción y envia una señal interna de reset al flip - flop RST 7.5.

.- El 8085 antes de responder a la interrupción RST 7.5 recibe la señal RESET IN desde un elemento externo. Esto activa tambien al reset interno.

.- El 8085 ejecuta una instrucción SIM con el bit 4 del acumulador a 1.



El tercer tipo de interrupción hardware es ser TRAP. Esta es una interrupción no enmascarable es decir, no depende de ninguna máscara ni instrucción de validación/invalidación

NOTAS

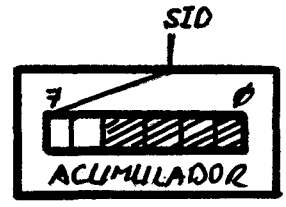
(1) En el caso de TRAP, RST 5.5 , RST 6.5 y RST 7.5, el contenido del contador del programa es guardado en la Pila antes del salto.

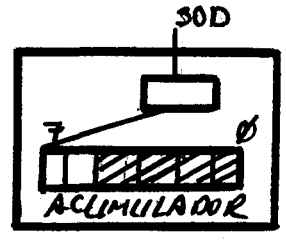
(2) Depende de la instrucción que se pone sobre el bus de datos cuando la interrupción es reconocida.

ENTRADA / SALIDA SERIE

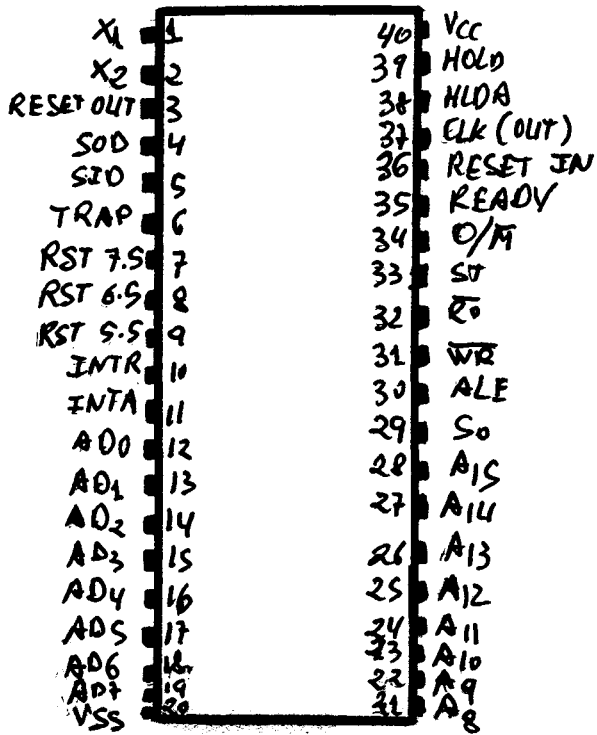
Cada vez que se ejecuta una instrucción RIM, el estado de la pata SID pasa al bit 7 del acumulador .

De la misma forma, la instrucción SIM saca el contenido del bit 7 del acumulador sobre la pata SOD, a través de un flip-flop interno. Además, pone a 1 el bit 6 del acumulador.



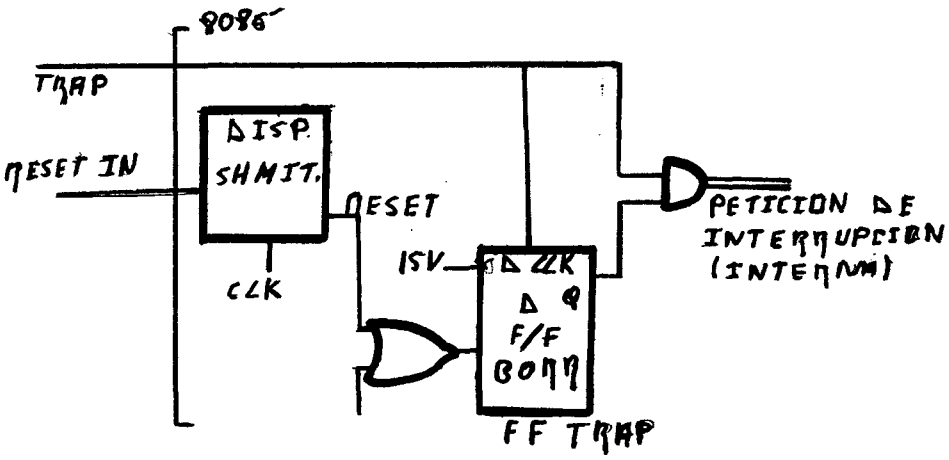


DEFINICION FUNCIONAL DE LAS PATAS DEL 8085



interrupciones.

Una subida sobre la pata trap produce la ejecución de la interrupción. Pero esta señal debe ser mantenida a nivel alto hasta que sea reconocida por el procesador.



El muestreo de todas las interrupciones ocurre durante el frente descendente del CLK , un ciclo antes del final de la intrucción en la cual se ha activado la entrada de interrupción.

Las funciones y prioridades de cada tipo de interrupción se muestran en la tabla siguiente

Nombres	Prioridad	Dirección del salto	Tipo de trigger
TRAP	1	24H	Flanco de subida y mantener alto este muestreo.
RST 7.5	2	3CH	Flanco de subida.
RST 6.5	3	34H	Nivel alto hasta el muestreo.
RST 5.5	4	2CH	Nivel alto " " "
INTR	5	(2)	Nivel alto hasta el muestreo-

$A_7 - A_{15}$ (PARENTESIS TRI- ESTADO)

Bus de direcciones . Está constituido por los 8 bits de mayor peso de direccionamiento, o los 8 bits de las direccionamiento, o los 8 bits de las direcciones de entrada salida.

Durante los modos HALT Y HOLD está en estado de alta impedancia o triestado.

$AD_0 - AD_7$ ENTRADA SALIDA TRI/ ESTADO)

Bus de datos/direcciones multiplexado. Constituyen los 8 bits de menor peso en el direccionamiento de la memoria o direcciones I/O aparecen sobre el bus durante el primer ciclo de reloj. Durante el segundo y tercer ciclo de reloj es el bus de datos,.
Durante los modos Hald-Hold está en tri-estado.

ALE (SALIDA)

Valida la dirección del bus. Ocurre durante el primer ciclo de reloj y válida la dirección para que pueda ser guardada en el lach interno de los perifèricos.

El frente de bajada de ALE se utiliza para asegurar que la información en el bus de direcciones es la adecuada(direcciones $A_o - A_{15}$). ALE se puede utilizar tambien para validad la información del estado.

SO-S1 (salida)

Espado del bus de datos. La codificación del estado del bus es:

S_0	S_1	
0	0	HALT
0	0	WRITE
1	0	PEAD
1	1	FETCH

S_1 se puede utilizar como un estado R/$\overline{W}$ avanzado.

RD/ (SALIDA TRI' ESTADO)

Indica que la CPU va ha efectuar una operación de lectura de la memoria o puerta y que el bus de datos está disponible para la transferencia de datos. Está en tri-estado durante hold y hald.

WR (SALIDAS 3-ESTADO)

Indica que el dato del bus de datos va ha ser escrito en la memoria o puerta selecionadas. El dato es enviado en el segundo frente de WR. Durante los modos hald y hold está en tri-estado.

READY (ENTRADA)

Si está a nivel alto durante un ciclo de lectura o escritura indica que la memoria o el periférico está listo para mandar o recibir datos. Si esta a nivel bajo, la CPU esperará a que se ponga a nivel alto para completar el ciclo de lectura o escritura.

HOLD (ENTRADA)

Indica que otro Master esta pidiendo el uso del bus de direcciones y datos . Cuando la CPU reciba la petición de hold, renuciará al uso de los buses tan pronto como complete en ciclo máquinas en ejecución. El proceso interno podrá continuar. El proceso solo podrá volver a utilizar los buses cuando la señal hold sea eliminada. Cuando el hold está reconocido, las lineas de direcciones, datos, RD, RW y IO / M están en 3-estado.

ALDA (SALIDA)

Reconocimiento de hold. Indica que la CPU ha recibido la petición de hold y que abandonará los buses los pone en tri - estado en el próximo ciclo de reloj.

HLDA pasa a nivel bajo cuando se quita la señal hold. La CPU se hace cargo de los buses medio ciclo de reloj después de que HLDA pasa a nivel 0.

INTR (ENTRADA)

Se utiliza como una interrupción para fines generales. Se muestrea durante el ciclo siguiente al último ciclo de reloj de la instrucción. Si está activada, el Contador de Programa nº se incrementará y se enviará una señal $\overline{INTA}$ de reconocimiento de interrupción. Durante este ciclo se puede insertar una instrucción CALL o RST_n para saltar a la subrutina de intervención.

INTR es validada e invalidad por software,. Tambien es invalidada por RESET e inmediatamente después que la interrupción ha sido aceptada.

INTA (SALIDA)

Reconocimiento de interrupción se utiliza en lugar de RD durante el ciclo de instrucción después de que INTR ha sido aceptada. Se puede utilizar para activar el 8259 (chip de interrupciones) o algunas otras puertas de interrupción.

RST 5.5, RST6,5, RST 7.5 (ENTRADAS)

Estas tres entradas tienen el mismo timing que INTR salvo la excepción de que estas hacen que un restart (salto a una dirección fija) interno sea incertado automáticamente.

RST 7.5	Mayor prioridad
RST 6.5	
RST 5.5	Menor prioridad

Estas interrupciones tienen mayor prioridad que INTR.

TRAP (ENTRADA)

Es una interrupción no enmascarable. Es muestreado al mismo tiempo que INTR. No es afectado por ninguna máscara o validación de interrupciones. Tienen la prioridad máxima

RESET IN (ENTRADA)

Pon el contador de programa a cero y tambien los flip-flops de validación de interrupción HLDA. Ningún otro flag o registro ecepto registro de instrucciones es aceptado. La CPU se mantiene en las condiciones citadas, mientras siga aplicada la señal.

RESET OUT (SALIDA)

Indica que sobre la CPU está actuando en reset. Se puede utilizar como el reset del sistema. Esta señal está sincronizada con el reloj del procesador.

X_1 X_2 (ENTRADAS)

Conecciones por el crital de cuafza o la red R-C para poner en marcha el reloj del generador interno/ X_1 se puede utilizar como la entrada de un generador de reloj externo, en lugar de un cristal. La frecuencia de entrada es dividida por dos para producir la frecuencia interna de funcionamiento.

CLK (SALIDA)

Salidad de reloj para utilizala de reloj del sistema, cuando un cristal de cuarzo o red R-C se utiliza como entrada a la CPU. El periodo CLK es el doble del periodo X_1 , X_2.

IO/M (SALIDA)

Indica si la lectura/escritura es a una memoria o a una puerta. En tri- estado durante hald/hold.

SID (ENTRADA)

Linea de entrada de datos en serie. El dato de esta linea se carga en el bit 7 del acumulador siempre que se ejecuta una instrucción.

SOD (SALIDA)

Linea de salida de datos en serie. Las salida sod espuesta a uno ó 0 según se especifica en la instrucción SIM.

Vcc

Alimentación más cinco V.

Vss

Referencia de masa.

BIBLIOGRAFIA:

.- A. GOLDEROS, R. MARTINEZ, J.R. NOMBELA, M. PARDO, J. SANTOS, E. MUÑOZ; COMUNICACION HOMBRE MAQUINA POR VOZ.

.- B. P. LATHI;,SISTEMAS DE COMUNICACION, edita LIMUSA.

.- ELIAS MUÑOS MERINO, CIRCUITOS INTEGRADOS.

.- INTRODUCCION AL ESTUDIO DEL MICROPROCESADOR; MANUAL DE USO DEL MICROPROCESADOR 2.000.

.- JACOB MILLMAN Y CHRISTOS C. HALKIAS; ELECTRONICA INTEGRADA.

.- JOHN P. CATER, ELECTRONICALLY SPEAKING COMPUTER SPECH GENERATION.

.- L. R. RABINER Y R. W. SCHFER; DIGITAL PROCESSING OF SPEECH SIGNALS.

.- MALEN RUIZ, ELVIRA Y GONZALO MORENO, SISTEMAS DE RECONOCIMIENTO DE SINTESIS DE VOZ (ORDENADORES QUE HABLAN Y ESCUCHAN).

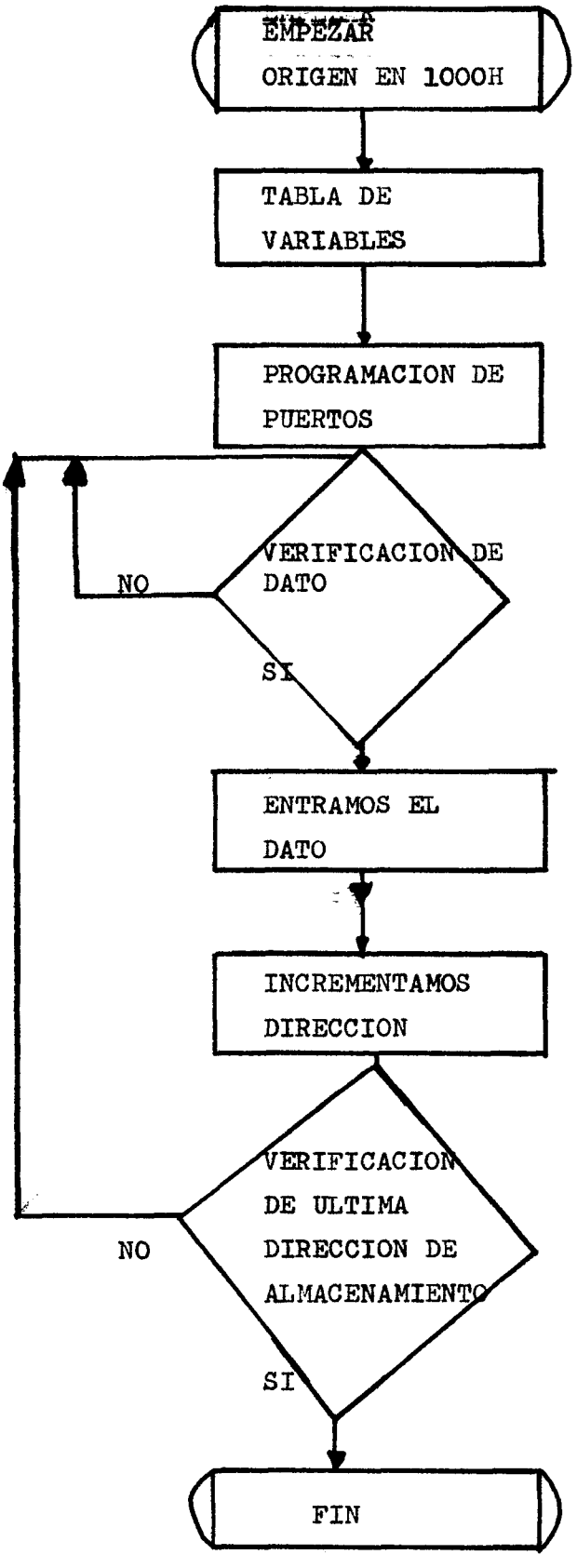
.- PEDRO PERDOMO DELGADO, TECNICA DE CODIFICACION MIC (Modulacion por impulsos codificados), ESCUELA POLITECNICA DE LAS PALMAS.

.- STEVE CIARCIA, ARTICULO PERIODISTICO EN REVISTA BYTE; TALK TO ME.

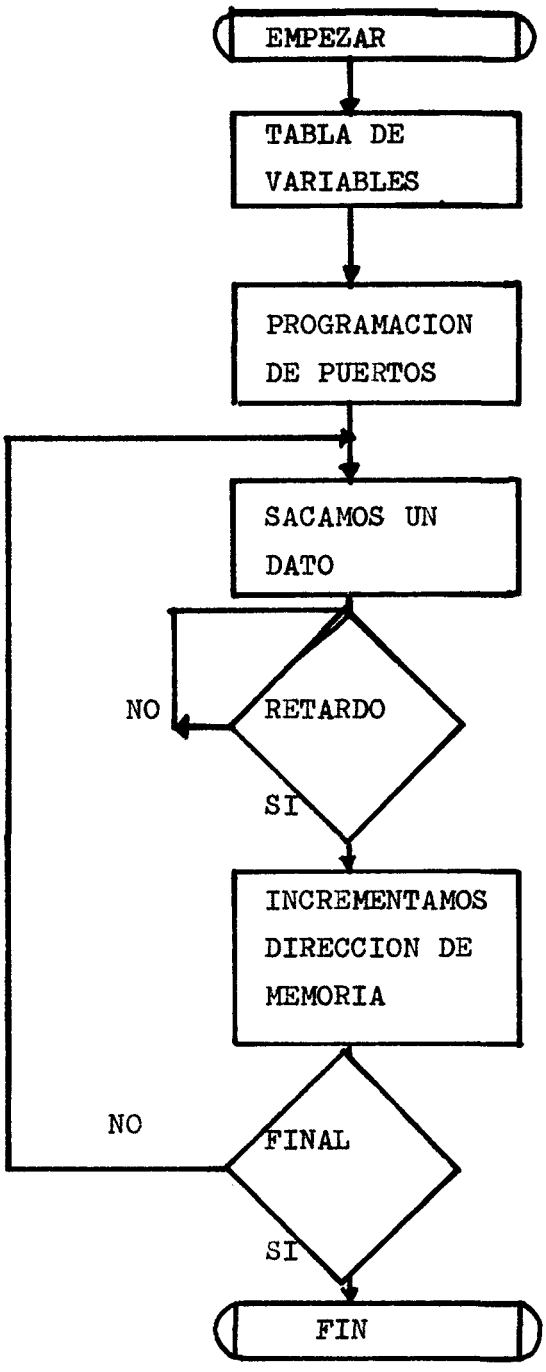
.- STEPHEN E. LEVINSON Y MARK Y. LIBERMAN, RECONOCIMIENTO DEL HABLA POR MEDIO DE ORDENADORES.

- APENDICE -

Organigrama del programa HABLAR para introducir las muestras en el ordenador.



Organigrama correspondiente al programa ESCUCH para sacar las muestras del ordenador.



```
ASM80 HABLAR MOD85 XREF


ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0          MODULE   PAGE    1


  LOC  OBJ           LINE         SOURCE STATEMENT

                        1 ;&       ROGRAMA PARA HABLARLE          &
                        2 ;&   -   AL ORDENADOR                   &
                        3 ;&       POR  PEDRO PERDOMO             &
                        4
  3000                  5          START   EQU     3000H
  6000                  6          FIN     EQU     6000H
  3800                  7          IPORT   EQU,    3800H
  3802                  8          CPORT   EQU     3802H
                        9
  1000                 10          ORG     1000H
                       11
  1000 3E91            12          MVI     A,91H
  1002 D20338          13          STA     3803H
  1005 210030          14          LXI     H,START
                       15
  1008 DA0238          16      VER:LDA     CPORT
  100B FE01            17          CPI     01H
  100D CA0810          18          JZ      VER
                       19
  1010 3A0038          20      INP:LDA     IPORT
  1013 77              21          MOV     M,A
  1014 23              22          INX     H
  1015 7C              23          MOV     A,H
  1016 FE60            24          CPI     60H
  1018 CA0810          25          JZ      VER
                       26
                       27          END


PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS
CPORT  A 3802    FIN    A 6000    INP    A 1010    IPORT  A 3800    START  A 3

ASSEMBLY COMPLETE,  NO ERRORS
```

```
ASM80 ESCUCH MOD85


ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0          MODULE    PAGE     1


  LOC  OBJ            LINE          SOURCE STATEMENT

                         1 ;        &        PROGRAMA PARA QUE EL                &
                         2 ;        &        ORDENADOR                           &
                         3 ;        &        POR PEDRO PERDOMO                   &
                         4
  3000                   5          INIC     EQU      3000H
  8000                   6          FIN      EQU      8000H
  3802                   7          CPORT    EQU      3802H
  3801                   8          OPORT    EQU      3801H
  000E                   9          RETAR    EQU      0EH
                        10
  1000                  11          ORG      1000H
                        12
  1000 3E71             13          MVI      A,71H
  1002 320338           14          STA      3803H
  1005 210030           15          LXI      H,INIC
                        16
  1008 7E               17     VER:MOV      A,M
  1009 320138           18          STA      OPORT
  100C CD1710           19          CALL     DELY
  100F 23               20          INX      H
  1010 7C               21          MOV      A,H
  1011 FE80             22          CPI      80H
  1013 C20810           23          JZ       VER
  1016 76               24          HLT
                        25
                        26
  1017 160E             27    DELY:MVI      D,RETAR
  1019 15               28     DEC:DCR      D
  101A C2191D           29          JNZ      DEC
  101D C9               30          RET
                        31          END


PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS
CPORT , A 3802    DEC    A 1019    DELY   A 1017    FIN    A 8000    INIC   A
VER    A 1008

ASSEMBLY COMPLETE,  NO ERRORS
```

ISIS-II 8080/8085 MACRO ASSEMBLER, V4.0 MODULE PAGE 1

```
LOC  OBJ            LINE     SOURCE STATEMENT

                       1 ;       &     PROGRAMA PARA HABLARLE AL        &
                       2 ;       &     ORDENADOR Y AHORRAR              &
                       3 ;       &     MEMORIA                          &
                       4 ;       &     POR PEDRO PERDOMO                &
                       5
3802                   6         CPORT   EQU     3802H
3800                   7         IPORT   EQU     3800H
3803                   8         START   EQU     3803H
8003                   9         PTR     EQU     8003H
3000                  10         ULT     EQU     3000H
                      11
1000                  12         ORG     1000H
                      13
1000 B7               14         ORA     A
1001 47               15         MOV     B,A
1002 320110           16         STA     1001H
1005 320310           17         STA     1003H
1008 210030           18         LXI     H,START
                      19
100B DA0D10           20         JC      VER
100E FE01             21         CPI     01H
1010 CA0810           22         JZ      VER
                      23
1013 3A0038           24         LDA     IPORT
1016 FE80             25         CPI     80H
1018 CA1C10           26         JZ      CARG
101B 04               27         INR     B
101C 57               28    CARG:MOV     D,A
101D 3A0110           29         LDA     1001H
1020 3C               30         INR     A
1021 320110           31         STA     1001H
1024 3A0310           32         LDA     1003H
1027 86               33         ADD     M
1028 320310           34         STA     1003H
102B 7A               35         MOV     A,D
102C FE03             36         CPI     03H
102E CA3B10           37         JZ      VER
1031 B7               38         ORA     A
1032 47               39         MOV     B,A
1033 3A0010           40         LDA     1000H
1036 FE00             41         CPI     00H
1038 CA4D10           42         JZ      AQUI
103B 3A0110           43         LDA     1001H
103E 320010           44         STA     1000H
1041 3A0310           45         LDA     1003H
1044 320210           46         STA     1002H
1047 220030           47         SHLD    ULT
104A C30B10           48         JMP     VER
                      49
104D 3A0110           50    AQUI:LDA     1001H
1050 210010           51         LXI     H,1000H
1053 86               52         ADD     M
1054 CA6410           53         JZ      ALLI
1057 3A0210           54         LDA     1002H
```

```
1044 C30B10          48          JMP     VER
                     49
104D 3A0110          50    AQUI: LDA     1001H
1050 210010          51          LXI     H,1000H
1053 86              52          ADD     M
1054 CA6410          53          JZ      ALLI
1057 3A0210          54          LDA     1002H
```



```
LOC  OBJ             LINE        SOURCE STATEMENT

105A 210310          55          LXI     H,1003H
105D 86              56          ADD     M
105E CA6410          57          JZ      ALLI
1061 C37F10          58          JMP     PPD
                     59
1064 3A0110          60    ALLI: LDA     1001H
1067 320010          61          STA     1000H
106A B7              62          ORA     A
106B 320110          63          STA     1001H
106E 3A0310          64          LDA     1003H
1071 320210          65          STA     1002H
1074 B7              66          ORA     A
1075 320310          67          STA     1003H
1078 7E              68          MOV     A,M
1079 320030          69          STA     ULT
107C C30B10          70          JMP     VER
                     71
107F 3A0410          72    PPD:  LDA     1004H
1082 3C              73          INR     A
1083 320410          74          STA     1004H
                     75
1086 3E00            76          MVI     A,00H
1088 320030          77          STA     ULT
108B 110030          78          LXI     D,ULT
108E 13              79          INX     D
108F 3A0410          80          LDA     1004H
1092 12              81          STAX    D
1093 13              82          INX     D
1094 1A              83          LDAX    D
1095 77              84          MOV     M,A
1096 3A0080          85          LDA     FIN
1099 86              86          ADD     M
109A C30B10          87          JZ      VER
                     88          END
```

PUBLIC SYMBOLS

EXTERNAL SYMBOLS

USER SYMBOLS

```
ALLI   A 1064   AQUI   A 104D   CARG   A 101C   CPORT  A 3802   FIN    A
START  A 3000   ULT    A 3000   VER    A 1005
```

ASSEMBLY COMPLETE, NO ERRORS

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

Escala	FUENTE DE ALIMENTACION		Plano nº 1
Dibujado	P.P.D.	Fecha 1-9-85	Sustituye a:
Proyec.	LAS PALMAS DE G.C.		Sustituido por:
Compro.	VOZEX PPD		Tolerancias Generales
Nº	Denominación	Cant. Material y dimensiones	Mecanizado

PRESUPUESTO:

No quisiera hacer un presupuesto muy cerrado a los precios del día de hoy, sino que quiero que sea un poco abierto y se rija por una media nacional, para que pueda ser más apr aproximado.

El sistema mínimo que yo utilicé fue el SDK 85 o La Tarjeta ALECOP, ésta última valdrá sobre las 50.000 pts.

LOs conversores pueden salir por las 1.000 a 1.500 pts.

Los diversos componentes: resistencias, condensadores,potenciometros ... alrededor de las 2.000pts.

No pongo coste del Software por la variedad de programas que se pueden emplear según su mayor o menor complejidad de hay la variación en su posible coste.

Un coste importante va ha ser la memoria empleada, cuanto mayor número de palabras queramos tener almacenadas en memoria mayor número de memoria tendremos que ponerle al prototipo, y por tanto el gasto será mucho mayor.
pongamos para 1 minuto de voz almacenada,serán suficientes 60 Kbytes, pongamos 5.000 a 7.000pts.

El prototipo nos va saliendo por un total aproximado de 60.000 pts.

National Semiconductor

A to D, D to A

ADC0808, ADC0809 8-Bit μP Compatible A/D Converters With 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

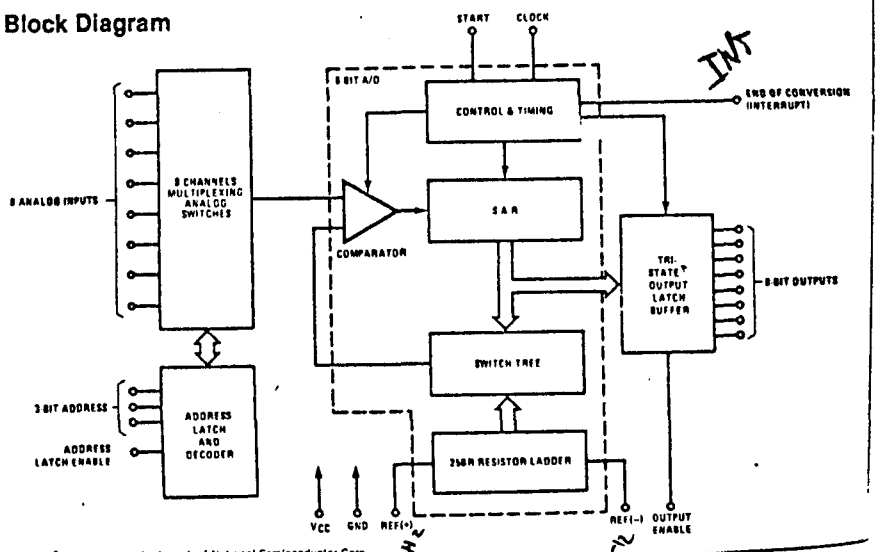
The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.

Features

- Resolution — 8-bits
- Total unadjusted error — ± 1/2 LSB and ± 1 LSB
- No missing codes
- Conversion time — 100 μs
- Single supply — 5 V_{DC}
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- 8-channel multiplexer with latched control logic
- Easy interface to all microprocessors, or operates "stand alone"
- Outputs meet T^2L voltage level specifications
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Standard hermetic or molded 28-pin DIP package
- Temperature range −40°C to +85°C or −55°C to +125°C
- Low power consumption — 15 mW
- Latched TRI-STATE® output

Block Diagram



TRI-STATE® is a registered trademark of National Semiconductor Corp

Absolute Maximum Ratings (Notes 1 and 2)

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage at Any Pin Except Control Inputs	−0.3V to (V_{CC} + 0.3V)
Voltage at Control Inputs START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	−0.3V to +15V
Storage Temperature Range	−65°C to +150°C
Package Dissipation at T_A = 25°C	875 mW
Lead Temperature (Soldering, 10 seconds)	300°C

Operating Ratings (Notes 1 and 2)

Temperature Range (Note 1) ADC0808CJ	$T_{MIN} \le T_A \le T_{MAX}$ −55°C ≤ T_A ≤ +125°C
ADC0808CCJ, ADC0808CCN, ADC0809CCN	−40°C ≤ T_A ≤ +85°C
Range of V_{CC} (Note 1)	4.5 V_{DC} to 6.0 V_{DC}

Electrical Characteristics

Converter Specifications: V_{CC} = 5 V_{DC} = $V_{REF(+)}$, $V_{REF(-)}$ = GND, $T_{MIN} \le T_A \le T_{MAX}$ and f_{CLK} = 640 kHz unless otherwise stated.

Parameter	Conditions	Min	Typ	Max	Units
ADC0808 Total Unadjusted Error (Note 5)	25°C			± 1/2	LSB
	T_{MIN} to T_{MAX}			± 3/4	LSB
ADC0809 Total Unadjusted Error (Note 5)	0°C to 70°C			± 1	LSB
	T_{MIN} to T_{MAX}			± 1 1/4	LSB
Input Resistance	From Ref(+) to Ref(−)	1.0	2.5		kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(−)	GND−0.10		V_{CC}+0.10	V_{DC}
$V_{REF(+)}$ Voltage, Top of Ladder	Measured at Ref(+)		V_{CC}	V_{CC}+0.1	V
$\frac{V_{REF(+)}+V_{REF(-)}}{2}$ Voltage, Center of Ladder		V_{CC}/2−0.1	V_{CC}/2	V_{CC}/2+0.1	V
$V_{REF(-)}$ Voltage, Bottom of Ladder	Measured at Ref(−)	−0.1	0		V
Comparator Input Current	f_C = 640 kHz, (Note 6)	−2	±0.5	2	μA

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CJ 4.5V ≤ V_{CC} ≤ 5.5V, −55°C ≤ T_A ≤ +125°C unless otherwise noted ADC0808CCJ, ADC0808CCN, and ADC0809CCN 4.75 ≤ V_{CC} ≤ 5.25V, −40°C ≤ T_A ≤ +85°C unless otherwise noted

Parameter	Conditions	Min	Typ	Max	Units
ANALOG MULTIPLEXER					
$I_{OFF(+)}$ OFF Channel Leakage Current	V_{CC} = 5V, V_{IN} = 5V, T_A = 25°C		10	200	nA
	T_{MIN} to T_{MAX}			1.0	μA
$I_{OFF(-)}$ OFF Channel Leakage Current	V_{CC} = 5V, V_{IN} = 0, T_A = 25°C	−200	−10		nA
	T_{MIN} to T_{MAX}	−1.0			μA
CONTROL INPUTS					
$V_{IN(1)}$ Logical "1" Input Voltage		V_{CC}−1.5			V
$V_{IN(0)}$ Logical "0" Input Voltage				1.5	V
$I_{IN(1)}$ Logical "1" Input Current (The Control Inputs)	V_{IN} = 15V			1.0	μA
$I_{IN(0)}$ Logical "0" Input Current (The Control Inputs)	V_{IN} = 0	−1.0			μA
I_{CC} Supply Current	f_{CLK} = 640 kHz		0.3	3.0	mA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CJ $4.5V \le V_{CC} \le 5.5V$, $-55°C \le T_A \le +125°C$ unless otherwise noted ADC0808CCJ, ADC0808CCN, and ADC0809CCN $4.75 \le V_{CC} \le 5.25V$, $-40°C \le T_A \le +85°C$ unless otherwise noted

Parameter		Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)						
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = -360\ \mu A$	$V_{CC}-0.4$			V
$V_{OUT(0)}$	Logical "0" Output Voltage	$I_O = 1.6$ mA			0.45	V
$V_{OUT(0)}$	Logical "0" Output Voltage EOC	$I_O = 1.2$ mA			0.45	V
I_{OUT}	TRI-STATE® Output Current	$V_O = 5V$			3	μA
		$V_O = 0$	−3			μA

Electrical Characteristics

Timing Specifications: $V_{CC} = V_{REF(+)} = 5V$, $V_{REF(-)} = GND$, $t_r = t_f = 20$ ns and $T_A = 25°C$ unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	*(Figure 5)*		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	*(Figure 5)*		100	200	ns
t_s	Minimum Address Set-Up Time	*(Figure 5)*		25	50	ns
t_H	Minimum Address Hold Time	*(Figure 5)*		25	50	ns
t_D	Analog MUX Delay Time From ALE	$R_S = 0\Omega$ *(Figure 5)*		1	2.5	μs
t_{H1}, t_{H0}	OE Control to Q Logic State	$C_L = 50$ pF, $R_L = 10$k *(Figure 8)*		125	250	ns
t_{1H}, t_{0H}	OE Control to HI-Z	$C_L = 10$ pF, $R_L = 10$k *(Figure 8)*		125	250	ns
t_c	Conversion Time	$f_c = 640$ kHz, *(Figure 5)* (Note 7)	90	100	116	μs
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	*(Figure 5)*	0		8 + 2 μs	Clock Period
C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE® Output Capacitance	At TRI-STATE® Outputs, (Note 12)		10	15	pF

Note 1: Absolute maximum ratings are those values beyond which the life of the device may be impaired.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of 7 V_{DC}.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.900 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See *Figure 3*. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See *Figure 13*.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence *(Figure 6)*. See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Functional Description

Multiplexer: The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. Table I shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE I

SELECTED ANALOG CHANNEL	ADDRESS LINE C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach *(Figure 1)* was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached +1/2 LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

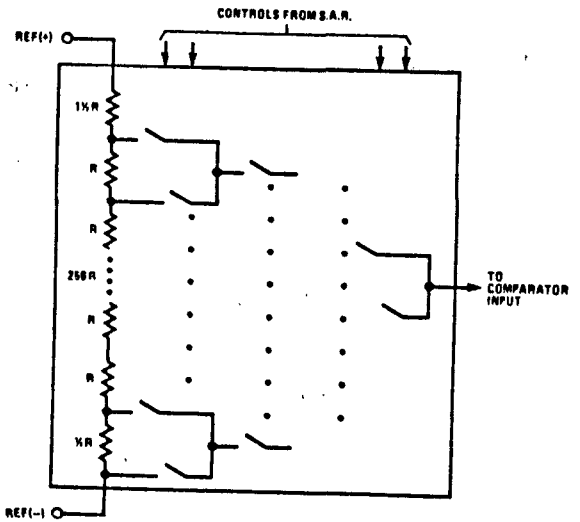


FIGURE 1. Resistor Ladder and Switch Tree

Functional Description (Continued)

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

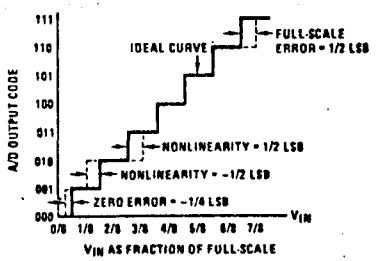


FIGURE 2. 3-Bit A/D Transfer Curve

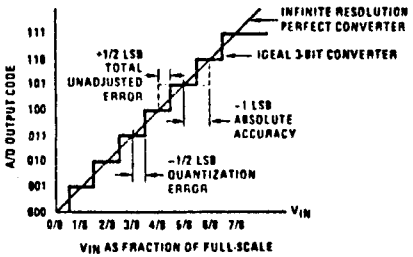


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

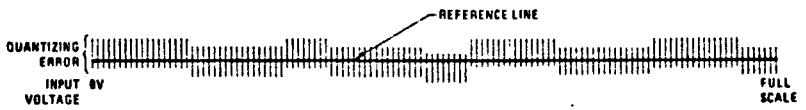
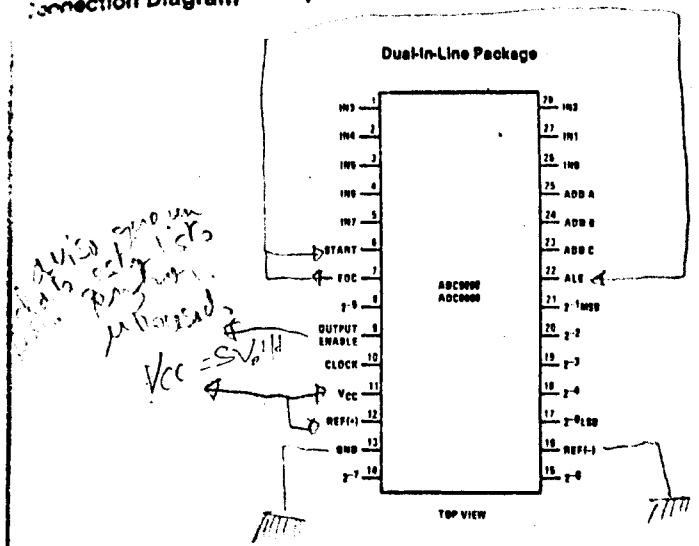


FIGURE 4. Typical Error Curve

Connection Diagram



Timing Diagram

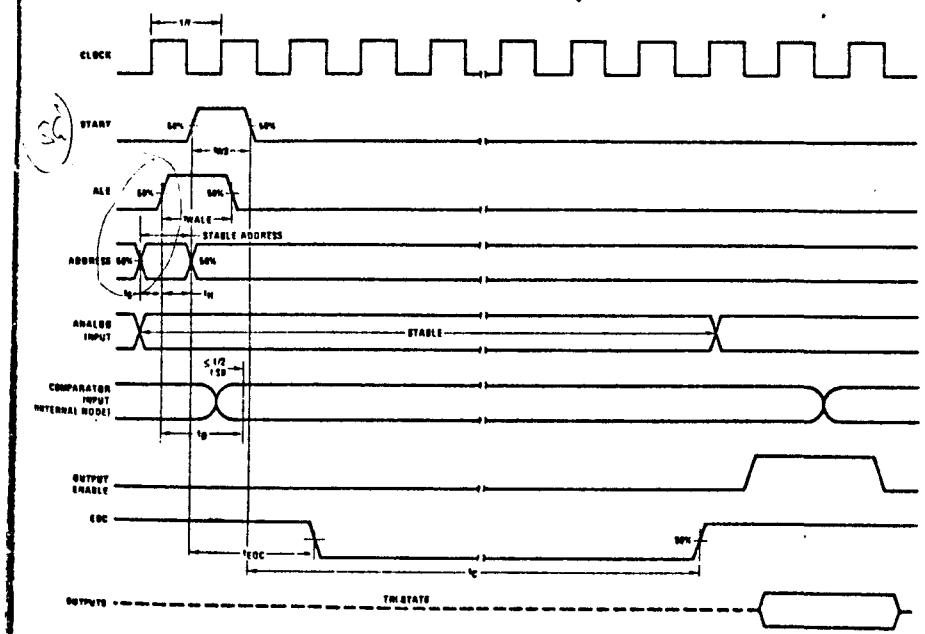


FIGURE 5

ADC0808, ADC0809

Typical Performance Characteristics

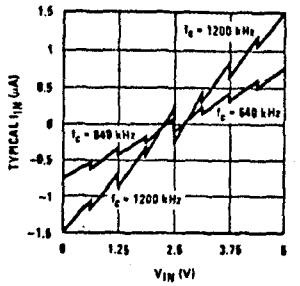


FIGURE 6. Comparator I_{IN} vs V_{IN} ($V_{CC} = V_{REF} = 5V$)

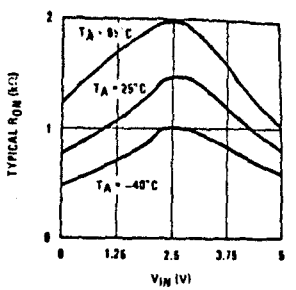


FIGURE 7. Multiplexer R_{ON} vs V_{IN} ($V_{CC} = V_{REF} = 5V$)

TRI-STATE® Test Circuits and Timing Diagrams

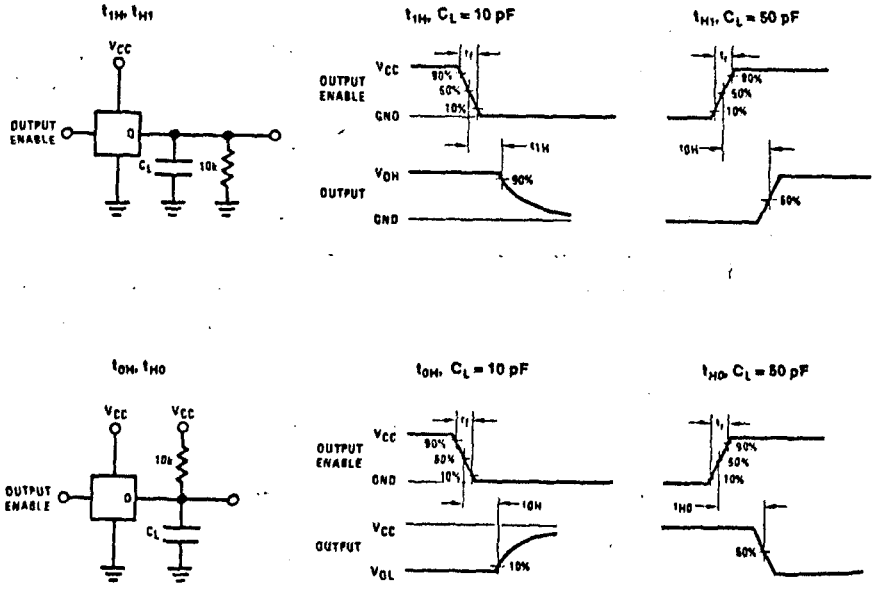


FIGURE 8

Applications Information

OPERATION

1.0 Ratiometric Conversion

The ADC0808, ADC0809 is designed as a complete Data Acquisition System (DAS) for ratiometric conversion systems. In ratiometric systems, the physical variable being measured is expressed as a percentage of full-scale which is not necessarily related to an absolute standard. The voltage input to the ADC0808 is expressed by the equation

$$\frac{V_{IN}}{V_{fs} - V_Z} = \frac{D_X}{D_{MAX} - D_{MIN}} \quad (1)$$

V_{IN} = Input voltage into the ADC0808
V_{fs} = Full-scale voltage
V_Z = Zero voltage
D_X = Data point being measured
D_{MAX} = Maximum data limit
D_{MIN} = Minimum data limit

A good example of a ratiometric transducer is a potentiometer used as a position sensor. The position of the wiper is directly proportional to the output voltage which is a ratio of the full-scale voltage across it. Since the data is represented as a proportion of full-scale, reference requirements are greatly reduced, eliminating a large source of error and cost for many applications. A major advantage of the ADC0808, ADC0809 is that the input voltage range is equal to the supply range so the transducers can be connected directly across the supply and their outputs connected directly into the multiplexer inputs, *(Figure 9)*.

Ratiometric transducers such as potentiometers, strain gauges, thermistor bridges, pressure transducers, etc., are suitable for measuring proportional relationships; however, many types of measurements must be referred to an absolute standard such as voltage or current. This means a system reference must be used which relates the full-scale voltage to the standard volt. For example, if $V_{CC} = V_{REF} = 5.12V$, then the full-scale range is divided into 256 standard steps. The smallest standard step is 1 LSB which is then 20 mV.

2.0 Resistor Ladder Limitations

The voltages from the resistor ladder are compared to the selected input 8 times in a conversion. These voltages are coupled to the comparator via an analog switch tree which is referenced to the supply. The voltages at the top, center and bottom of the ladder must be controlled to maintain proper operation.

The top of the ladder, Ref(+), should not be more positive than the supply, and the bottom of the ladder, Ref(−), should not be more negative than ground. The center of the ladder voltage must also be near the center of the supply because the analog switch tree changes from N-channel switches to P-channel switches. These limitations are automatically satisfied in ratiometric systems and can be easily met in ground referenced systems.

Figure 10 shows a ground referenced system with a separate supply and reference. In this system, the supply must be trimmed to match the reference voltage. For instance, if a 5.12V is used, the supply should be adjusted to the same voltage within 0.1V.

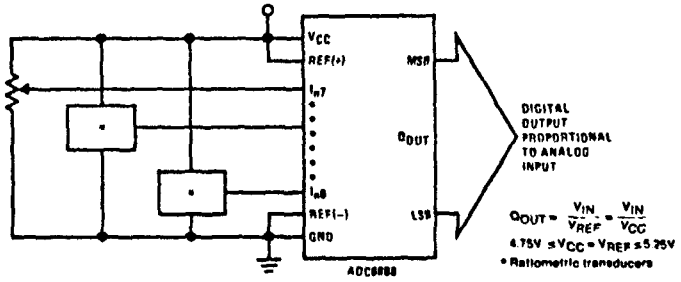


FIGURE 9. Ratiometric Conversion System

Applications Information (Continued)

The ADC0808 needs less than a milliamp of supply current so developing the supply from the reference is readily accomplished. In *Figure 11* a ground referenced system is shown which generates the supply from the reference. The buffer shown can be an op amp of sufficient drive to supply the milliamp of supply current and the desired bus drive, or if a capacitive bus is driven by the outputs a large capacitor will supply the transient supply current as seen in *Figure 12*. The LM301 is overcompensated to insure stability when loaded by the 10 μF output capacitor.

The top and bottom ladder voltages cannot exceed V_{CC} and ground, respectively, but they can be symmetrical, not less than V_{CC} and greater than ground. The center of the ladder voltage should always be near the center of the supply. The sensitivity of the converter can be increased (i.e., size of the LSB steps decreased) by using a symmetrical reference system. In *Figure 13*, a 2.5V reference is symmetrically centered about $V_{CC}/2$ since the same current flows in identical resistors. This system with a 2.5V reference allows the LSB bit to be half the size of a 5V reference system.

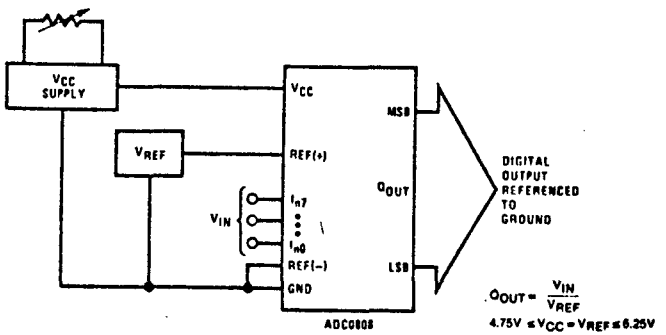


FIGURE 10. Ground Referenced Conversion System Using Trimmed Supply

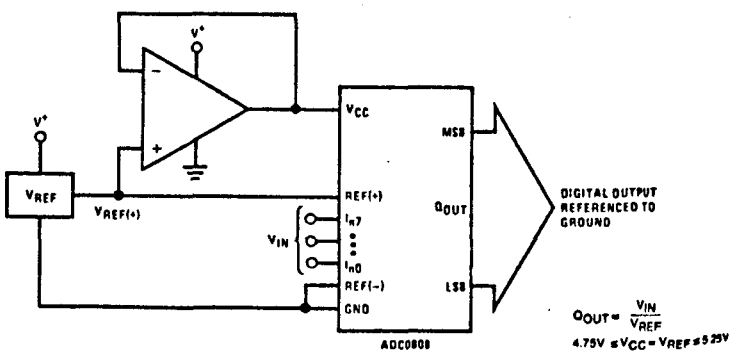


FIGURE 11. Ground Referenced Conversion System with Reference Generating V_{CC} Supply

Applications Information (Continued)

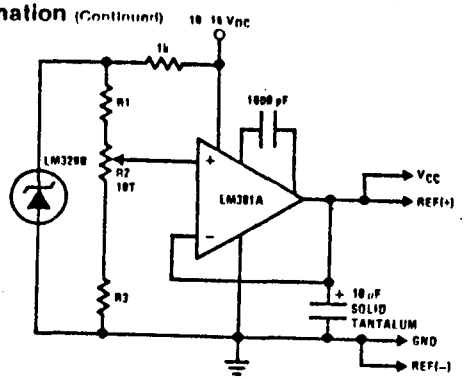


FIGURE 12. Typical Reference and Supply Circuit

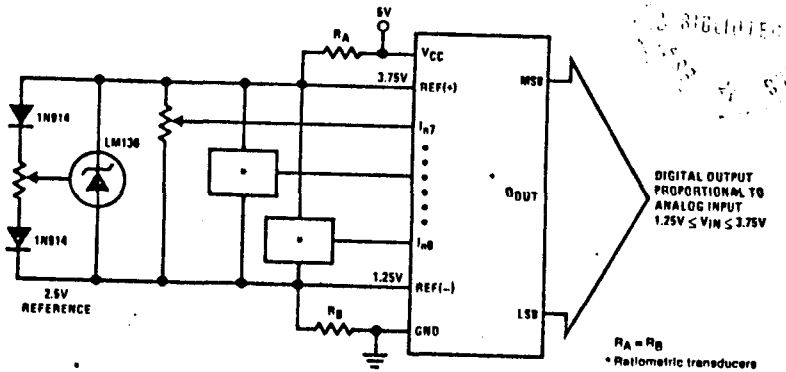


FIGURE 13. Symmetrically Centered Reference

3.0 Converter Equations

The transition between adjacent codes N and N+1 is given by:

$$V_{IN} = \left[(V_{REF(+)} - V_{REF(-)})\left[\frac{N}{256} + \frac{1}{512}\right] \pm V_{TUE}\right] + V_{REF(-)} \quad (2)$$

The center of an output code N is given by:

$$V_{IN} = \left[(V_{REF(+)} - V_{REF(-)})\left[\frac{N}{256}\right] \pm V_{TUE}\right] + V_{REF(-)} \quad (3)$$

The output code N for an arbitrary input are the integers within the range:

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \times 256 \pm \text{Absolute Accuracy} \quad (4)$$

where: V_{IN} = Voltage at comparator input
$V_{REF(+)}$ = Voltage at Ref(+)
$V_{REF(-)}$ = Voltage at Ref(−)
V_{TUE} = Total unadjusted error voltage (typically $V_{REF(-)}$ = 512)

4.0 Analog Comparator Inputs

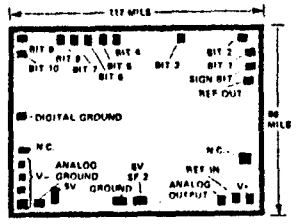
The dynamic comparator input current is caused by the periodic switching of on-chip stray capacitances. These are connected alternately to the output of the resistor ladder/switch tree network and to the comparator input as part of the operation of the chopper stabilized comparator.

The average value of the comparator input current varies directly with clock frequency and with V_{IN} as shown in *Figure 6*.

If no filter capacitors are used at the analog inputs and the signal source impedances are low, the comparator input current should not introduce converter errors, as the transient created by the capacitance discharge will die out before the comparator output is strobed.

If input filter capacitors are desired for noise reduction and signal conditioning they will tend to average out the dynamic comparator input current. It will then take on the characteristics of a DC bias current whose effect can be predicted conventionally.

DICE DIMENSIONS



NOTE:
VOLTAGE OUTPUT SET FOR 10V RANGE. MAY BE PROGRAMMED FOR 5V RANGE AS FOLLOWS: JUMPER PAD 5V/SF2 TO ANALOG OUTPUT AND PAD 5V TO ANALOG GROUND.

PRELIMINARY

DAC-808

8-BIT HIGH-SPEED "MICROPROCESSOR COMPATIBLE" MULTIPLYING D/A CONVERTER

FEATURES

- Dual 4-Bit Input Latch Coupled to 8-Bit Latched DAC
- 8 and 4-Bit μP Compatible
- Easily Interfaced to 8080, and Z-80 Processors
- TTL Logic Compatible
- Programmable Mode Control
- High Output Impedance and Compliance
- Proven DAC-08 Analog Flexibility
- Nonlinearity to ±0.1% Maximum
- Low Power Dissipation 150mW

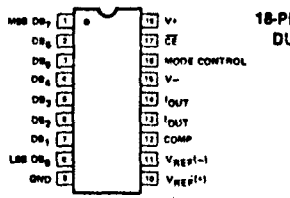
GENERAL DESCRIPTION

The BYTEDAC™ DAC-808 is a Double-Buffered Latch Input Digital-to-Analog Converter designed specifically for 8- and 4-bit microprocessors. The double latch concept allows the processor to load data in the master latch without disturbing existing data in the slave latch which controls the analog output. The DAC-808 operates in five modes which are selected by the user under processor control. Data transfer is accomplished in two 4-bit nibbles, one 4-bit nibble, or one 8-bit byte.

The Analog section consists of a "Field-Proven" DAC-08 D/A Converter. Monotonic multiplying performance is attained over a wide 40 to 1 reference current range. Matching to within 1 LSB between reference and full-scale currents eliminates full-scale adjustments in most applications.

DAC-808 applications include graphic display drivers, high-speed modems, A/D converters, programmable waveform generators and power supplies, analog meter drivers, audio encoders and programmable attenuators, and other applications where low cost, high speed and double-buffering flexibility are required.

PIN CONNECTIONS & ORDERING INFORMATION



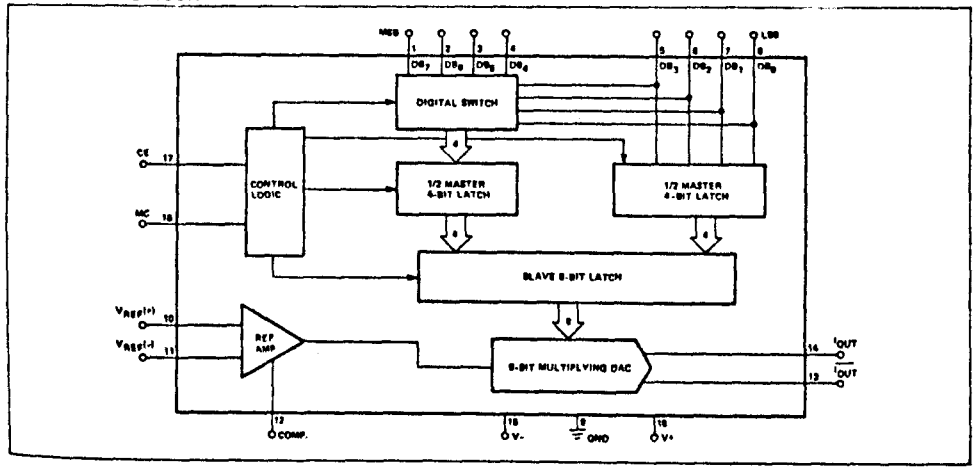
18-PIN HERMETIC DUAL-IN-LINE (X-Suffix)

DAC-808AX
DAC-808BX
DAC-808EX
DAC-808FX
DAC-808GX

Military Temperature Range Devices With MIL-STD-883 Class B Processing

ORDER: DAC808AX/883
DAC808BX/883

EQUIVALENT CIRCUIT



ABSOLUTE MAXIMUM RATINGS

Operating Temperature
- DAC-808A/B ..... −55°C to +125°C
- DAC-808E/F ..... −25°C to +85°C
- DAC-808G ..... 0°C to +70°C

Storage Temperature ..... −65°C to +150°C
Power Dissipation 300mW
Derate above 100°C ..... 10mW/°C
Lead Soldering Temperature ..... 300°C (60 sec)

V+ Supply to V− Supply 15.5V
Logic Inputs ..... 0V to 5.5V
Analog Current Outputs ..... −5mA
Reference Inputs (V_{14}, V_{15}) ..... V− to V+
Reference Input Differential Voltage (V_{14} to V_{15}) ..... ±15V
Reference Input Current 5.0mA

ELECTRICAL CHARACTERISTICS

at V+ = +5V, V− = −10V, I_{REF} = 2.0mA, T_A = −55°C to +125°C for DAC-808A/B, unless otherwise noted. T_A = −25°C to +85°C apply for DAC-808E/F; T = 0°C to +70°C apply for DAC-808G. Output characteristics refer to both I_{OUT} and $\overline{I_{OUT}}$.

PARAMETER	SYMBOL	CONDITIONS	DAC-808A/E MIN	DAC-808A/E TYP	DAC-808A/E MAX	DAC-808B/F MIN	DAC-808B/F TYP	DAC-808B/F MAX	DAC-808G MIN	DAC-808G TYP	DAC-808G MAX	UNITS
Resolution			8	8	8	8	8	8	8	8	8	Bits
Monotonicity			8	8	8	8	8	8	8	8	8	Bits
Nonlinearity			—	—	±0.1	—	—	±0.19	—	—	±0.39	%FS
Full Scale Tempco	TCI$_{FS}$		—	±10	±50	—	±10	±80	—	±10	±80	ppm/°C
Output Voltage Compliance	V_{OC}	Full Scale Current Change < ½ LSB R_{OUT} > 20 MΩ Typ.	−5	—	+8	−5	—	+8	−5	—	+8	V
Full Range Current	I_{FR14}	V_{REF} = 5.00V R_{11},R_{10} = 2.500kΩ T_A = 25°C	1.94	1.99	2.04	1.94	1.99	2.04	1.94	1.99	2.04	mA
Full Range Symmetry	I_{FRS}	I_{FR14}-I_{FR13}	—	±1.0	±8.0	—	±1.0	±8.0	—	±1.0	±8.0	μA
Zero Scale Current	I_{ZS}		—	0.2	2.0	—	0.2	2.0	—	0.2	2.0	μA
Output Current Range	I_{FSR}	V− = −10V	0	2.0	2.1	0	2.0	2.1	0	2.0	2.1	mA
Reference Bias Current	I_B		—	−1.0	−3.0	—	−1.0	−3.0	—	−1.0	−3.0	μA
Reference Input Slew Rate	dI/dt		4.0	8.0	—	4.0	8.0	—	4.0	8.0	—	mA/μs
Power Supply Sensitivity	$PSSI_{FR+}$	V+ = +4.5V to 5.5V	—	±0.0003	±0.01	—	0.0003	±0.01	—	±0.0003	±0.01	$\frac{\%\Delta I_{FS}}{\%\Delta V+}$
	$PSSI_{FR-}$	V− = −4.5V to −10V I_{REF} = 1mA	—	±0.002	±0.01	—	±0.002	±0.01	—	±0.002	±0.01	$\frac{\%\Delta I_{FS}}{\%\Delta V-}$
Power Supply Current	I+	V_S = +5V, −10V, I_{REF} = 2.0mA	—	12	18	—	12	18	—	12	18	mA
	I−		—	8	9	—	8	9	—	8	9	
Power Dissipation	P_d	+5V, −10V, I_{REF} = 2.0mA	—	120	170	—	120	170	—	120	170	mW
Logic Input Levels												
Logic Input "0"	V_{IL}		—	—	0.8	—	—	0.8	—	—	0.8	V
Logic Input "1"	V_{IH}		2.0	—	—	2.0	—	—	2.0	—	—	

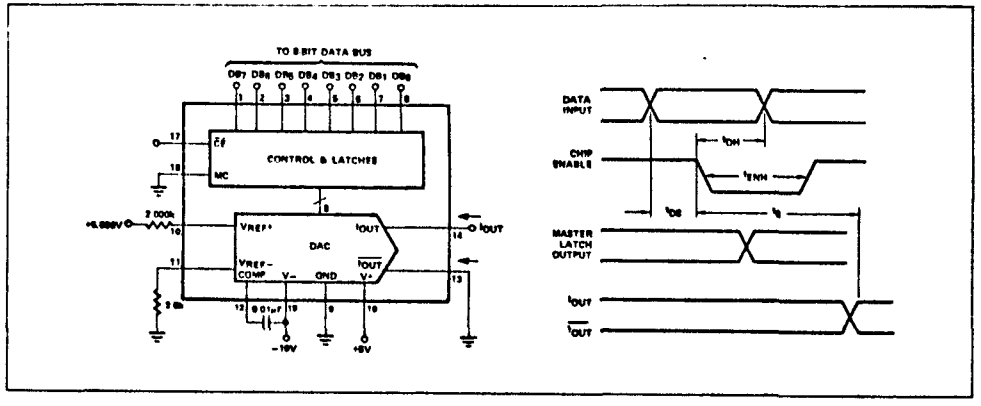
ELECTRICAL CHARACTERISTICS TARGET SPECIFICATIONS — A.C. PARAMETERS (These are design goals)

PARAMETER	SYMBOL	CONDITIONS	DAC-808A/E MIN	DAC-808A/E TYP	DAC-808A/E MAX	DAC-808B/F MIN	DAC-808B/F TYP	DAC-808B/F MAX	UNITS
Settling Time	t_s	From CE Negative Edge to ± ½ LSB, All Bits Switched ON or OFF, T_A = 25°C	—	300	500	—	300	500	ns
Data Input Setup Time	t_{DS}	T_A = 25°C	50	30	—	50	30	—	ns
Data Input Hold Time	t_{DH}	T_A = 25°C	—	30	100	—	30	100	ns
Address Input Setup Time (Bits 7 and 8)	t_{AS}	4-Bit Mode, T_A = 25°C	150	100	—	150	100	—	ns
Address Hold Time	t_{AH}	4-Bit Mode, T_A = 25°C	—	0	10	—	0	10	ns
Chip Enable Negative Hold Time	t_{ENH}	T_A = 25°C	250	100	—	250	100	—	ns
Chip Enable Positive Hold Time	t_{EPH}	T_A = 25°C	350	200	—	350	200	—	ns

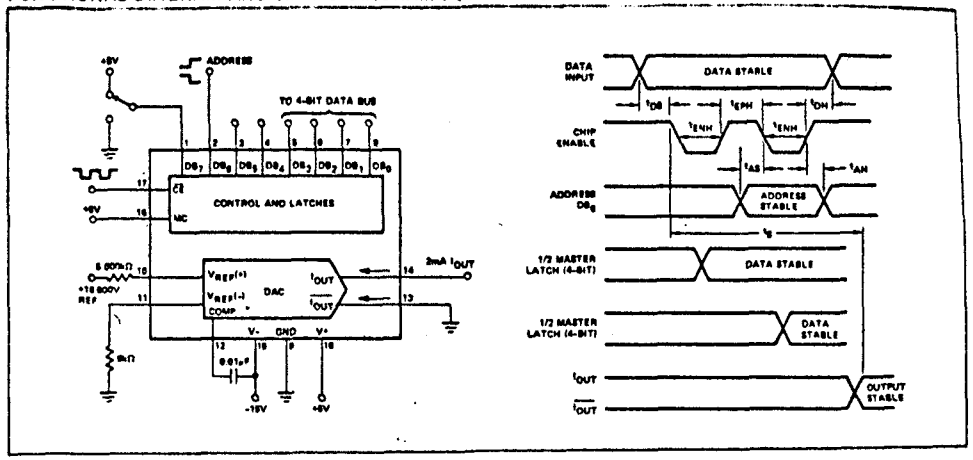
DAC-808 PIN DESCRIPTION

SYMBOL	DESCRIPTION
DB_0-DB_7	DATA BIT — Bits 0-7 are digital, active-high inputs that have DB_7 assigned the MSB.
CE	CHIP ENABLE — An active low input control serving a dual purpose in that it's both the device enable and chip write input terminal.
MC	MODE CONTROL — A control that places the DAC in 8-bit operation when low and 4-bit operation when high.
I_{OUT}-$\overline{I_{OUT}}$	CURRENT OUTPUT — Complementary current outputs when added equal I_{FS}.
V_{REF-}, V_{REF+}	VOLTAGE REFERENCE — Differential inputs that accept a negative, positive, or bipolar input and are used to adjust I_{FS}.
COMP	COMPENSATION — The reference amplifier frequency compensating terminal.

FUNCTIONAL DIAGRAM AND TIMING DIAGRAM FOR 8-BIT OPERATION



FUNCTIONAL DIAGRAM AND TIMING DIAGRAM FOR 4-BIT OPERATION



DAC-808 FUNCTION TABLE

NO.	MODE	FUNCTION	DESCRIPTION	CE	MC	PIN 1 DB7	PIN 2 DB6
1	8-Bit	8-bit byte transfer (1 cycle)	a) Data transfer to master	↓	0	X	X
			b) Data to slave match	↑	0	X	X
2	4-Bit (2 nibbles)	Two 4-bit transfers using 2 cycles. DB0-DB3 LSB first.	a) 4 bits to LSB's master	↓	1	0	0
			b) No change in latch	↑	1	0	0
			c) 4 bits to MSB's master	↓	1	0	1
			d) 8 bits in master to slave and output	↑	1	0	1
3	4-Bit (2 nibbles)	Two 4-bit transfers using 2 cycles. DB0-DB3 MSB first.	a) 4 bits to MSB's master	↓	1	1	1
			b) No change	↑	1	1	1
			c) 4 bits to LSB's master	↓	1	1	0
			d) 8 bits in master to slave and output	↑	1	1	0
4	4-Bit	4-bit transfer using 1 cycle.	a) 4 bits to LSB's master	↓	1	1	0
			b) 4 bits in master to LSB slave and output	↑	1	1 OR	0
		Input DB0-DB3. Ground DB4-DB7.			0	X	X
					0	X	X
5	4-Bit	4-bit transfer using 1 cycle.	a) 4 bits to MSB's master	↓	1	0	1
			b) 4 bits in master to MSB slave and output	↑	1	0 OR	1
		Input DB4-DB7. Ground DB0-DB3.			0	X	X
					0	X	X
6	None	No operation	Chip disabled — Previous output still present	1	X	X	X

LSB — Least Significant Bit
MSB — Most Significant Bit
X — Insignificant (Don't Care)

↑ — Positive Transition
↓ — Negative Transition
DB — Data Bit

DIGITAL INFORMATION

The DAC-808 is a monolithic microprocessor or compatible device consisting of a quad digital switch, two 4-bit master latches, one 8-bit slave, latch, control circuitry, and one 8-bit multiplying DAC; all housed in an 18-pin Dual In-line Package.

The DAC-808 can be thought of, in the 4-bit mode, as a quad 1 to 2-line digital demultiplexer which selects a 4-bit input and transfers this data to one of two 4-bit master latches.

The BYTEDAC accepts a straight binary digital byte at the master latch which is a fast edge-triggered device. Two 4-bit independent latches make up the master latch and are clocked separately depending on the state of the Mode Control (MC). The second latch, or slave latch, is 8 bits and connects directly with the DAC. The Chip Enable (CE) is used to clock data to and from both latches. When CE is high, the DAC will output a current equal to the last digital value entered (refer to the Equivalent Circuit).

8-BIT MODE #1

To load 8-bit parallel data, a low must be present at MC which sets both master latches in a condition for simultaneous clocking. The negative transition on CE will now transfer data to the master latch while the positive transaction clocks data to the slave latch and input to the DAC (see the Timing Diagram). The CE line can be held low for an indefinite period to prevent data transfer.

INTERFACE TO 4-BIT BUS

The DAC-808 is able to handle 4-bit data in four ways, which will be discussed here. Modes 2 and 3 transfer two 4-bit nibbles which are assembled at the slave latch into an 8-bit byte (refer to Function Table). Modes 4 and 5 transfer a single 4-bit nibble only.

LOADING LSB NIBBLE MODE #2

For all 4-bit operations the MC pin must be high. A low must be applied to Data Bit 6 (DB6) which now acts as an address pin. Data is brought in at DB0 through DB3 and clocked into the LSB master latch on the negative transition of CE. Nothing occurs on the positive transition of the CE's first cycle. DB6 must now go high to enable the second master latch which is loaded through the digital switch on the next negative transition at CE. Both the MSB and LSB nibbles are then loaded in the slave latch on the positive transition at CE. Data Bit 7 (DB7) must remain low in this mode.

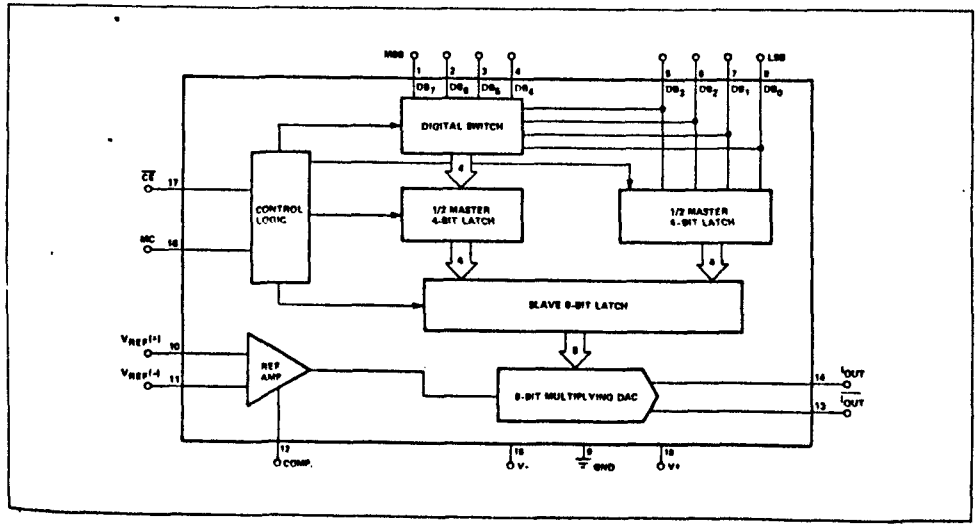
LOADING MSB NIBBLE FIRST MODE #3

This mode is identical to Mode 2 except DB7 remains high and DB6 is high during the first cycle and low during the second cycle. The MSB nibble is loaded into the master latch through the digital switch during the first CE cycle. The second CE cycle loads LSB nibble and transfers all 8 bits to slave latch and DAC.

LOADING 4 BITS (DB0 THROUGH DB3) MODE #4

By applying a low at MC and entering data at DB0 through DB3, 4 bits of data can be loaded. Again, the nibble is latched into the LSB master latch on negative CE and clocked to the lower slave inputs at CE positive. DB7 must be high and DB6 must be low. The MSB nibble will contain and hold the last data entered into it. If four LSBs of resolution are all that will be required, the data may be entered in the 8-bit mode with MC low and DB4 through DB7 tied high or low.

DAC-808 EQUIVALENT CIRCUIT

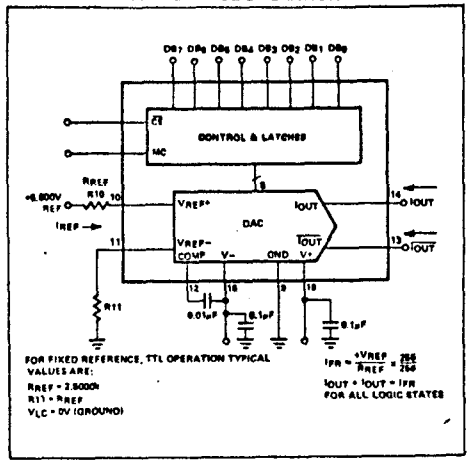


LOADING 4 BITS (DB4 THROUGH DB7) MODE #5

This is the same as Mode 4 except that DB7 is now low and DB6 is now high. The data is still entered into DB0 through DB3, but the data is now loaded into the MSB nibble. The LSB nibble will contain and hold the last data entered into it. If 4 MSBs of resolution are all that will be required, the data may be entered in the 8-bit mode with MC low and DB0 through DB3 tied high or low.

ANALOG INFORMATION

BASIC POSITIVE REFERENCE OPERATION



REFERENCE AMPLIFIER SETUP

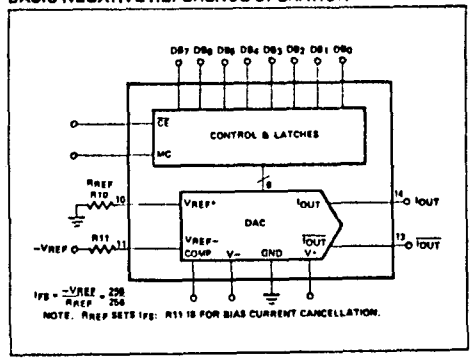
The DAC-808 is a multiplying D/A converter in which the output current is the product of a digital number and the input reference current. The reference current may be fixed or may vary from nearly 0 to +4.0mA. The full range output current is a linear function of the reference current and is given by:

$$I_{FR} = \frac{255}{256} \times I_{REF} \text{ where } I_{REF} = I_{10}$$

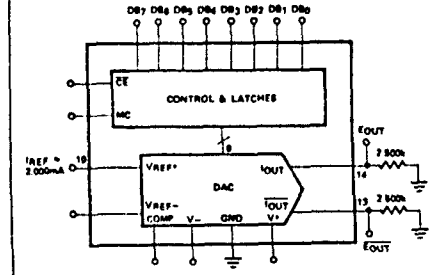
In positive reference applications, an external positive reference voltage current flows through R_{10} into the $V_{REF(+)}$ terminal of the reference amplifier. Alternatively, a negative reference may be applied to $V_{REF(-)}$; reference current flows from ground through R_{10} into $V_{REF(+)}$ as in the positive reference case. This negative reference connection has the advantage of a very high impedance presented at pin 11. The voltage at pin 10 is equal to and tracks the voltage at pin 11 due to the high gain of the internal reference amplifier. R_{11} (nominally equal to R_{10}) is used to cancel bias current errors; R_{11} may be eliminated with only a minor increase in error.

For most applications the tight relationship between I_{REF} and I_{FR} will eliminate the need for trimming I_{REF}. If required, full-scale trimming may be accomplished by adjusting the value of R_{10} or by using a potentiometer for R_{10}. An improved method of full-scale trimming which eliminates potentiometer TC effects is shown in the Recommended Full Scale Adjustment Circuit.

BASIC NEGATIVE REFERENCE OPERATION



BASIC UNIPOLAR NEGATIVE OPERATION



	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	I_OmA	$\overline{I_O}$mA	E_O	$\overline{E_O}$
FULL SCALE – 1 LSB	1	1	1	1	1	1	1	1	1.992	0.000	– 4.980	0.000
FULL SCALE – 2 LSB	1	1	1	1	1	1	1	0	1.984	0.008	– 4.960	– 0.020
HALF SCALE + LSB	1	0	0	0	0	0	0	1	1.008	0.984	– 2.520	– 2.460
HALF SCALE	1	0	0	0	0	0	0	0	1.000	0.992	– 2.500	– 2.480
HALF SCALE – LSB	0	1	1	1	1	1	1	1	0.992	1.000	– 2.480	– 2.500
ZERO SCALE + LSB	0	0	0	0	0	0	0	1	0.008	1.984	– 0.020	– 4.960
ZERO SCALE	0	0	0	0	0	0	0	0	0.000	1.992	– 0.000	– 4.980

RECOMMENDED FULL SCALE ADJUSTMENT CIRCUIT

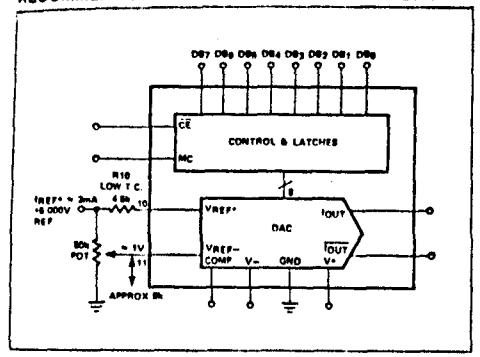


TABLE 1. REFERENCE AMPLIFIER COMPENSATION

REF. INPUT RESISTANCE	SUGGESTED C_C
1kΩ	15pF
2.5kΩ	37pF
5kΩ	75pF

NOTE: A 0.01μF capacitor is suggested for fixed references.

Using lower values of reference current reduces negative power supply current and increases reference amplifier negative common mode range. The recommended range for operation with a DC reference current is +0.2mA to +4.0mA.

The reference amplifier must be compensated by using a capacitor from pin 12 to V–. For fixed reference operation, a 0.01μF capacitor is recommended. For variable reference applications, see "Reference Amplifier Compensation for Multiplying Applications" section.

REFERENCE AMPLIFIER COMPENSATION FOR MULTIPLYING APPLICATIONS

AC reference applications will require the reference amplifier to be compensated using a capacitor from pin 12 to V–. The value of this capacitor depends on the impedance presented to pin 10 (see Table 1).

For fastest response to a pulse, low values of R_{10}, enabling small C_C values, should be used. If pin 10 is driven by a high current source, none of the above values will suffice and the amplifier must be heavily compensated which will decrease overall bandwidth and slew rate. For $R_{10} = 1k\Omega$ and $C_C = 15pF$, the reference amplifier slews at 4mA/μs, enabling a transition from $I_{REF} = 0$ to $I_{REF} = 2mA$ in 500ns (see Figure 6).

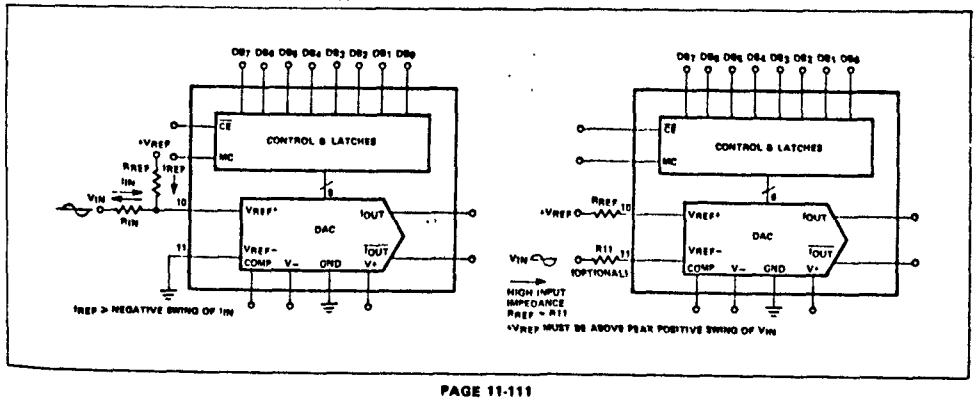
Bipolar references may be accommodated by offsetting V_{REF} or pin 11, as shown in Figure 5. The negative common mode range of the reference amplifier is given by $V_{CM} = V- + (I_{REF} \times 1k\Omega)$ plus 2.5V. The positive common mode range is V+ less 1.5V.

When a DC reference is used, a reference bypass capacitor is recommended. A 5.0V TTL Logic supply is not recommended as a reference. If a regulated power supply is used as a reference, R_{10} should be split into two resistors with the junction bypassed to ground with a 0.1μF capacitor.

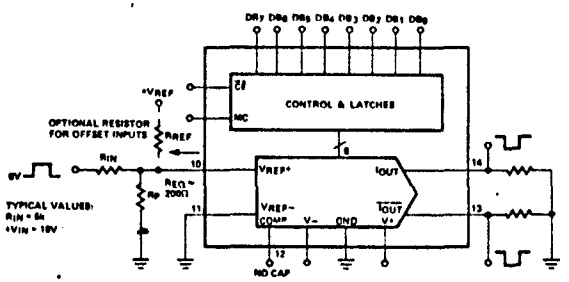
ANALOG OUTPUT CURRENTS

Both true and complemented output sink currents are provided, where $I_O + \overline{I_O} = I_{FR}$. Current appears at the "true" output when a "1" is applied to each logic input. As the binary count increases, the sink current at pin 14 increases proportionally in the fashion of a "positive logic" D/A converter. When a "0" is applied to any input bit, that current is turned off at pin 14 and turned on at pin 13. A decreasing logic count increases $\overline{I_O}$ as in a negative or inverted logic D/A converter. Both outputs may be used simultaneously. If one of the outputs is not required it must still be connected to ground or to a point capable of sourcing I_{FS}; do not leave an unused output pin open.

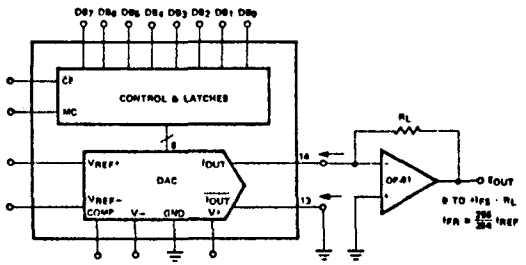
ACCOMMODATING BIPOLAR REFERENCES



ALTERNATE PULSED REFERENCE OPERATION

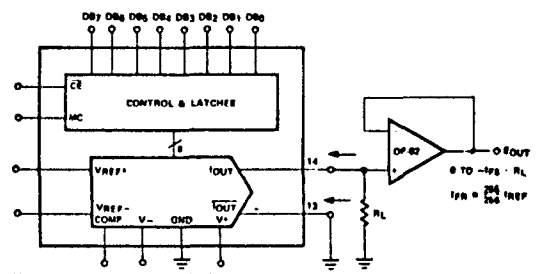


POSITIVE LOW IMPEDANCE OUTPUT OPERATION



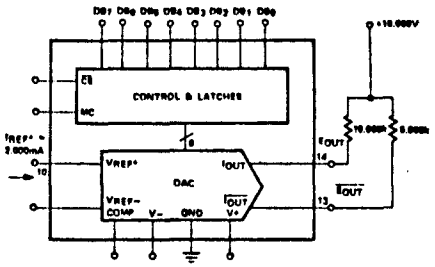
FOR COMPLEMENTARY OUTPUT (OPERATION AS NEGATIVE LOGIC DAC), CONNECT INVERTING INPUT OF OP AMP TO $\overline{I_{OUT}}$. CONNECT I_{OUT} TO GROUND.

NEGATIVE LOW IMPEDANCE OUTPUT OPERATION



FOR COMPLEMENTARY OUTPUT (OPERATION AS NEGATIVE LOGIC DAC), CONNECT INVERTING INPUT OF OP AMP TO $\overline{I_{OUT}}$. CONNECT I_{OUT} TO GROUND.

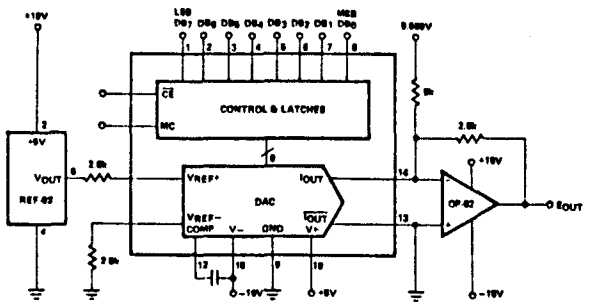
BASIC BIPOLAR OUTPUT OPERATION



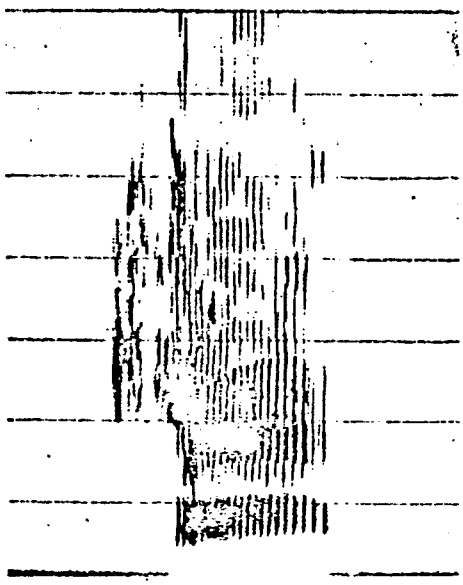
	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	E_O	E_O
POSITIVE FULL SCALE	1	1	1	1	1	1	1	1	−4.960	5.000
POSITIVE FULL SCALE − LSB	1	1	1	1	1	1	1	0	−4.920	4.960
ZERO SCALE + LSB	1	0	0	0	0	0	0	1	−0.040	0.080
ZERO SCALE	1	0	0	0	0	0	0	0	0.000	0.040
ZERO SCALE − LSB	0	1	1	1	1	1	1	1	0.040	0.000
NEGATIVE FULL SCALE + LSB	0	0	0	0	0	0	0	1	4.900	−4.920
NEGATIVE FULL SCALE	0	0	0	0	0	0	0	0	5.000	−4.960

Both outputs have an extremely wide voltage compliance, enabling fast direct current-to-voltage conversion through a resistor tied to ground or other voltage source. Positive compliance is 18V above V− and is independent of the positive supply. Negative compliance is given by V− plus (I_{REF} × 1kΩ) plus 2.5V.

SYMMETRICAL OFFSET BINARY OPERATION

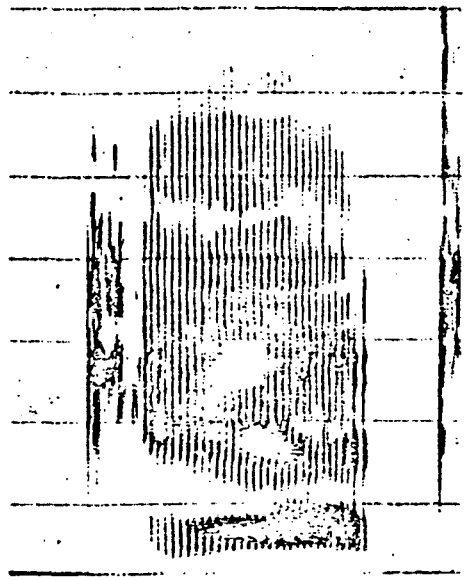


	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	E_O
POSITIVE FULL SCALE − 1 LSB	1	1	1	1	1	1	1	1	4.960
POSITIVE FULL SCALE − 2 LSB	1	1	1	1	1	1	1	0	4.920
(+) ZERO SCALE	1	0	0	0	0	0	0	0	0.020
(−) ZERO SCALE	0	1	1	1	1	1	1	1	−0.020
NEGATIVE ZERO SCALE + 2 LSB	0	0	0	0	0	0	0	1	4.920
NEGATIVE FULL SCALE + 1 LSB	0	0	0	0	0	0	0	0	−4.960

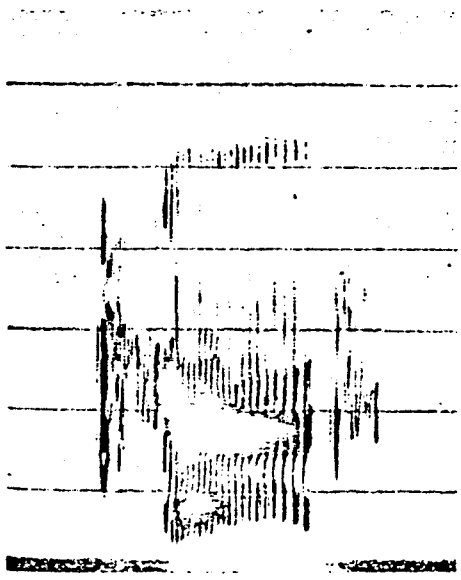


"CAT" [KæT]
("GATO") HABLANTE 1 MICROFONO

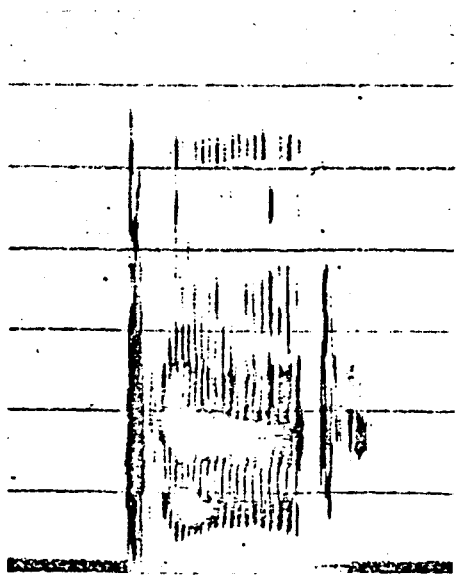
"CAT" [KæT]
("GATO") HABLANTE 1 CUCHICHEADO



"CAT" [KæT]
("GATO") HABLANTE 2 MICROFONO



"CAT" [KæT]
("GATO") HABLANTE 1 TELEFONO



"PAT" [PæT]
("PALMADA") HABLANTE 1 TELEFONO

"CAT" [KæT]
("GATO") HABLANTE 3 MICROFONO

VERSATILIDAD del habla, ilustrada aquí por espectrogramas; constituye una de las principales dificultades con que tropieza la construcción de un sistema automático para el reconocimiento del habla. Los espectrogramas de palabras distintas pero acústicamente similares pueden resultar mas parecidos que los espectrogramas de la misma palabra pronunciada en condiciones diversas por hablantes diferentes. El reconocimiento automático del habla debe ser capaz de atender tan sólo a las diferencias espectrales pertinentes (cuando existen) y soslayar aquellas que son lingüísticamente irrelevantes. Los espectrogramas de sonido representan una serie de espectros de amplitud a lo largo del tiempo. El factor tiempo discurre por el eje horizontal y la frecuencia por el vertical. Cuanto más oscura es la mancha del grafico mayor es la amplitud de la onda en el momento y la frecuencia respectivos.

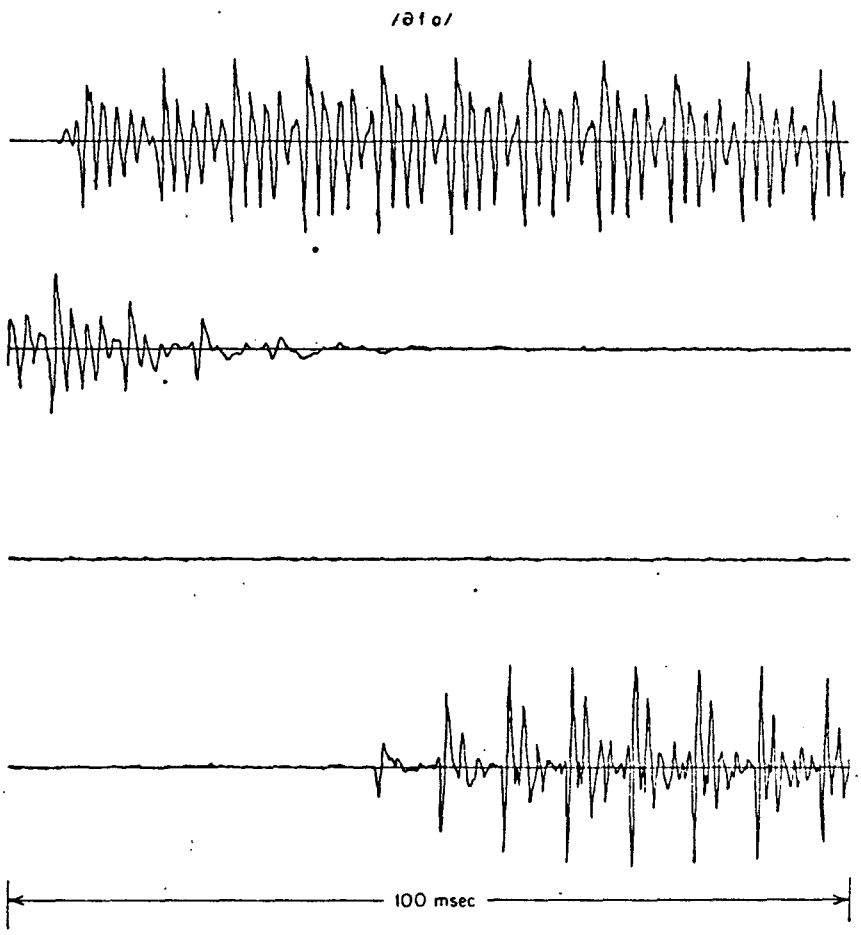


Fig. 3.9 Acoustic waveforms and spectrograms for */UH–F–A/*, */UH–S–A/*, and */UH–SH–A/*.

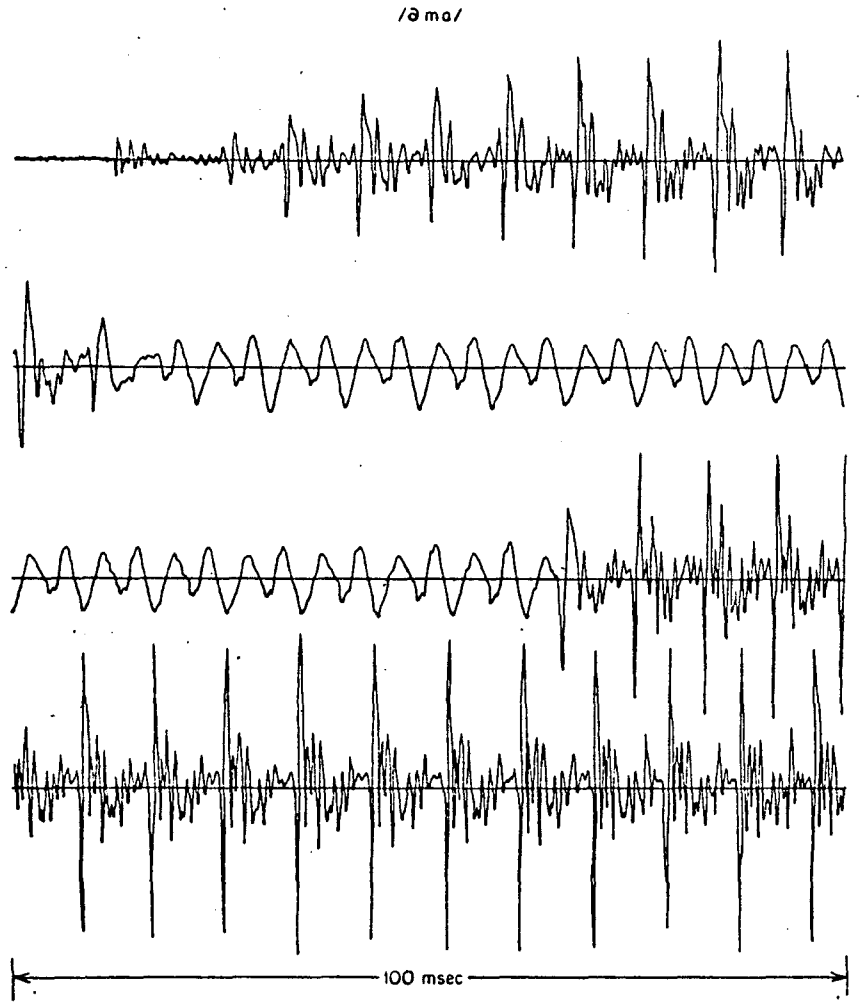


Fig. 3.8 Acoustic waveforms and spectrograms for utterances */UH–M–A/* and */UH–N–A/*.

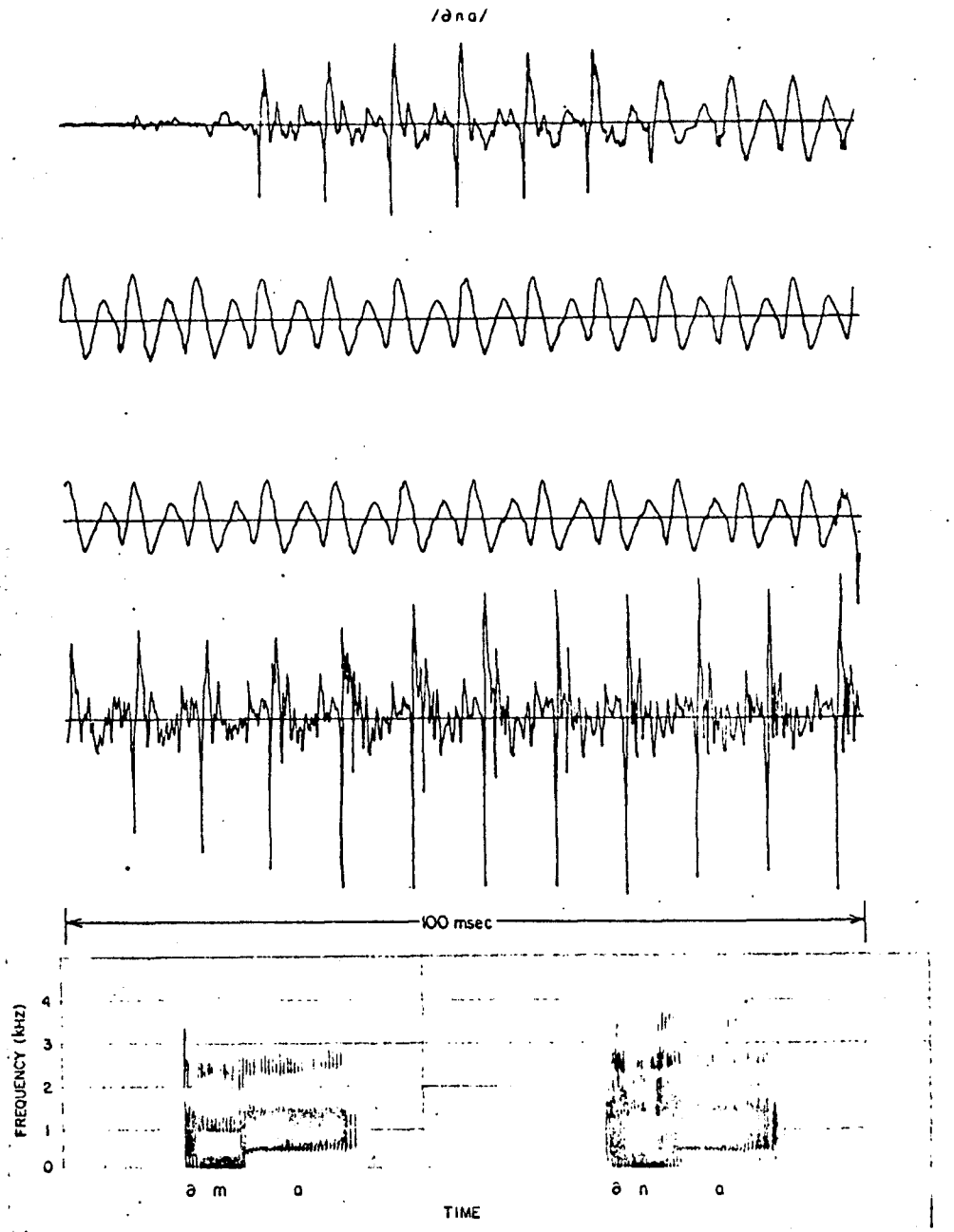


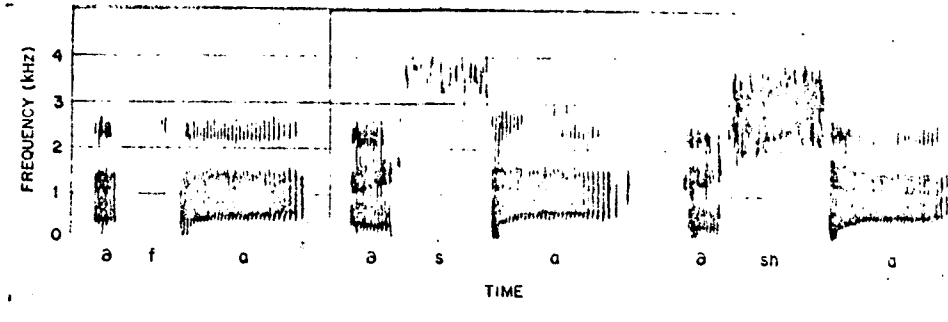
Fig. 3.8 (Continued)

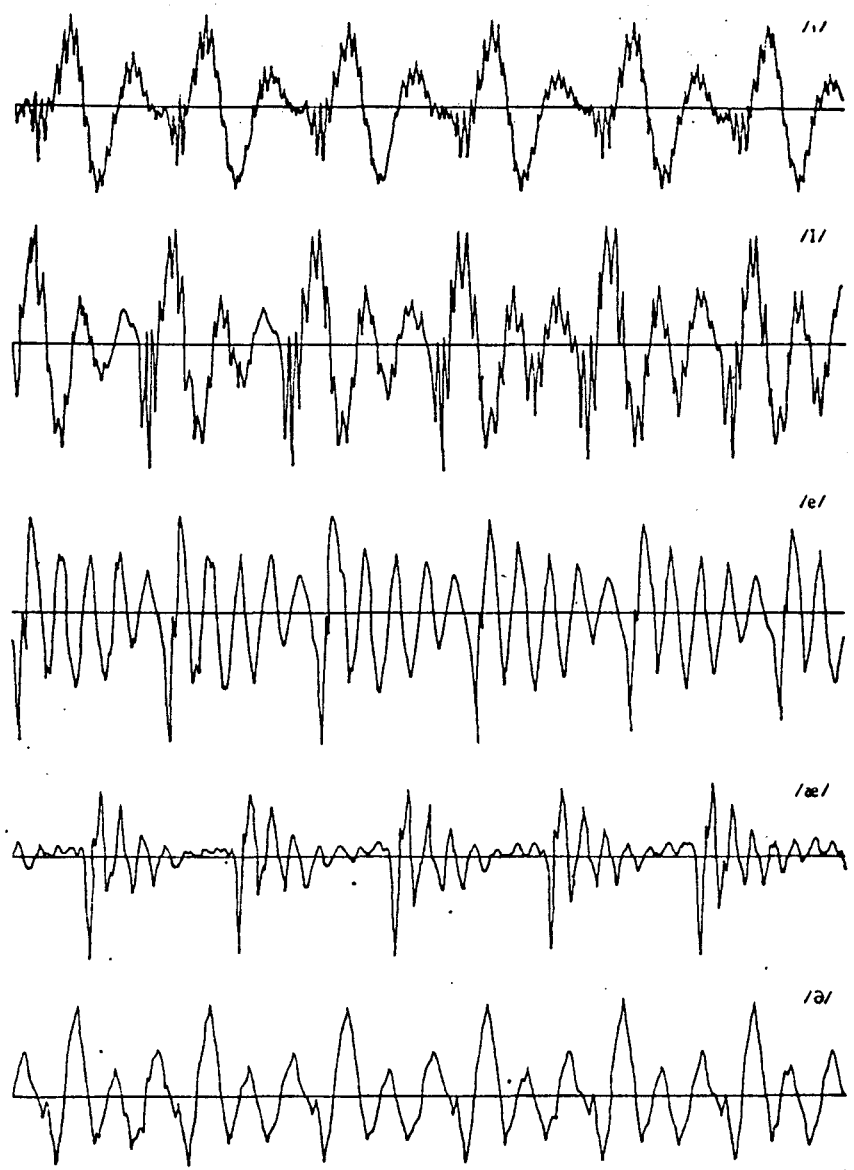
/əsa/

100 msec

/əsna/

100 msec





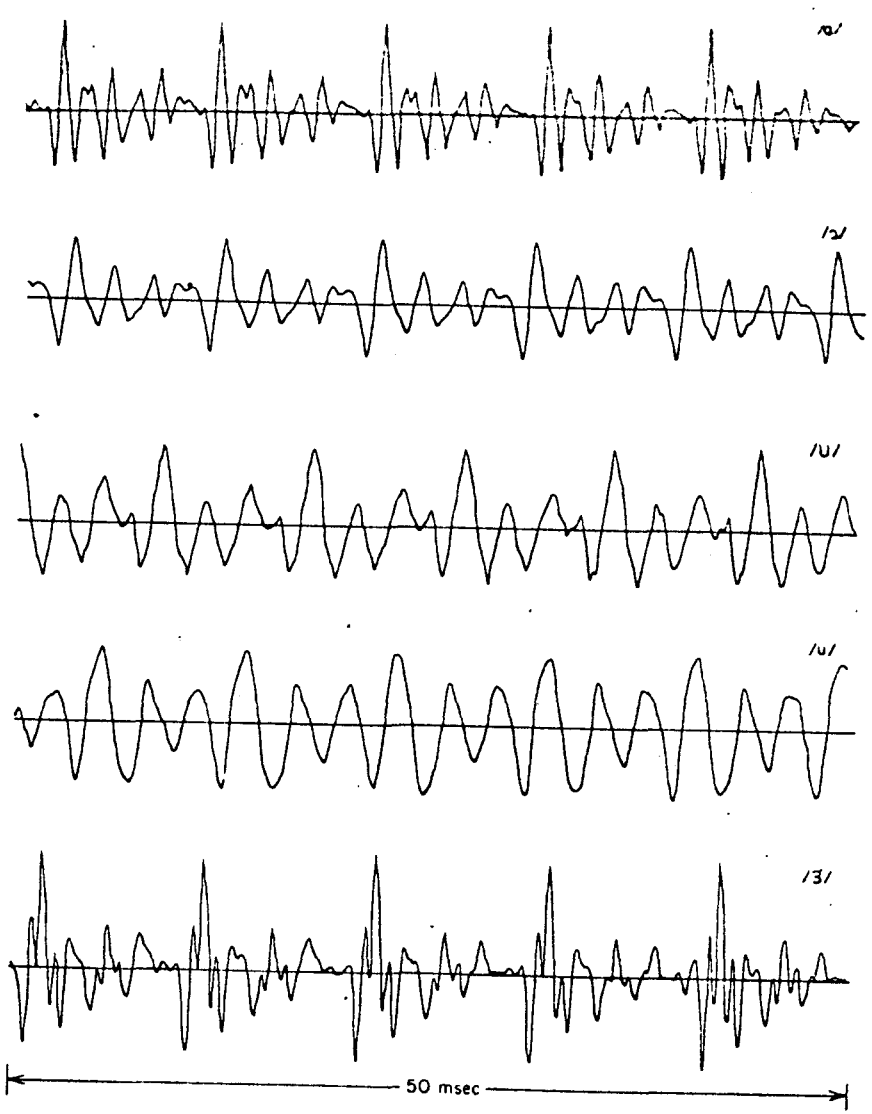


Fig. 3.6 (Continued)

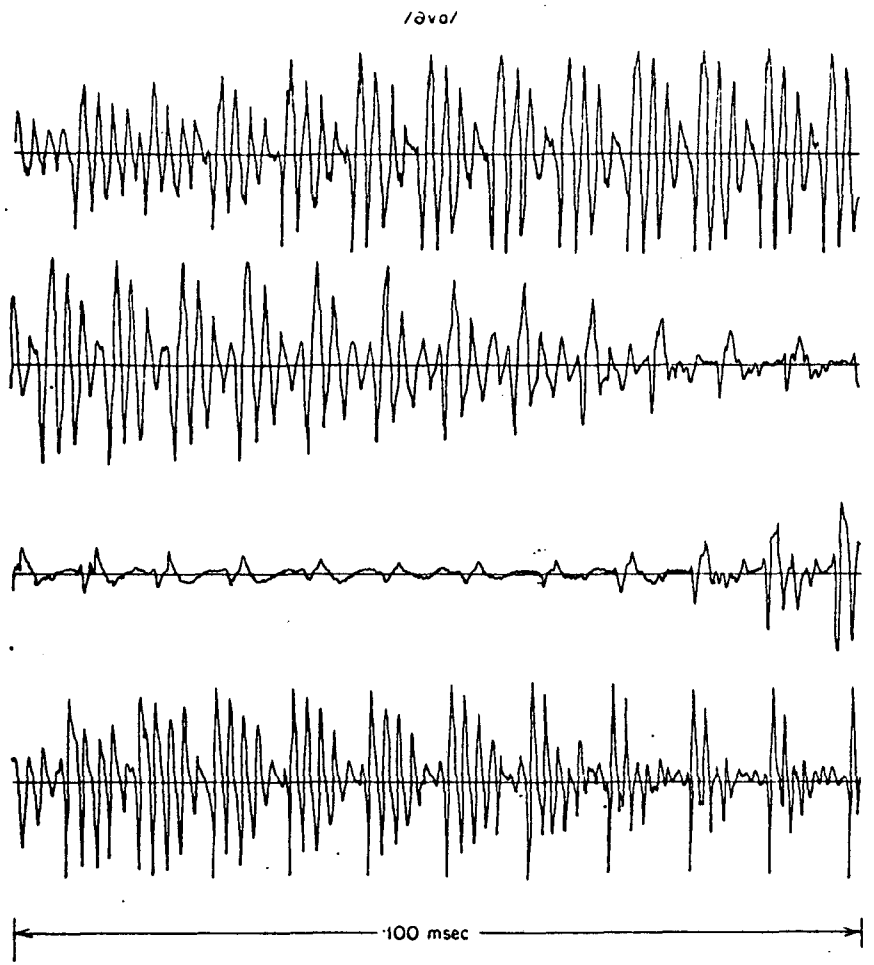


Fig. 3.10 Acoustic waveforms and spectrograms for utterances */UH–V–A/* and */UH–ZH–A/*.