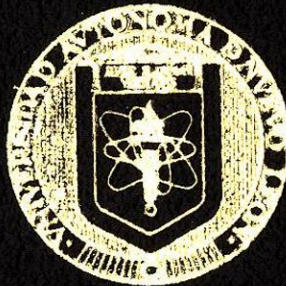


UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO-MATEMATICAS



EL MICROPROCESADOR 68000: SUS AVANCES Y SUS
DESCENDIENTES (68008, 68010, 68012 Y 68020)

TESIS

QUE PARA OBTENER EL TITULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES
PRESENTA

GERARDO CHAVEZ CASTILLO

MONTERREY, N. L.

AGOSTO DE 1988

TL
QA76
.8
.S4
C43
1988
c.1



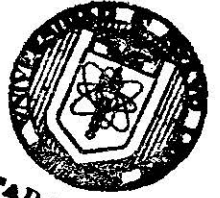
8/sep/88

C.C.

106

cop II.

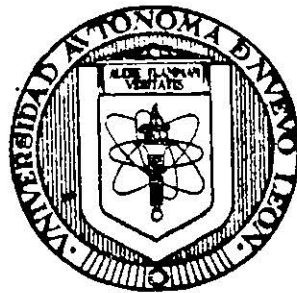
Aut: Ing. Oscar León Cortés
See: Lic. Oscar Aguilar
Voc: Lic. Cesar A. Jiménez.



FACULTAD DE CIENCIAS
UNIVERSIDAD DE CIENCIAS MATEMÁTICAS
BIBLIOTECA

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CIENCIAS FISICO-MATEMATICAS



**EL MICROPROCESADOR 68000: SUS AVANCES Y SUS
DESCENDIENTES (68008, 68010, 68012 Y 68020)**

T E S I S

**QUE PARA OBTENER EL TITULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES
PRESENTA**

GERARDO CHAVEZ CASTILLO

MONTERREY, N. L.

AGOSTO DE 1988

DEDICACIONES

A DIOS :

MUCHAS GRACIAS por darme vida y salud para alcanzar esta meta que era un sueño en mi infancia y hoy es toda una realidad.

A MIS PADRES :

Sr. Rubén Chávez Sillas.

Sra. Jesus Juanita Castillo de Chávez.

GRACIAS a su gran sacrificio y confianza que supieron depositar en mí ya que lo que ahora soy y seré en el futuro se lo debo a ustedes queridos padres.

A MIS HERMANOS :

Ma. ANTONIA

RUBEN

JOSE

SOCORRO

Gracias por su ayuda y cuidados que me
brindaron desde pequeño para alcanzar la
culminación de mi carrera profesional.

A MI ASESOR :

Sr. Ing. Jose Oscar Recio Cantu.

Por su ayuda y orientación desinteresada en
la elaboración de esta tesis.

A MIS MAESTROS :

Les agradezco infinitamente por todos los conocimientos y sabios consejos recibidos de su parte en todas las etapas de mi formación académica.

A MIS AMIGOS :

A todos mis amigos, compañeros y amistades que de una u otra forma ayudaron a alcanzar la meta fijada al comenzar mis estudios.

Al Sr. Ing. Francisco
Delgado Corona

Jefe del Depto. de
Resistencia de
Materiales, F.I.M.E.

Por permitirme usar el equipo de computo de su depto. para la transcripción de esta tesis.

A todos ellos Muchas Gracias.

P R E F A C I O

La realizaci3n de esta tesis es con el fin de que el estudiante de la Licenciatura en Ciencias Computacionales o cualquier persona interesada en el tema tenga conocimiento de los avances en esta area de la ciencia computacional que esta siendo en estos dias, que las PC's tengan capacidades de procesamiento que se igualan a las minicomputadoras, y no se note tanta diferencia entre ambas clases de computadoras.

El tema de esta tesis, abarca nada mas, la familia de Microprocesadores 68000 (incluyendo los modelos 68008, 68010, 68012 y 68020) de la Compañia Motorola.

Para esto el enfoque de esta tesis esta en dar un analisis de los miembros de esta familia en relaci3n con varios factores como son modos que operan, modos de direccionamiento, conjunto de instrucciones que manejan, señales de entrada y salida que manejan (grupos funcionales), procesos excepcionales que soportan, manejan y como lo hacen.

Otros factores que no son incluidos en esta tesis son como su arquitectura interna de cada uno de ellos, sus precios en el mercado actual, que marcas de microcomputadoras manejan estos modelos (o vienen con estos modelos), etc.

En otras palabras, el enfoque que se maneja en esta tesis es sobre los avances que representa la familia 68000 (con todos sus miembros) en su velocidad, su conjunto de modos de direccionamiento asi como de instrucciones, es decir, sus avances internos mäs que externos en lo que a microprocesadores se refiere.

El lector de esta tesis deberä tener conocimientos basicos de computaci3n o electronica o circuitos electricos o electronicos o l3gicos, por ejemplo, estudiantes de carreras a fines a computaci3n o que tengan materias relacionadas con el tema. Lo anterior con el fin de que usted (amigo lector) se le haga mäs sencillo entender la mayoria de los terminos que aqui se manejan.

Si quiere investigar mäs acerca de un sub-tema en particular puede ver la "bibliografia" en donde encontrará los documentos(libros, revistas, etc.) que se utilizaron en la realizaci3n de esta tesis.

Esperando que su inquietud sea cubierta con plena satisfacción y su acervo cultural sea enriquecido y agrandado con la lectura de esta tesis. Pero que su curiosidad no se haya quedado truncada o detenida en la lectura de la misma, ya que este tema es un "mundo" de avances tecnológicos hoy en día.

Gerardo Chávez Castillo.

CONTENIDO GENERAL

DEDICACIONES

PREFACIO

P A R T E I

I N T R O D U C C I O N

| | | |
|-------|---|----|
| 1.1 | Introducción. | 1 |
| 1.2 | Antecedentes de los microprocesadores. | |
| 1.2.1 | Historia de la Computación. | 2 |
| 1.2.2 | Evolución de los Microprocesadores. | 5 |
| 1.3 | Tecnología de fabricación de los Microprocesadores. | 14 |
| 1.4 | Definición del concepto "Microprocesador". | 16 |
| 1.5 | Aplicaciones de los Microprocesadores. | 18 |
| 1.6 | La importancia de los Microprocesadores. | 19 |
| 1.7 | La familia MC68000. | 20 |

P A R T E II

D E S C R I P C I O N F U N C I O N A L

| | | |
|-----|--------------------------------|----|
| 2.1 | Introducción. | 23 |
| 2.2 | Modos de ejecución. | 24 |
| 2.3 | Registros del Modo Usuario. | 25 |
| 2.4 | Registros del Modo Supervisor. | 30 |
| 2.4 | Organización de la Memoria. | 37 |
| 2.5 | Memoria Virtual. | 39 |
| 2.6 | Maquina Virtual. | 40 |

P A R T E I I I

M O D O S D E D I R E C C I O N A M I E N T O

| | | |
|--------|--|----|
| 3.1 | Introducción. | 42 |
| 3.2 | Codificación de direcciones. | 43 |
| 3.3 | Modos de direccionamiento. | 47 |
| 3.3.1 | Directo a registro de datos. | 47 |
| 3.3.2 | Directo a registro de direcciones. | 47 |
| 3.3.3 | Indirecto a registro de dirección. | 47 |
| 3.3.4 | Indirecto con registro de dirección y post-incremento. | 48 |
| 3.3.5 | Indirecto con registro de dirección y pre-decremento. | 48 |
| 3.3.6 | Indirecto con registro de dirección y desplazamiento. | 49 |
| 3.3.7 | Indirecto con reg. de dir. e índice y desplazamiento. | 50 |
| 3.3.8 | Indirecto con memoria y post-índice. | 51 |
| 3.3.9 | Indirecto con memoria y pre-índice. | 52 |
| 3.3.10 | Indirecto con PC y desplazamiento. | 53 |
| 3.3.11 | Indirecto con PC e índice y desplazamiento. | 53 |
| 3.3.12 | Indirecto a memoria con PC y post-índice. | 54 |
| 3.3.13 | Indirecto a memoria con PC y pre-índice. | 55 |
| 3.3.14 | Absoluto corto. | 56 |
| 3.3.15 | Absoluto largo. | 56 |
| 3.3.16 | Dato inmediato. | 56 |
| 3.4 | Categorías. | 57 |

P A R T E I V

R E P E R T O R I O D E I N S T R U C C I O N E S

| | | |
|------|--|----|
| 4.1 | Introducción. | 60 |
| 4.2 | Transferencias de datos. | 61 |
| 4.3 | Aritmética entera. | 62 |
| 4.4 | Aritmética booleana. | 64 |
| 4.5 | Desplazamientos y rotaciones. | 65 |
| 4.6 | Manipulación de bits individuales. | 66 |
| 4.7 | Manipulación de campos de bits. | 67 |
| 4.8 | Manipulación en Decimal Codificado Binario (BCD). | 68 |
| 4.9 | Control del flujo de programa. | 69 |
| 4.10 | Control del sistema. | 70 |
| 4.11 | Comunicaciones Multitarea/Multiprogramación | 71 |
| 4.12 | Precaptación de instrucción/segmentación/ enlazamiento/aceleración. | 72 |

P A R T E V

D E S C R I P C I O N D E S E Ñ A L E S

| | | |
|-----|--|----|
| 5.1 | Introducción. | 75 |
| 5.2 | Señales para el 68000,68008,68010 y 68012. | 76 |
| 5.3 | Señales para el 68020. | 83 |

P A R T E V I

P R O C E S A M I E N T O S E X C E P C I O N A L E S

| | | |
|------|---|-----|
| 6.1 | Introducción. | 91 |
| 6.2 | Modos de Operar. | 92 |
| 6.3 | Tipos de excepciones. | 93 |
| 6.4 | Prioridades de excepción. | 95 |
| 6.5 | Tabla de vectores de excepción. | 97 |
| 6.6 | Estructuras de pila. | 99 |
| 6.7 | Secuencias de procesamiento de excepciones. | 105 |
| 6.8 | Intercepción("trap") de instrucciones. | 105 |
| 6.9 | Instrucciones Ilegales/no existentes. | 106 |
| 6.10 | Error de direccionamiento. | 106 |
| 6.11 | Traza. | 107 |
| 6.12 | Puntos de ruptura. | 107 |
| 6.13 | Error de direccionamiento. | 108 |
| 6.14 | Interrupciones. | 109 |
| 6.15 | Error en bus. | 110 |
| 6.16 | Reinicio("Reset"). | 111 |

P A R T E V I I

M I S C E L A N E A

| | | |
|--------------------------|--|-----|
| 7.1 | Introducción. | 114 |
| 7.2 | Interfaz con la familia 6800. | 115 |
| 7.3 | Interfaz con un coprocesador. | 116 |
| 7.4 | Sus diferencias entre sus modelos. | 118 |
| 7.5 | El acelerador "CACHE" del 68020. | 121 |
| 7.5.1 | Introducción. | 121 |
| 7.5.2 | Como opera el acelerador(cache) del 68020. | 121 |
| 7.5.3 | Control del acelerador. | 122 |
| 7.5.4 | Inhabilitación del acelerador. | 123 |
| C O N C L U S I O N E S. | | 124 |
| B I B L I O G R A F I A. | | 127 |

C O N T E N I D O

P A R T E I

I N T R O D U C C I O N

| | | |
|-------|---|----|
| 1.1 | Introducción. | 1 |
| 1.2 | Antecedentes de los microprocesadores. | |
| 1.2.1 | Historia de la Computación. | 2 |
| 1.2.2 | Evolución de los Microprocesadores. | 5 |
| 1.3 | Tecnología de fabricación de los Microprocesadores. | 14 |
| 1.4 | Definición del concepto "Microprocesador". | 16 |
| 1.5 | Aplicaciones de los Microprocesadores. | 18 |
| 1.6 | La importancia de los Microprocesadores. | 19 |
| 1.7 | La familia MC68000. | 20 |

1.1 INTRODUCCION.

El tema presentado en esta tesis titulada "EL MICROPROCESADOR 68000: Sus avances y sus descendientes (68008, 68010, 68012 y 68020)" tiene el enfoque dirigido a personas que hayan cursado carreras de Ingenieria de Sistemas Computacionales o de Licenciatura, o bien que cursen materias de circuitos logicos, electricos o digitales o bien que conozcan internamente a las microcomputadoras (su hardware) o bien a personas que tengan interes en conocer superficialmente el tema.

En el punto 1.2 se hablara de los antecedentes de los MP en dos grandes partes. La primera de estas partes tratara sobre la "Historia de la Computacion" ya que los acontecimientos de esta area dieron origen a los microprocesadores (MP). La segunda parte de este punto trata sobre la evolucion de los mismos (MP) durante los primeros diez años despues de su nacimiento. Esto tomando como referencia varios MPs de esos años que son considerados como generaciones por sus avances tecnologicos que aportaron.

En el punto 1.3 se hablara de las diferentes tecnologias de fabricacion usadas hasta el momento en los MP.

En el punto 1.4 se definira de diversas formas el concepto de microprocesador segun en punto de vista.

En el punto 1.5 se hablara de la importancia de los Microprocesadores en relacion a la forma de vida que tenemos en estos dias de la decada de los 80's.

En el punto 1.6 se dara una vista general de las diversas aplicaciones en que se emplean los microprocesadores.

En el punto 1.7 se tratara de la familia de MP MC68000, (las siglas MC antes del 68000 significan Motorola Corporation que es la compania que los fabrico y desarrollo, sin embargo a ultimas fechas esta compania a dado permiso a otras de fabricarlos y por eso en algunas publicaciones usted (lector) se encontrara con las siglas M68000) en forma breve, sobre su evolucion, sus caracteristicas generales, etc. En lo que resta de la tesis se hablara mas profundamente de esta familia y sus descendientes.

1.2.-ANTECEDENTES DE LOS MICROPROCESADORES

1.2.1 Historia de la computación.

En seguida se describirá una cronología de los diferentes dispositivos que ha hecho el hombre a través de la historia para realizar cálculos y reducir el esfuerzo necesario para ello.

El primer "dispositivo" que se tiene conocimiento más antiguo es el ABACO creado y utilizado por los CHINOS en 2600 a. de J.C. que consistía en un cuadro de madera con alambres paralelos los cuales contenían bolas móviles o deslizables que servían para enseñar el cálculo.

Se tuvo que transcurrir 40 siglos y en 1642 d. de J.C., un francés llamado Blas Pascal creó y desarrolló una máquina Sumadora que funcionaba mediante ruedas contadoras digitales, como una rueda dentada.

Treinta años después, en 1671 d. de J.C., un Alemán llamado Godofredo Leibniz, desarrolló la máquina calculadora que aplicó el principio de multiplicar mediante la repetición de sumas. Esta máquina funcionaba con un mecanismo casi igual que el del Pascal pero utilizando un cilindro de Leibniz que consiste en un cilindro con ranuras de diferentes longitudes o tamaños lo que permitía realizar las operaciones de suma y multiplicación nada más.

Pasaron casi dos siglos para el siguiente acontecimiento relevante que fue la máquina Analítica del inglés Carlos Babbage en 1833; En la que se aplicó por primera vez el principio de las tarjetas perforadas para resolver un problema matemático complicado. Esta máquina contenía además una unidad de memoria para almacenar información contenida en tarjetas con perforaciones; contaba además con un dispositivo para realizar cálculos aritméticos y una unidad para dirigir las operaciones.

En 1884 d. de J.C. el norteamericano William Seward Burroughs creó una máquina sumadora impresora, la cual fue puesta a la venta en 1891 y tuvo un gran éxito comercial.

En 1887 el Dr. Herman Hollerith creó el sistema electromecánico de tarjetas perforadas. Este sistema se desarrolló para procesar la información del censo de 1890 de Estados Unidos de América. Esto marcó el inicio de las "máquinas" en el procesamiento de datos y no nada más como "calculadoras" para resolver problemas numéricos ya que este sistema podía capturar información como nombre, edad, sexo, profesión, etc. que se requería manejar tanto números como letras y caracteres especiales.

En el decenio de 1935 a 1945, en vista que fué el periodo comprendido por la Segunda Guerra Mundial tuvo un gran auge la creaci6n y desarrollo de "maquinas" capaces de decifrar los codigos secretos de Alemania-nazi, en esto intervinieron los gobiernos britanico y Norteamericano principalmente.

A estos dispositivos se les llam6 computadoras. A continuaci6n una cronica de los acontecimientos mas relevantes de este decenio.

A inicios de 1930, Bajo pedido del ejercito de U.S.A. se desarrollaron las Computadoras de relés de Bell, con la direcci6n de George Stibitz, de Bell telephone Laboratories. Se les llam6 asi debido a que utilizaban relés electromagneticos como componentes operacionales basicos.

En 1937, Howard Aiken cre6 una computadora electromagnetica con el nombre Automatic Sequence-Controlled Calculator llamada MARK-1 utilizaba tambien relés electromagneticos y era capaz de ejecutar una secuencia arbitraria de operaciones aritmeticas controladas por una segunda secuencia de instrucciones codificadas. Esta computadora estuvo en operaci6n hasta 1944.

En 1943 el Dr. John W. Mauchly y J. Presper Eckert, estuvieron a cargo del desarrollo de ENIAC (Electronic Numerical Integrator And Computer) la primera computadora totalmente electronica ya que utilizaba tubos de vacio electronicos en vez de relés magneticos que se usaban sus antecesoras. Su operaci6n requeria una instalaci6n bastante grande ya que pesaba 30 toneladas y utilizaba muchas conecciones de alambres. Esto se fué haciendo muy molesto ya que se necesitabas varias horas para poner a todos los alambres en un lugar determinado para que resolviera un problema 6 programa en escasos minutos.

Hasta estas fechas el problema primordial de estos proyectos era que cada vez que se iba a utilizar tenian que hacer "todo" de nuevo desde el principio. Por lo que los siguientes a6os (1945 a 1950) se desarrollaron computadoras de programas almacenados.

Un ejemplo de lo anterior, en 1946 Von Neumann, el grupo de ENIAC y H. H. Goldstine iniciaron la construccion de una computadora de programas almacenados llamada EDVAC (Electronic Discrete Variable Automatic Computer) se termino en 1952.

En 1951 los se6ores Mauchly y Eckert construyeron la UNIVAC-1 que utilizaba diodos de cristal en lugar de tubos de vacio preludivando la era del estado solido. Era la primera computadora comercial ya que sus antecesoras habian sido desarrolladas por gobiernos con fines belicos, por ejemplo, calcular el angulo de tiro de los ca6ones de largo alcance como tambien desifrar los codigos de se6ales, etc.

La Compañía de maquinas de negocios internacional (IBM) construyó la IBM 701 en 1953 y dos años después en 1955 sacó la IBM 702, ambas utilizaban memoria en un tubo de rayos catódicos. En 1957 a 1959 salió la IBM 705 que utilizaba memoria rápida y confiable basada en núcleos magnéticos. En 1959 salió la IBM 7090 que utilizaba totalmente transistores.

En los años 60's las computadoras fueron construidas con circuitos miniaturizados los cuales eran de tamaño muy reducido lo cual condujo a la creación y construcción de las minicomputadoras. Se les denomina así por su costo y tamaño más bajo que las anteriores y son accesibles a medianas empresas y negocios.

A mediados de los 60's se crearon los circuitos integrados los cuales se les denominó chips y que condujeron al desarrollo de los MICROPROCESADORES que originaron una nueva generación de computadoras que se les llamó microcomputadoras que comenzaron a salir a mediados de los 70's.

La historia de los microprocesadores y su evolución la voy a tratar en el siguiente punto más a fondo y con más detalle. Lo anterior fue expuesto para dar una idea de como fueron surgiendo nuevas y mejores tecnologías de fabricación de las computadoras para mejorar sus limitaciones y así conocer las raíces de los microprocesadores que hoy en día el corazón de una microcomputadora es un Microprocesador.

2.2 EVOLUCION DE LOS MICROPROCESADORES.

Uno de los avances tecnologicos más importantes de las ultimas dos decadas fue la aparición de los circuitos de integración a gran escala (LSI:Large-scale integrated). En el dominio de la logica digital, esta evolución pasó por las etapas de producción de subunidades lógicas estándar en paquetes de circuito integrado (IC).

En la primera etapa se lograron compuertas y flip-flops simples en chips con integración a pequeña escala (SSI:Small Scale Integrater). Esto fue seguido por chips con integración a mediana escala (MSI) que tenían registros, contadores, codificadores, decodificadores, etc.

Conforme aumentó la capacidad para construir IC con alta densidad de elementos logicos, se volvió una ventaja tomar en cuenta circuitos que requerian muchos elementos, pero relativamente pocas conexiones externas. El resultado fué la aparición de paquetes mas complejos, tales como los chips de ALU (Unidad Aritmetica Logica), capaces de efectuar las funciones aritmeticas y logicas usuales, con operandos de 4-bits.

La continuación logica de esta tendencia propicio la aparición de procesadores completos en un solo chip, a principios de la decada de los setentas. Primero, hubo chips capaces de operar con 4-bits en paralelo. Ya que tenían capacidades de procesamiento, pero no tamaño ni velocidad comparable con los de las minicomputadoras, resultó razonable denominarlos a los nuevos dispositivos MICROPROCESADORES.

Muy poco después hubieron chips mas grandes. En este momento se fabrican una variedad de chips microprocesadores que operan en paralelo con datos de 8 bits ó 16 bits. Además la tecnologia que mejora cada vez más ha llegado al punto de fabricar circuitos con integración a muy gran escala (VLSI : Very-Large-Scale Integrated) para implantar un microprocesador de 32 bits en un solo chip.

Al aumentar la capacidad de procesamiento de los chips de IC, se tiende a incrementar el numero necesario de conexiones externas, lo que da como resultado paquetes más grandes.

Después de la introducción del primer microprocesador en 1971, la ejecución del microprocesador ha sido incrementada impresionantemente. Mejorando la ejecución en tres maneras:

- 1.- Mas tecnologías avanzadas han sido envueltas en la fabricación de microprocesadores.
- 2.- El diseño logico detallado o arquitectura del microprocesador ha mejorado.

3.- Más memoria e interfaces de Entrada/Salida han sido incluidas en el mismo chip del microprocesador. Así reduciendo el número requerido de componentes externos.

En breve, el microprocesador de hoy en día es un dispositivo con solamente entradas, salidas y un programa residente. Pero haciendo un poco de historia se dará la cronología de los primeros 10 años del desarrollo de los microprocesadores tomando de ejemplo los más relevantes en ese período.

I N T E L 8 0 0 8

Es un microprocesador de 8-bits, la primera-generación que fué introducido en enero de 1972. Esto es actualmente el segundo de la primera-generación de procesadores. Siguiendo el 4004, un procesador de 4-bit introducido en Noviembre de 1971. El 8008 este fabricado usando el proceso P-channel MOS, y empaquetado en un dual in-line de 18-pin(DIP).

En la siguiente figura presenta un resumen de los pins de salida del 8008, es decir, las patas de salida.

F I G U R A 1

| Direcciones | Datos | Bus de Control | Poder de aplicacion |
|---|---------------------|--------------------------------|------------------------|
| | | Input | Output |
| 14-bits multiplexar sobre el bus de datos | 8-bit bidireccional | interrupt Ready 01 02 | Syne S0 S1 S2 |
| | | | +5 -9 |

I N T E L 8 0 8 0

Fue el primero de la segunda-generación de microprocesadores y fue introducido en noviembre de 1973. Hoy, es uno de los mas ampliamente usados en el mundo. El 8080 esta fabricado usando el proceso N-channel MOS. Este esta empaquetado en uno de 40-pin DIP.

La siguiente figura es un resume de los pins de salida del 8080.

F I G U R A 2

| Direcciones | Datos | Bus de Control | | Fuente de poder |
|-------------|-------------------------|----------------|--------|-----------------|
| 16 lineas | 8-bits bidireccional | Input | Output | |
| | | O1 | DBIN | +5 |
| | | O2 | INTE | +12 |
| | | HOLD | SYNC | -5 |
| | | READY | WR | |
| | | INT | WAIT | |
| | | RESET | HLDA | |

Z 8 0

El primero de la tercera-generación de microprocesadores es el Z80, introducido en abril de 1976. Usando carga modo-depleción, fabricación N-channel MOS. Solamente un poder de +5-V (volts) es requerido por el microprocesador. Solamente un signo de reloj externo de +5-V es requerido. Los signos de salida de control adicional son generadas por el Z80 lo cual elimina la necesidad para multiplexar status información sobre el bus de datos, como lo hace con el 8080. El Z80 esta contenido en el mismo estilo de 40-pin DIP que es usado para el 8080.

La siguiente figura es el resumen de los pins de salida del Z80.

F I G U R A 3

| Direcciones | Datos | Bus de Control | | Fuente de poder |
|-------------|------------------------|----------------|--------|-----------------|
| 16 líneas | 8 bit bidireccional | Input | Output | |
| | | CLOCK | HALT | |
| | | INT | MREG | |
| | | NMI | IORQ | |
| | | WAIT | RD | |
| | | BUSRQ | WR | |
| | | RESET | BUSAK | |
| | | | M1 | |
| | | | RFSH | |

E L 8 7 4 8

Un paso dramático en la evolución de microprocesadores ocurrió con la introducción del 8748 en febrero de 1977. La característica muy marcada de este microprocesador es que contenía 1024 bytes de memoria programable sobre el mismo chip. El 8748 es empacado en un convencional chip de 40-pin DIP, pero es cubierto con un líquido de cuarzo transparente. Un usuario puede iluminar con luz ultravioleta intensa a través de este líquido para borrar el programa almacenado, y subsecuentemente reprogramar la memoria de el microprocesador.

El 8748 esta construido usando circuiteria N-channel MOS.

La siguiente figura muestra los miembros de la familia de Microprocesadores 8748.

F I G U R A 4

| | 8021 | 8035 | 8039 | 8048 | 8049 | 8748 |
|------------------------------------|-------------|-------------|------|-------|-------|-------------|
| Memoria de Prog. residente | 1kROM | --ninguno-- | | 1kROM | 2kROM | 1kEPROM |
| Memoria residente en RAM(bytes) | 64 | 64 | 128 | 64 | 128 | 64 |
| N's de lineas de I/O | 21 | 27 | 27 | 27 | 27 | 27 |
| N's de pins IC | 28 | 40 | 40 | 40 | 40 | 40 |
| | Menor costo | -----> | | | | Mayor costo |

I N T E L 8 0 8 6

Fue introducido en junio de 1978, es el primero de una generaci3n de microprocesadores de 16-bits de alta-ejecuci3n. El 8086 viene en un chip de 40-pin DIP y esta fabricado usando una alta-velocidad N-channel MOS proceso bajo el nombre de HMOS. El 8086 puede direccionar directamente 1 megabyte (1024 kbytes) de memoria externa.

La siguiente figura es el resumen de pins de salida del 8086.

F I G U R A 5

| Direcciones | Datos | Bus de Control | Fuente de Poder |
|--------------|----------|----------------|-----------------|
| 20-bits | 16-bits | Input | 5V |
| multiplexed | bidirec- | NMI | SO-S7 |
| sobre el | | INTR | BHE |
| bus de datos | | CLK | RD |
| y el bus de | | READY | QSO |
| control | | RESET | QS1 |
| | | TEST | HLDA |
| | | MN/MX | GT0 |
| | | RQ | GT1 |

E L Z 8 0 0 0

El Z8000 es un procesador de 16-bits. Está disponible en chips de 40-pin b 48-pin. El de 40-pin puede direccionar a 64k de memoria; el modelo de 48-pin puede direccionar a 8 Megabytes de memoria.

La siguiente figura muestra el resumen de los pins de salida del Z8000.

F I G U R A 6

| Direcciones | Datos | Bus de Control | Poder de aplicación |
|---|----------------------|----------------|---------------------|
| 16-bits en la versión de 40-pins multiplexed sobre el bus de datos y 23-bits en la versión de 48-pins | 16-bit bidireccional | Input | Output |
| | | WAIT | ST0-ST3 |
| | | STOP | AS |
| | | BUSRQ | DS |
| | | NMI | MREQ |
| | | VI | READ/WRITE |
| | | NVI | NORMAL/SYSTEM |
| | | MI | WORD/BYTE |
| | | SEGT | BUSAK |
| | | RESET | M0 |
| | | CLK | |
| | | | 5V |

Ahora teniendo en cuenta que el diseño del 8008 empezó en 1969 y que el Z8000 fue introducido en el año de 1979 podemos hacer una comparación de la ejecución (performance) de los microprocesadores representativos sobre 10 años de evolución.

| | 8008 | 8080 | Z80 | 8748 | 8086 | Z8000 |
|--|--------------|----------|------|------|-------------|-------|
| Año de introducción | 1972 | 1973 | 1976 | 1977 | 1978 | 1979 |
| Patrón de datos (bits) | 8 | 8 | 8 | 8 | 16 | 16 |
| Espacio de memoria direccionable directamente. (Bytes) | 16k | 64k | 64k | 4k | 1024k | 8192k |
| Voltage requerido | | +12 | | | | |
| Poder de aplicación. | +5 -9 | +5 -5 | +5 | +5 | +5 | +5 |
| Memoria de prog. residente. | -- Ninguno-- | | | 1k | --Ninguno-- | |
| Throughput relativo estimado (8008=1). | 1 | 10 | 20 | 20 | 100 | 100 |

La evolución de los microprocesadores tiene lugar en una secuencia de generaciones. Cada generación sucesiva ha producido microprocesadores con más capacidad en los siguientes términos:

- 1.- Velocidad.
- 2.- Poder del conjunto de instrucciones.
- 3.- Facilidad de uso.

1.3.- TECNOLOGIA DE FABRICACION DE LOS MICROPROCESADORES.

Los microprocesadores a bajo-costo estan disponibles hoy en dia solamente porque los tremendos avances de fabricacion en la tecnologia de circuitos-integrados(IC) en la antepasada decada (a finales de los 60's). Los procesos de integracion permitieron la fabricacion economica de circuitos complejos, consistiendo de miles de transistores sobre pequeños chips de silicon. Un circuito microprocesador completo puede ser fabricado sobre un chip de silicon que es solamente de unos pocos milímetros de lado.

La idea de usar la tecnologia de circuitos-integrados para construir microprocesadores fue primero concebida por el Dr. Ted Hoff en la Intel Corporation en 1969. El Dr. Federico Faggin entonces dirigió el diseño en Intel para desarrollar el primer microprocesador del mundo. Despues de la introduccion del primer microprocesador en 1971. Los avances en la tecnologia de los circuitos-integrados tienen a su vez más avances en microprocesadores.

La tecnologia MOS de canal P fue el primer proceso de fabricacion de los circuitos de LSI (Larga escala de integracion). El MOS de canal N es el más utilizado en la actualidad para aplicaciones de velocidad baja y mediana. Permite una densidad de elementos logicos en el chip.

La tecnologia CMOS (MOS complementario) se utiliza para microprocesadores que tienen bajo consumo de energia y considerable flexibilidad en cuanto a necesidades del suministro de potencia.

Se han fabricado chips de mejor rendimiento con los procesos TTL (Logica de Transistor a Transistor), en particular el proceso Schottky de baja potencia.

Es posible producir chips de muy buen desempeño con la tecnologia de ECL (Logica de Acoplamiento por Emisor). Sin embargo el alto rendimiento de los circuitos de ECL y TTL va a la par con un mayor consumo de potencia y una menor complejidad de los circuitos logicos de los chips.

La tecnologia de I²L (Logica de Inyeccion Integrada) permite alta densidad de empacada así como baja potencia, y ofrece un producto de costo comparativamente más bajo.

Las 6 tecnologías de circuitos-Integrados usadas en la fabricación de microprocesadores son:

- 1.-E C L bipolar (Emitter-coupled logic).
- 2.-Schottky bipolar.
- 3.-I²L bipolar (Integrated injection logic).
- 4.-Complementary MOS (METal-oxide silicon).
- 5.-N-channel MOS.
- 6.-P-channel MOS.

Un resumen de las 6 mejores tecnologías usadas en la fabricación de microprocesadores esta dada en la siguiente figura con ejemplos representativos de cada uno de ellos.

| Bipolar | | | MOS | | | |
|-------------------|--|---|------------------|-------------------------------|---------------------------------|-------------------------|
| Unsaturated | | saturated | | | | |
| ECL | Schottky | I ² L | CMOS | | N-channel | P-channel |
| Motorola 10800 | Intel 3001 | Texas Instruments SBP0400 SBP9900 | Intersil 6100 | RCA | Intel 8080,8085 8748,8086 | Intel 4004 4040 |
| | Advanced Micro Device Am 2901 | | COSMAC | Motorola 6800,6802 6809 | Fairchild F8 | National 8008 IMP |
| | | | | | MOS MCS 6502 | PACE |
| | | | | | Zilog Z-80,Z-8 Z-8000 | |

1.4.- DEFINICION DEL CONCEPTO "MICROPROCESADOR".

En este punto se trataran varias preguntas que se plantean y a su vez varias respuestas de las mismas. El fin de este punto es dar una amplia gama de opiniones de este tema.

A continuaci3n dar3 varias definiciones de la palabra "Microprocesador" que se abrevia "MP" pero tal vez en algunos libros lo vaya a encontrar asi "MPU" ya que algunos autores hacen referencia a una Unidad de Microprocesador.

Microprocesador: Una pieza del hardware que almacena las partes de c3mputo de una computadora en una tablilla de circuito 3 en un conjunto de Circuitos Integrados. Es una parte integral de una microcomputadora; pero no contiene las interfaces de entrada y salida (I/O), ni la unidad de memoria. Se les ha llamado a los microprocesadores "computadoras de una oblea".

Microprocesador: Cualquier unidad de procesamiento central (CPU) que este contenida en un solo chip.

Ahora tomando como referencia la historia de los microprocesadores a estos se les dio este nombre por ser de tama1o mucho menor a los procesadores normales de sus antecesoras que son las minicomputadoras, y tambien de menor costo, pero de iguales capacidades de procesamiento.

Microprocesador: Cuando la capacidad de integraci3n de los fabricantes de componentes alcanz3 el grado de perfecci3n superficie para colocar en un 3nico chip semiconductor todos los elementos de una Unidad Central de Proceso de un computador naci3 un nuevo bloque funcional: el microprocesador.

Microprocesador: Es la parte del control y procesamiento de una peque1a computadora o microcomputadora, que puede ser construido con circuiteria LSI MOS usualmente sobre un solo chip. Como en los procesadores de todas las computadoras, los MP's pueden manejar ambos datos l3gicos y aritmeticos en una forma de bit-paralelo bajo el control de un programa. Los MP's son distinguidos de los procesadores de las minicomputadoras por su uso de LSI por su baja fuente de poder (150-200 Watt) y su bajo costo y de otros dispositivos LSI (a excepci3n de chips calculadoras) por su comportamiento programable.

En la actualidad hablar de la capacidad de una microcomputadora se hace referencia a su microprocesador implicitamente ya que es el "coraz3n" de la misma.

Que elementos basicos constituyen al MP ?

- R: a) La unidad de control : que interpreta las instrucciones y genera las señales adecuadas para su ejecución.
- b) El Registro de Instrucción : Que memoriza temporalmente la instrucción mientras se ejecuta.
- c) Contador de Programa : Que memoriza la dirección de la próxima instrucción a ejecutar.
- d) Acumulador : Que memoriza uno de los operandos mientras se ejecutan las instrucciones operativas.
- e) Unidad Aritmética y Lógica : Que realiza las operaciones aritmeticas y logicas bajo la supervisión de la unidad de control.

P: Quién inventó el MP ?

R: La corporación Intel tiene el crédito por la exploración del concepto de MP y fue el primer MP en salir al mercado, pero mucho crédito se debe también a muchas compañías y personas quienes contribuyeron en alguna manera al desarrollo de la integración a larga escala (Large-Scale Integration [LSI]).

P: Algunas ventajas de un microprocesador ?

- R: 1.- La reducción del costo y complejidad del hardware por el remplazo de diseños lógicos-random existentes por pocas partes (MP's).
- 2.- Suma de la capacidad aritmética o de cálculo (computacional) no disponible con lógicos-random.
- 3.- El logro de un "pequeño" instrumento que puede ejecutar una secuencia de instrucciones bajo el control de un programa y posiblemente para controlar o interactuar con otros instrumentos.
- 4.- Hay el potencial para extensiones futuras de las técnicas de diseño para otras aplicaciones.

1.5.-APLICACIONES DE LOS MICROPROCESADORES.

En este punto vamos a ver las areas en donde tienen aplicaci3n los microprocesadores desde sus inicios hasta nuestros dias.

1.- Su impacto inicial se present3 en las calculadoras.-Hoy en dia, el coraz3n de una calculadora de bolsillo es un Microprocesador.

2.- Los microprocesadores tambi3n se incluyen l3gicamente en el equipo estandar de los sistemas de computaci3n. Se emplean en los controladores de discos magneticos, asi como en las terminales "inteligentes". Los sistemas de adquisici3n de datos pueden simplificarse usando microprocesadores, en puntos remotos, como dispositivos de recolecci3n de datos. Esto permite algun procesamiento previo al envio de datos a la computadora principal.

3.-La mayoria de las maquinas que requieren amplios circuitos de control. Su empleo incluye las terminales que se usan en los lugares de venta y de negocios en general.

4.-En el hogar los microprocesadores pueden controlar dispositivos complejos, sistemas de seguridad, de temperatura y de iluminaci3n, adem3s de proporcionar entretenimiento con los juegos de participaci3n de tiempo real.

5.-En las comunicaciones. Un ejemplo son los conmutadores telefonicos privados; denominados PABX(private automatic branch-exchange).Es posible controlar un numero considerable de lineas telefonicas (200 b m3s) usando un solo microprocesador.-Tales sistemas pueden proporcionar numerosas características operativas que ser3a dif3cil implantar sin el control de una computadora.

6.-En la Instrumentaci3n.-Los circuitos de control electronico pueden remplazarse por las instrucciones de una "memoria de control", que ejecuta un microprocesador que genera las Se3ales que, de otra forma, producir3an los circuitos de control.-Esta estrategia da como resultado reducciones importantes de costos.-La memoria de control consistir3a mas que nada en chips de ROM.-Las ventajas a considerar son muchas.-Ademas del costo mas bajo; el proceso de dise3o se simplifica, ya que hay menos componentes.-Es posible corregir muchos errores de dise3o y agregar nuevas opciones, simplemente con cambiar el contenido de la memoria de control.-El problema posible solamente surge cuando la velocidad de operaci3n es un factor critico.-Ya que la velocidad de operaci3n de la memoria de control es mucho menor que la de los circuitos de control pero aplicando adecuadamente la memoria de control se minimiza este posible problema.

1.6.- LA IMPORTANCIA DE LOS MICROPROCESADORES.

Como podremos ver, si recordamos el punto 1.2 veremos que desde el primer "dispositivo" creado por el hombre hasta el segundo y así sucesivamente hasta llegar al siglo 1900 ha sido impresionante la forma en que se ha desarrollado la tecnología en el tramo de 1930 a la fecha (1988) en la rama de las "maquinas" electronicas para realizar o resolver problemas de calculo.

Con el desarrollo de lenguajes de alto nivel que se podrian entender facilmente por una persona que no supiera nada del lenguaje de una maquina en particular se empezò a desarrollar una cantidad grande de lenguajes de alto nivel para cada area en particular donde se utiliza a la microcomputadora como una herramienta en "todo" ya que en la actualidad hay paquetes tanto en la administraciòn de una empresa o negocio, como en el proceso de fabricaciòn en una empresa de la industria,etc.

Desde la decada pasada -1970- con la fabricaciòn de las microcomputadoras se abriò un gran campo de acciòn de aplicaciòn ya que cualquier tipo de empresa tenia la facilidad de adquirir una.

Hoy en dia, en esta decada de los 80's por terminar, en cualquier actividad que realicemos nos encontraremos con que interviene el "trabajo" de un microprocesador aunque no lo veamos, ni nos demos cuenta de ello. Por señalar varios ejemplos darè los siguientes:

en el hogar:

- en el despertador (reloj digital).
- en el control de temperatura (aire acondicionado).
- en el sistema de seguridad,puertas electronicas,etc.
- en los juegos de video para niños (o grandes).
- etc.

en el trayecto de la casa al trabajo:

- en el automovil (puertas,tablero,vidrios electronicos).
- en los semaforos al cruzar las calles o avenidas.

En la actualidad, cualquier aparato de control en automoviles,aviones,barcos principalmente en naves espaciales se utiliza los microprocesadores ya que vivimos en una epoca en que todo aplican su uso ya sea en hospitales modernos casi todos los aparatos son electronicos, para sacar diagnosticos, analisis,etc.

De ahí su importancia es muy grande dada su amplia y diversa aplicaciòn en estos dias.

1.7.- LA FAMILIA MC68000.

SU EVOLUCION.

El primer miembro de esta familia fue el microprocesador MC68000 introducido en el año de 1979, con 8Mhz, de 16-bits externos y 32-bits internos. Con 17 registros de propósito general y registros de dirección y datos de 32-bits. El MC68000 procesa en una estructura de bus asincrónico con bus de datos de 16-bits y bus de direcciones de 32-bits no-multiplexados. Está provisto para un rango de direccionamiento directo de 16 megabytes.

En el año de 1981 salió el MC68000 con 12.5 Mhz y con la misma compatibilidad de su antecesor de 8 Mhz.

En el año de 1982 Motorola lanzó al mercado el MC68008 de 8 Mhz, una versión de 8-bits del MC68000, como el anterior totalmente compatible en código objeto, permitiendo desarrollar software para ser corrido en ambos MP's. Con un megabyte de espacio de direcciones lineal no-segmentadas, programas modulares largos pueden ser desarrollados y ejecutados eficientemente sin las limitaciones de ejecución (performance) forzadas por la segmentación de arquitectura de otros, MP's de 8-bits que no son de Motorola.

En el año de 1983, Motorola lanzó esta vez el MC68010 de 12.5 Mhz que era la versión con memoria virtual del microprocesador MC68000 y como es de suponerse también altamente compatible en código objeto con sus miembros de su familia anteriores, por su arquitectura interna, conjunto de instrucciones y su estructura de bus, ya que son idénticas al MC68000. En adición de todos los recursos del MC68000, el MC68010 ofrece alta velocidad en modo de operación lo cual permite una rápida ejecución de ciclos de software (programas). La característica del soporte de la memoria virtual y el tiempo de ejecución de instrucciones hacen del MC68010 uno de los más rápidos y versátiles MP's en el mercado.

A mediados de ese año -1983- Motorola lanzó el MC68012 de 12.5 Mhz que era la versión de memoria virtual extendida del MC68000. El MC68012 provee todos los recursos del MC68010 y mantiene la alta compatibilidad de la familia MC68000. Las capacidades de la memoria virtual extendida son mejoradas por los recursos de direccionamiento adicionales los cuales permiten tener un rango de dos-gigabytes. Los recursos adicionales mejoran la operación multiprocesador.

Y a finales del año de 1984 lanzó el MC68020 de 12.5 Mhz, un microprocesador de 32-bits con todos los recursos de sus descendientes (memoria virtual, memoria virtual extendida, etc.). Tiene buses de datos y direcciones de 32-bits no-multiplexados con técnicas innovadoras para mejorar la eficiencia del bus, un conjunto de instrucciones y instrucción cache (acelerador) para

alta velocidad ventajas de eficiencia, y nuevas instrucciones y modos de direccionamiento para mejorar el soporte del sistema operativo. En adición, el MC68020 es el primer MP que tiene una interface de coprocesador la cual extiende las capacidades de un sistema M68000 bajo los límites de cualquier simple elemento procesador.

A principios del 86' salió el MC68020 de 16.67 Mhz, un microprocesador de 32-bits.

A finales de 1986 salió el MC68020 de 20 Mhz.

A principios de 1987 se lanzó el MC68020 de 25 Mhz.

Y esta en espera que para principios de 1988 este listo el MC68030.

LAS CARACTERISTICAS DE CADA MIEMBRO.

En la siguiente figura se muestran algunas de las características más generales de cada miembro de la familia de los microprocesadores 68000.

M I C R O P R O C E S A D O R E S

| NUMERO | DESCRIPCION | VELOCIDADES |
|--------|---|-----------------------|
| 68000 | 16/32-Bit Microprocesador. | 8,10,12.5 Mhz |
| 68008 | Version de bus reducido de 8-bit. | 8,10, Mhz |
| 68010 | MPU de 16-bit de Memoria Virtual. | 8,10,12.5 Mhz |
| 68012 | MPU de 16-bit de Memoria Virtual Expandida. | 8,10,12.5 Mhz |
| 68020 | 32/32-Bit Microprocesador. | 12.5,16.67,20, 25 Mhz |

C O N T E N I D O

P A R T E I I

D E S C R I P C I O N F U N C I O N A L

| | | |
|-----|--------------------------------|----|
| 2.1 | Introducción. | 23 |
| 2.2 | Modos de ejecución. | 24 |
| 2.3 | Registros del Modo Usuario. | 25 |
| 2.4 | Registros del Modo Supervisor. | 30 |
| 2.5 | Organización de la Memoria. | 37 |
| 2.6 | Memoria Virtual. | 39 |
| 2.7 | Maquina Vrtual. | 40 |

2.1 INTRODUCCION.

Esta parte presenta una descripción de la familia 68000 desde el punto de vista de su programación, y en términos de sus modos de ejecución, registros del procesador. Lo anterior expuesto en los puntos 2.2,2.3,2.4.

En el punto 2.5 se habla de la organización de la memoria. Como está constituida,etc.

En los puntos 2.6 y 2.7 se hablará de la memoria virtual y de las maquinas virtuales. Que son los dos temas más recientemente explotados por los microprocesadores.

La similitud entre los distintos miembros de la familia 68000 pesan más que sus diferencias. Sin embargo, los miembros más recientes de la familia naturalmente tienen ciertas características no presentes en los miembros iniciales. Cuando se describan las propiedades comunes a todos los miembros de la familia, usará el término "68K". Cuando se describan una característica no encontrada en todos los elementos, especificaré el nombre del modelo o modelos; por ejemplo, "68010" se refiere al tercer miembro de la serie.

2.2 MODOS DE EJECUCION.

La familia 68000 puede operar en uno de dos modos, el modo "usuario" o el modo "supervisor". Un bit del registro de estado llamado "bit modo usuario" determina el modo de funcionamiento del procesador. Como indica su nombre, el modo usuario es para cuando el procesador ejecuta operaciones a nivel de aplicaciones, mientras que el modo supervisor está proyectado para programas a nivel del sistema operativo.

El modo supervisor tiene su propio apuntador de la pila, así como instrucciones adicionales, privilegiadas, que un programa ejecutándose en modo usuario no puede utilizar.

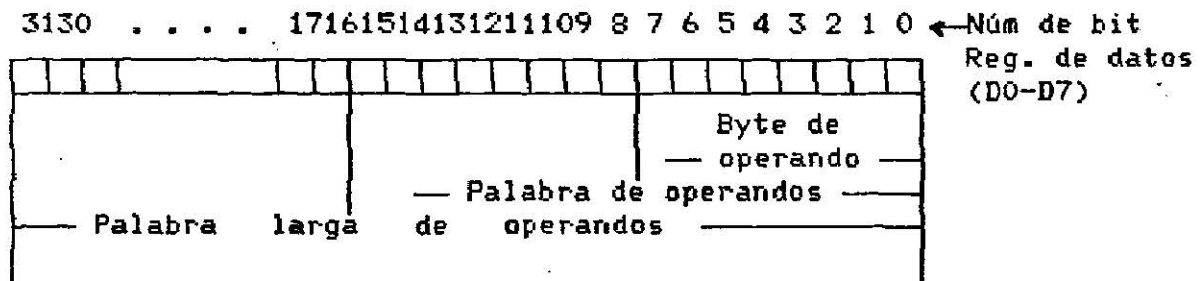
El circuito integrado del procesador proporciona en sus terminales de salida una señal codificada que representa el código función (FC0-FC2) que indica el modo de ejecución en curso, así como la actividad actual del bus (acceso a datos, acceso a programa o reconocimiento de interrupción). La lógica externa como puede ser una unidad de manejo de memoria, puede hacer uso de la información del modo de ejecución que aparece en esas líneas para diferenciar la memoria del sistema de la memoria del usuario.

2.3 REGISTROS DEL MODO USUARIO.

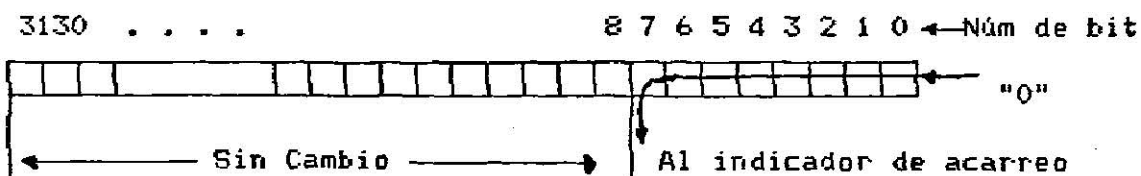
En el modo usuario, la familia 68000 tiene ocho registros (D0-D7) de 32 bits, siete registros de dirección (A0-A6) de 32-bits, un apuntador de pila (SP) de 32 bits, un contador de programa de 32 bits y un registro de código de condición de 8 bits.

Los registros de datos pueden usarse como datos de un solo bit, datos de 8 bits (bytes), datos de 16 bits (palabra) y datos de 32-bits (palabras largas). El bit menos significativo le denominaremos bit 0, y al más significativo bit 31. La siguiente ilustración muestra cómo se posicionan dentro de los registros los distintos tamaños de operandos.

Las operaciones con un único bit son para comprobación, puesta a uno y puesta a cero de bits individuales internos a un byte, palabra o palabra larga. Las instrucciones de manipulación de bits generalmente suministran el número de bit, ya sea directamente o a través de un registro de datos.



Los operandos byte ocupan los bits 0 al 7, los operandos palabra ocupan de los bits 0 al 15, y las palabras largas utilizan el registro completo, bits del 0 al 31. Cuando una instrucción utiliza un registro de datos como operando fuente o destino, sólo la porción correspondiente del registro es alterada -los bits de mayor orden del registro permanecen sin cambio-. Por ejemplo, una operación aritmética de un tamaño de un byte, de desplazamiento a izquierda (ASL.B) desplaza sólo los bits del 0 al 7, como se muestra en la siguiente ilustración. Los bits 8 al 31 permanecen como estaban anteriormente.



Además de su función como fuente o destino de operandos en una operación, los registros de datos pueden funcionar como registros índices o contadores en buses.

El 68020 permite tamaños de datos de "palabras cuádruples". Las palabras cuádruples son de 64 bits de longitud y son usadas sólo por las instrucciones de multiplicar y dividir con 32 bits (almacenando el producto de dos palabras largas o manteniendo el dividendo de un divisor de 32 bits). Las palabras cuádruples pueden residir en dos cualesquiera de los registros de datos sin restricción en el orden o emparejamiento, el formato de las instrucciones de multiplicación y división larga especifica los registros tanto alto (bits 32-63) como bajo (bits 0-31).

El 68020 también permite "campos de bit". Esto hace que un programa pueda acceder hasta 32 bits consecutivos que no estén necesariamente confinados en un byte, palabra o palabra larga.

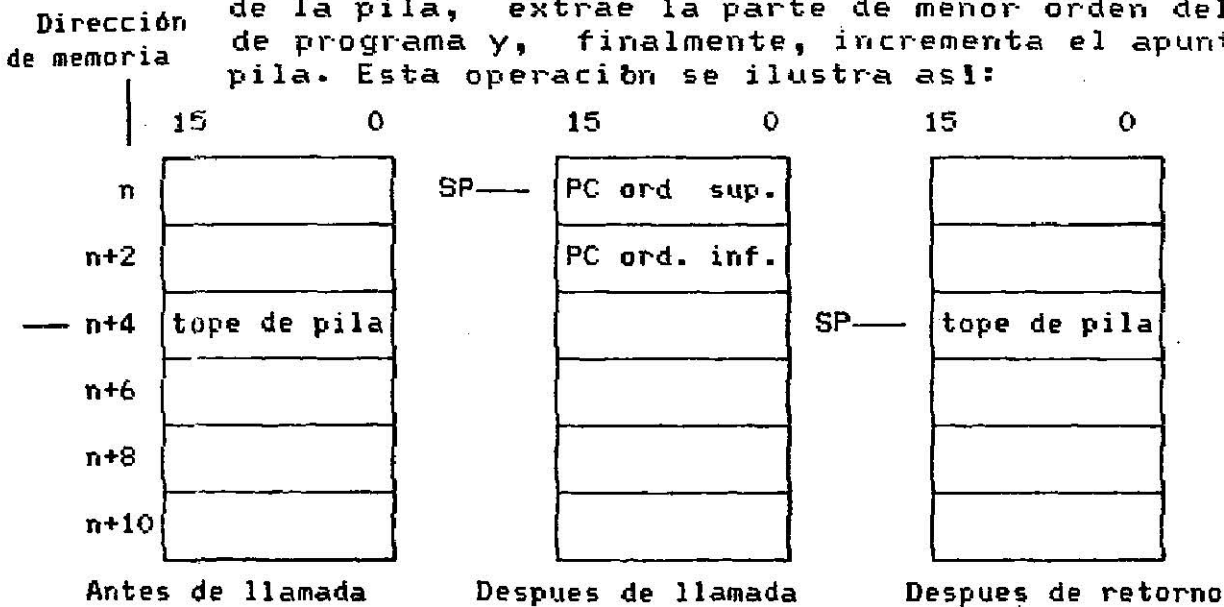
La familia 68000 tiene siete registros de dirección de uso general, A0-A6. Dependiendo del modo de direccionamiento, estos registros pueden mantener direcciones de operandos, direcciones de apuntadores a operandos, direcciones base e índices. Los registros de dirección pueden mantener datos de 16 bits o de 32 bits. No hay operaciones directas con bytes en los registros de dirección.

Cuando se utilicen directamente como operandos fuente, los valores de 16 bits son ampliados en signo antes de usarlos (la palabra de mayor orden no afecta ni es afectada por la operación). Cuando se usan directamente como destino de operandos, las fuentes de 16 bits están ampliadas en el signo y rellenan completamente los 32 bits del registro de dirección de destino.

El procesador utiliza un octavo registro de direcciones, A7, como apuntador de pila. Como se afirmó antes, el procesador tiene dos apuntadores de pila distintos, dependiendo del modo de procesamiento en curso. Estos registros (A7 y A7') son comúnmente más conocidos como APUNTADOR DE LA PILA DEL USUARIO (USP) y APUNTADOR DE LA PILA DEL SISTEMA (SSP). El procesador conoce implícitamente que el apuntador de pila debe utilizarse para insertar en la pila operaciones basadas en el modo de ejecución en curso. El 68020 tiene dos versiones distintas del SSP, como se discutirá más adelante.

El microprocesador rellena una pila de posiciones altas a bajas de memoria pre-decrementando o post-incrementando los modos de direccionamiento indirecto para insertar y extraer, respectivamente. Por ejemplo, en una llamada a subrutina, el microprocesador decrementa el apuntador de la pila, inserta la porción de menor orden del contador de programa, decrementa de nuevo el apuntador de la pila y, finalmente, inserta la porción de mayor orden del contador del programa.

Cuando se regresa ("return") de una subrutina el microprocesador extrae la parte de mayor orden del contador de programa, incrementa el apuntador de la pila, extrae la parte de menor orden del contador de programa y, finalmente, incrementa el apuntador de la pila. Esta operación se ilustra así:



Como la pila debe procesar llamadas a subrutinas y excepciones, ambas de las cuales insertar y extraer los valores del contador de programa en forma de palabras largas, la pila debe mantener un alineamiento par (de palabras). El microprocesador utiliza el byte superior de la palabra que esta en la pila cuando inserta o extrae bytes; el byte de menor orden no se usa. Es decir, se cumple el requerimiento de frontera para la pila.

El contador de programa es un registro de uso especial de 32 bits. Apunta a la primera palabra de la siguiente instrucción a ser ejecutada (entre instrucciones) o a la siguiente palabra de una instrucción (durante la captación de instrucción). Como las instrucciones deben estar alineadas en direcciones frontera pares (palabras), el contador de programa debe contener siempre un valor par.

En el modo usuario, el microprocesador tiene acceso al byte de menor orden del registro de estado. Este byte (el registro de código de condición o CCR) mantiene bits indicadores cuyos valores dependen del resultado de ciertas operaciones, tales como suma, resta, desplazamiento y otras.

El bit de "acarreo" ("carry" o C) se pone a uno cuando hay una salida de acarreo en el bit más significativo como consecuencia de una operación suma o si un dígito es requerido en una resta.

El bit de "desbordamiento" ("overflow" o V) es el valor exclusive-or entre el bit de salida de acarreo más significativo y el siguiente bit de orden superior del operando como resultado de operaciones aritméticas. Un uno en el bit de desbordamiento significa una magnitud desbordada, ya que el resultado no puede ser representado con el tamaño de operando especificado.

El bit "cero" ("zero" o Z) es uno, si el resultado de una operación es cero. Es cero, si el resultado es distinto de cero.

El bit "negativo" (N) es uno o cero, dependiendo del valor del bit más significativo del resultado de una operación con signo. Cuando es uno, el resultado es negativo; cuando es cero, el resultado es positivo o cero.

El bit de "extensión" (X) es utilizado en operaciones aritméticas de múltiple precisión. Cuando es afectado por una instrucción, es idéntico al bit de acarreo.

Los registros en modo usuario se representan en la Figura 2-1.

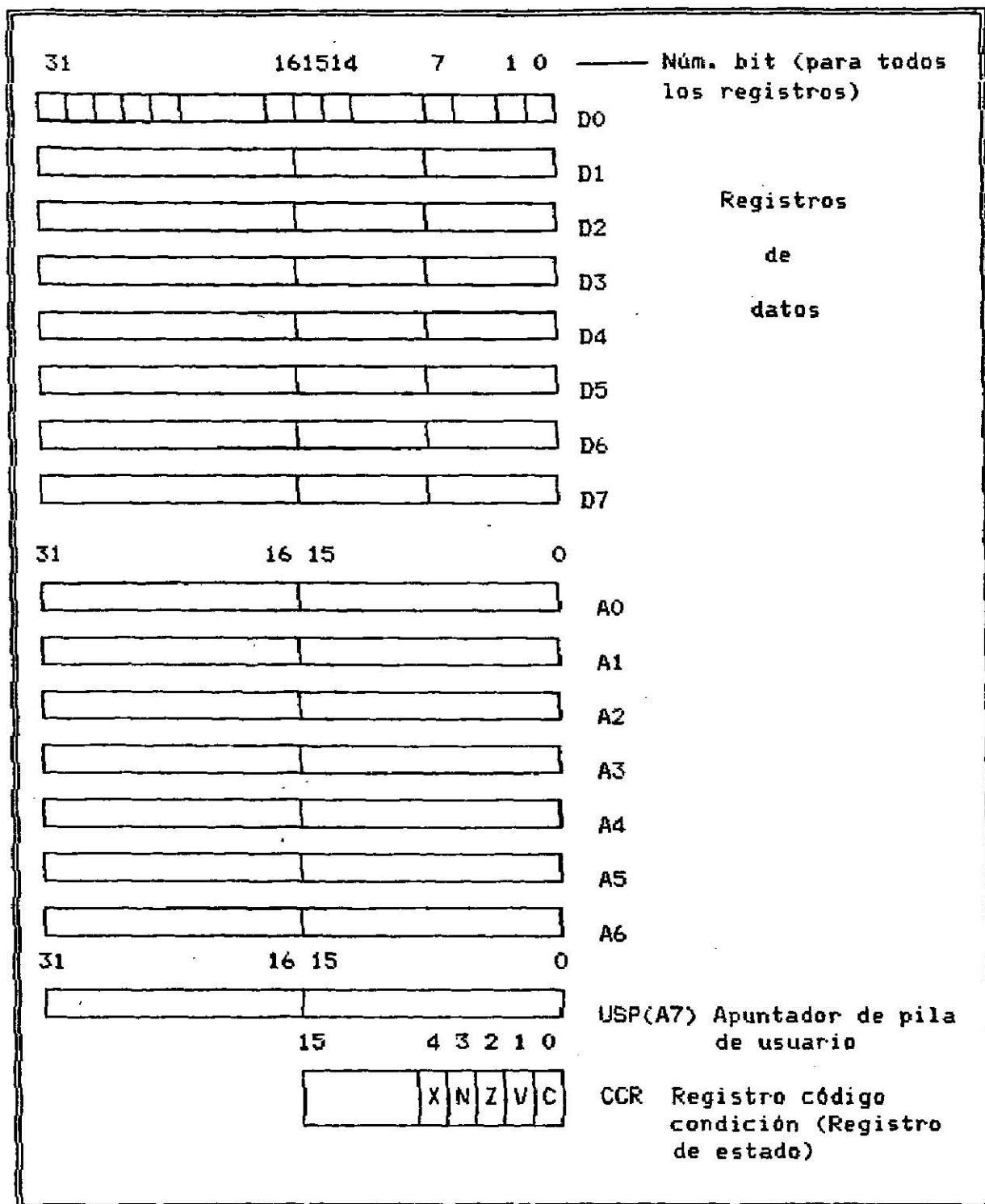


Figura 2-1.
Conjunto de registros para el modo usuario.

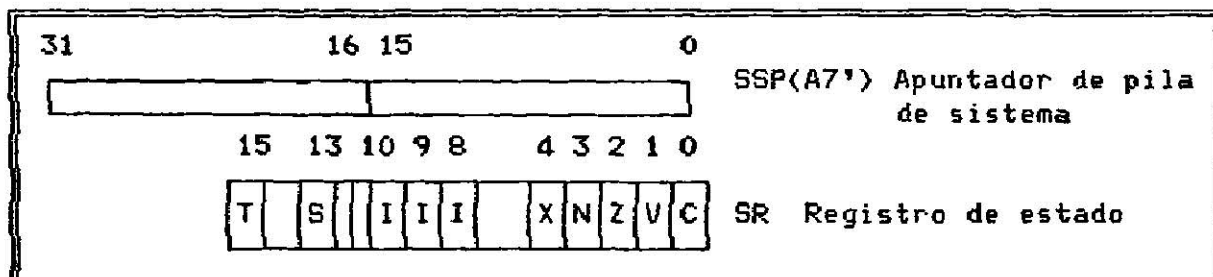


Figura 2-2.
Registros adicionales disponibles en modo supervisor (68000,68008).

2.4 REGISTROS PARA EL MODO SUPERVISOR.

Adicionalmente a los registros de datos, direcciones y de código de condición disponibles en el modo usuario, los elementos de la familia 68000 tienen varios registros adicionales utilizables en el modo supervisor. El tipo y número de registros del modo supervisor varía entre los miembros individuales de la familia 68000; por ello, se discutirá cada miembro separadamente. A menos que indique lo contrario, puede suponer que toda característica de un modelo inferior se transfiere a los modelos superiores.

68000 Y 68008

Los dos elementos inferiores, el 68000 y 68008, tienen dos registros adicionales para el modo supervisor, tal como muestra la Figura 2-2. Estos son el apuntador de la pila supervisor (A7' o SSP) y el byte superior del registro de estado. Similarmente al apuntador de la pila del usuario, el apuntador de la pila del modo supervisor tiene una longitud de 32 bits y inserta descendentemente en memoria; las referencias a la pila deben estar alineadas a palabra (direcciones pares). El procesador automáticamente utiliza el apuntador de la pila del supervisor cuando está en el modo supervisor. El supervisor no puede acceder al apuntador de la pila del usuario.

El byte del sistema del registro de estado junto con el registro de código de condición forman el registro de estado de 16 bits. El byte del sistema contiene dos bits indicadores y tres bits de máscara de interrupción.

El bit "supervisor" (S) especifica el modo de ejecución del procesador. Si es uno, el procesador está en modo supervisor; si es cero, el procesador está en modo usuario.

El bit de "modo traza" (T), cuando es uno, especifica que el procesador está en modo traza. En este modo, después de ejecutar una instrucción, el procesador automáticamente "intercepta" a través del vector número 9 de la tabla de vectores. Se discutirán los procesos excepcionales tales como éste en una parte posterior; por ahora, es suficiente decir que el modo traza conduce a un modo paso a paso de ejecución. Esto permite a un programa depurador monitorizar los resultados de un programa de aplicación ejecutándolo instrucción a instrucción.

Los procesadores 68K pueden operar en cualquiera de uno de ocho niveles o prioridades. Los dispositivos externos pueden intentar interrumpir al procesador con señales de validación según cualquier combinación en las líneas de petición de interrupción IPL0-IPL2. El valor binario de las líneas de interrupción representa la importancia del dispositivo que interrumpe; un dispositivo de prioridad alta (tal como el reloj) tiene un nivel de interrupción de valor superior al de un dispositivo de prioridad baja, tal como un controlador de terminal. La prioridad mayor es 7 (en binario 111) y el nivel inferior es 0 (en binario 000).

La máscara de interrupción en el registro de estado define el nivel de operación en curso del microprocesador. El código del modo normal del usuario, por convenio, normalmente es 0. Los controladores de dispositivos operan a niveles superiores; cuando un dispositivo interrumpe al procesador, la lógica interna compara el valor de la máscara de interrupción con el valor del dispositivo contenido en IPL0-IPL2.

Si las líneas de interrupción indican un nivel superior de prioridad, el procesador inicia un proceso excepcional. Si las líneas están a un nivel menor o igual, el procesador temporalmente ignora la petición de interrupción hasta que el código de ejecución cae a su propia prioridad (el valor en la máscara de interrupción).

La máscara de interrupción, en conjunción con las líneas de interrupción, suministra un medio utilísimo de jerarquizar las prioridades de ejecución. Esto asegura que los periféricos lentos pueden interrumpir la ejecución de programas normales y que los dispositivos sensibles al tiempo pueden, en turno, interrumpir los controladores periféricos lentos.

El procesamiento excepcional conduce a más complejidad como para discutirlo aquí. Así que una parte más adelante describe con completo detalle las acciones tomadas por el procesador y periféricos cuando tratan con procesos excepcionales.

El 68010 y 68012, además del apuntador de la pila del supervisor y el byte del sistema, del registro de estado, incluye un "registro base de vectores" de 32 bits y dos registros de código de función alternativos. Estos registros adicionales se ilustran en la figura 2-3.

Tanto el 68000 como el 68008 tienen una tabla de vectores de excepción fija. Esta tabla comienza en la dirección 0000H y es cargada por el software a nivel de sistema con las direcciones de las rutinas de procesamientos excepcionales. El procesador automáticamente "apunta" (o "vectoriza") con esas direcciones cuando encuentra una condición de excepción. En esta parte es suficiente decir que mientras el 68000 y el 68008 requieren por hardware que la tabla de vectores comience en 0000H, el 68010 permite que por software se especifique la dirección de comienzo de la tabla de vectores por medio del registro base de vectores (VBR).

Como mencionamos antes, la familia de procesadores 68K incluye tres líneas de código de función como terminales de salida del circuito integrado. Estos códigos de función indican a cualquier periférico el tipo de acceso actualmente en progreso en el procesador. Implícitamente, estos modos pueden indicar acceso al programa del usuario, acceso a los datos del usuario, acceso al programa supervisor, acceso a los datos del supervisor y reconocimiento de interrupción.

Los registros de código de función alternativos (SFC y DFC) en el 68010 permiten a un programa a nivel de sistema especificar su propia salida del código de función durante la fase de captación/almacenamiento fuente y destino de la instrucción MOVES (llevar de/hacia registro estado).

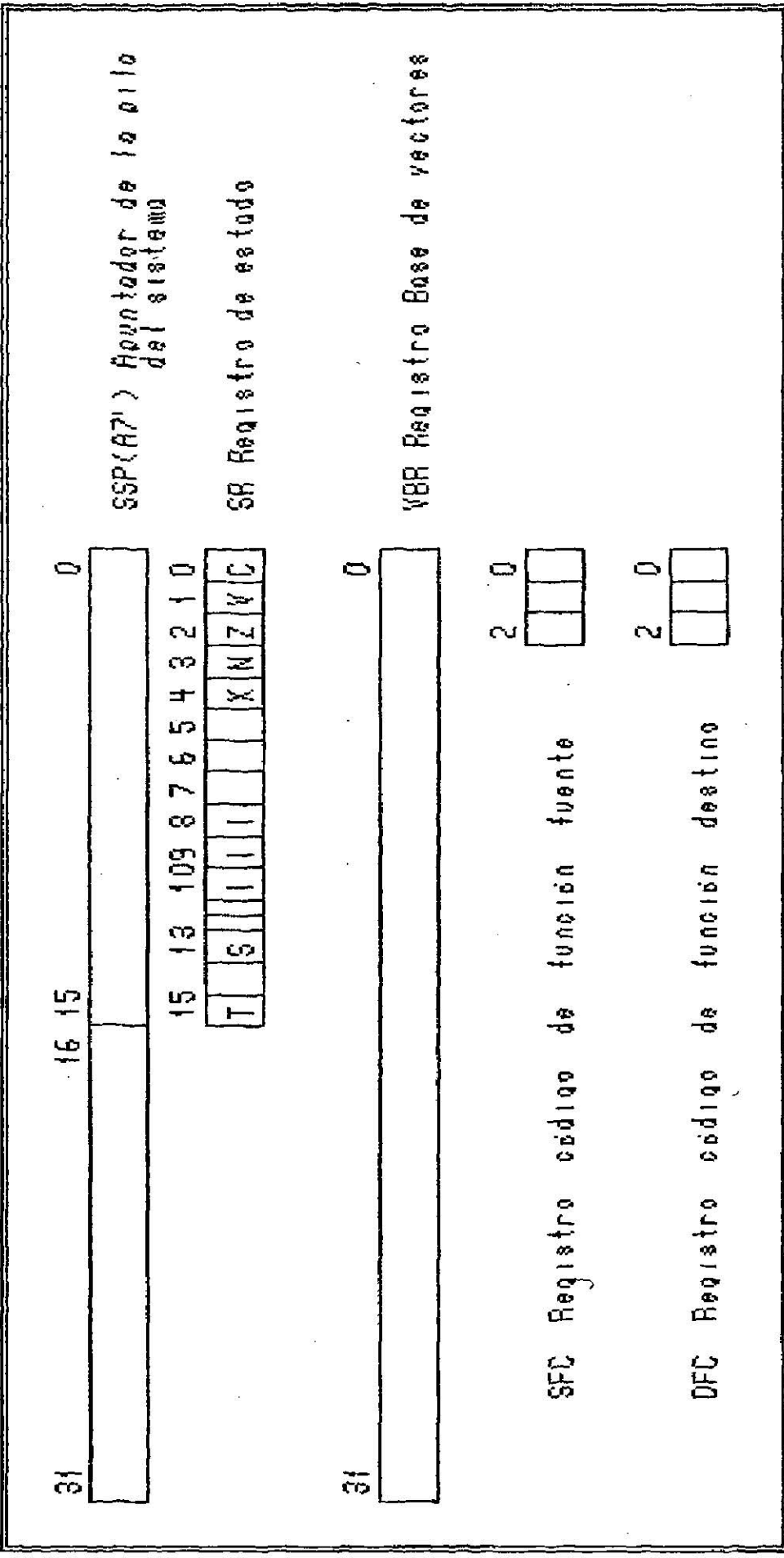


Figura 2-3
Registros adicionales en modo supervisor (68010, 68012).

El 68020 utiliza todos los registros del supervisor discutidos dentro de los otros procesadores 68K. Además, el 68020 tiene en la pastilla dos registros aceleradores de instrucciones y dos bits más definidos en el byte del supervisor del registro de estado. Todos estos registros adicionales del 68020 se muestran en la figura 2-4.

Los registros aceleradores (en inglés: "cache") permiten la manipulación software del acelerador de instrucciones interior al circuito integrado. Este acelerador de instrucciones permite acceder muy rápidamente a instrucciones, permitiendo una ejecución rápida de segmentos de código en buses. El registro de control de acelerador (CACR) sirve para el control y accesos al estado del acelerador de instrucciones, mientras el registro de dirección de memoria rápida (CAAR) mantiene la dirección para las funciones de control del acelerador como se requiera.

El registro de estado del 68020 contiene un segundo bit de habilitación de traza (T0) para combinar con el bit de que disponen los otros miembros 68K. Los dos bits juntos realizan una traza más específica. Cuando es igual a 00, no se efectúa la traza. Cuando es igual a 01, únicamente se produce la traza en caso de un cambio del flujo del programa (tal como ejecución de un salto o llamada a una subrutina). Cuando es igual a 10, el procesador realiza la traza después de cualquier instrucción, como es habitual en los otros miembros 68K. El valor 11 no está definido.

El 68020 separa las operaciones de la pila del supervisor con dos apuntadores de pila distintos, el apuntador de pila de interrupción (ISP) y el apuntador de la pila principal (MSP). El bit principal (M) en el registro estado determina qué apuntador de la pila usará el procesador en modo supervisor (cuando el bit S es uno). Cuando el bit M es cero, el procesador utiliza el apuntador de la pila de interrupción. Esto es equivalente a los otros procesadores 68K. Si el bit M está a uno, el procesador utiliza el apuntador de la pila principal.

Tener dos apuntadores de la pila del supervisor es útil en sistemas de multiarea ya que ello permite al sistema operativo distinguir entre excepciones asíncronas (E/S, por ejemplo) y llamadas al sistema operativo desde una tarea de usuario. Esto suministra una interfaz "más clara", ya que una tarea de usuario puede requerir el servicio del ejecutivo a través de una instrucción TRAP. La pila principal puede mantener información dependiendo de la tarea y hacer de almacén

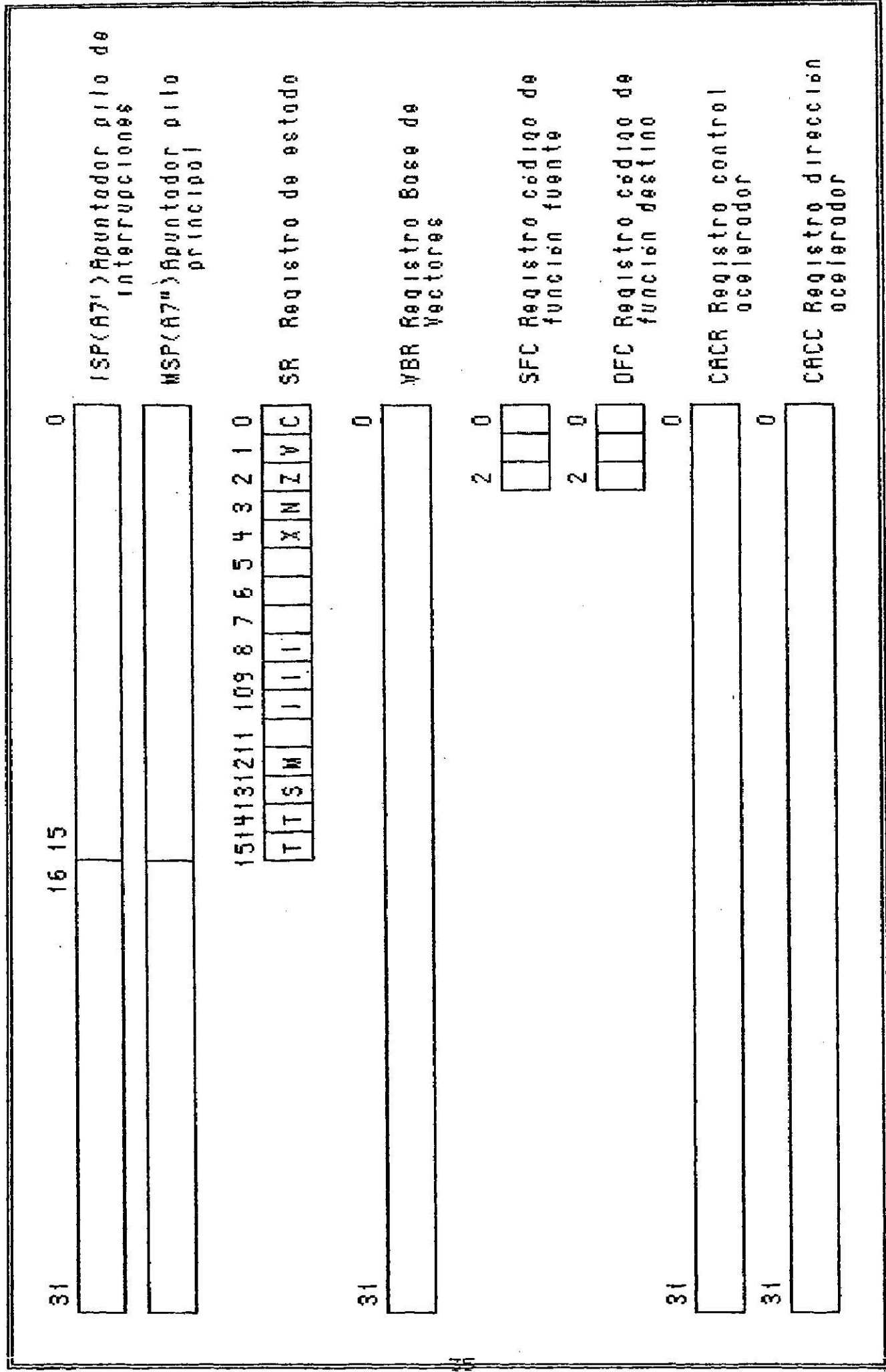


Figura 2-4. Registros adicionales disponibles en modo supervisor(68020)

para las peticiones de servicio del usuario. El apuntador de la pila de interrupción puede entonces usarse estrictamente para manejar excepciones asíncronas, que son generalmente independientes de las tareas procesándose.

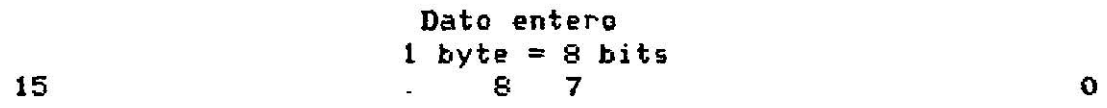
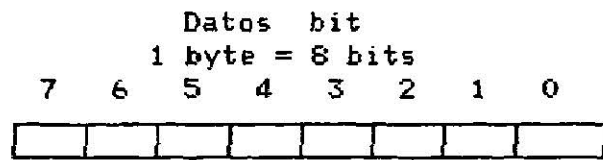
2.5 ORGANIZACION DE LA MEMORIA.

La familia 68K permite acceder a la memoria en grupos de bits construyendo un byte, palabra o palabra larga. Una dirección específica la posición del byte o, para datos de varios bytes, la posición del byte más significativo. El byte menos significativo de una palabra está en esa dirección más 1. El byte menos significativo de una palabra larga está en esa dirección más 3. La figura 2-5 muestra cómo están alineados en memoria los bytes, palabras y palabras largas.

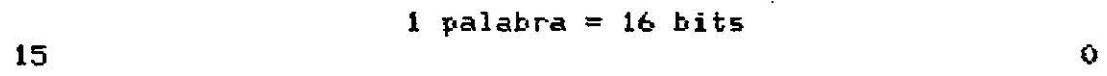
En el 68000, 68008, 68010 y 68012, las palabras y palabras largas (incluyendo palabras instrucción) deben comenzar en una dirección par. Este es un requerimiento impuesto por la naturaleza del bus de datos. Si usted intenta acceder a una palabra o palabra larga en una dirección impar, el procesador inicia un procesamiento excepcional, que se describirá más adelante en una parte completa.

El 68020 solventa esta limitación comprobando los accesos de palabras impares; si así ocurre, hace dos o más accesos parciales a la palabra o palabra larga. Naturalmente, esto añade una sobrecarga a la ejecución, y usted debería, cuando sea posible, intentar alinear las palabras en fronteras pares (bit 0 igual a cero) y las palabras largas en fronteras de palabras largas (bits 0 y 1 iguales a 0).

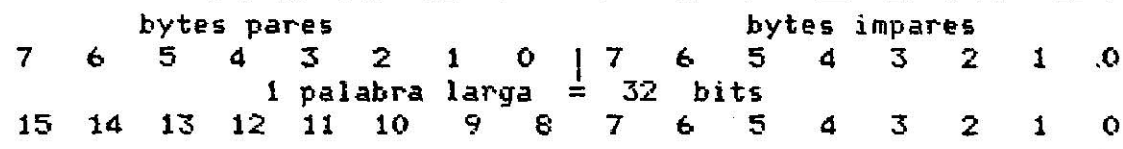
Para una máxima eficiencia de ejecución el 68020 incluye la restricción de la familia 68K de que las instrucciones comiencen en fronteras de palabras. El 68020 también utiliza en las inserciones y extracciones en su sistema pila palabras largas. Para mejor funcionamiento, la pila debería ubicarse en una frontera de palabra larga.



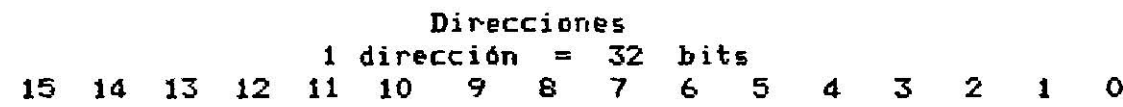
| | | | |
|-----|--------|-----|--------|
| MSB | Byte 0 | LSB | |
| | Byte 2 | | Byte 1 |
| | | | Byte 3 |



| | | |
|-----|-----------|-----|
| MSB | palabra 0 | LSB |
| | palabra 1 | |
| | palabra 2 | |

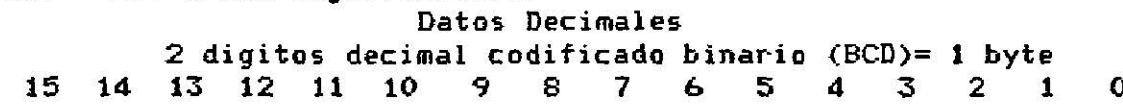


| | | |
|-----|-----------------|----------------|
| MSB | palabra larga 0 | Orden Superior |
| | | Orden Inferior |
| | palabra larga 1 | LSB |
| | palabra larga 2 | |



| | | |
|-----|-------------|----------------|
| MSB | dirección 0 | Orden Superior |
| | | Orden Inferior |
| | dirección 1 | LSB |
| | dirección 2 | |

MSB = Bit más significativo.
LSB = Bit menos significativo.



| | | | | |
|-----------|-------|-----|-------|-------|
| MSD BCD.0 | BCD.1 | LSD | BCD.2 | BCD.3 |
| BCD.4 | BCD.5 | | BCD.6 | BCD.7 |

MSD = Dígito más significativo LSD = Dígito menos significativo

Figura 2-5. Organización de la memoria.

2.6 MEMORIA VIRTUAL.

La familia de procesadores 68K puede acceder a una gran memoria. Sin embargo, en muchos casos no resulta económico tener la máxima capacidad de memoria físicamente presente. El 68010, 68012 y 68020 permiten programas que utilizan técnicas de "memoria virtual" para hacer que el sistema aparezca como si tuviese el espacio de direcciones completo presente en la máquina.

Si el procesador intenta acceder a memoria que no está físicamente presente en el sistema, la unidad de manejo de memoria indica que hay un "fallo de página", lo que fuerza a una intercepción en el bus con un vector de excepción de error. Si la fuente indica que el programa estaba tratando de acceder a memoria que está en la actualidad en memoria secundaria (por ejemplo, disco), el sistema operativo puede leer el bloque de memoria necesitado en el disco y llevarlo a memoria física.

Una vez que los datos necesarios están en memoria, el sistema operativo devuelve el control al programa del usuario. El procesador utiliza una instrucción de continuación para completar la instrucción interrumpida. Utilizando este método, el procesador continúa la ejecución del programa en el ciclo de bus fallido (frecuentemente interno a la ejecución de una instrucción). Para administrar la continuación de la instrucción adecuadamente, el procesador debe salvar cierta información de su interior antes de comenzar a ejecutar el código máquina de la excepción. El 68010, 68012 y 68020 salvan esta información en la pila antes de comenzar el código de excepción y devuelven esta información al procesador después de ejecutar la instrucción de retorno (RTE) del proceso excepcional.

2.7 MAQUINAS VIRTUALES.

Frecuentemente, alrededor del desarrollo de sistemas, puede desear escribir programas en código máquina para acceder a un dispositivo que no está físicamente en su sistema --por ejemplo, a un puerto serial de comunicaciones --. También puede desear realizar instrucciones que su procesador no permite, por ejemplo, instrucciones de punto flotante para coprocesador.

Una "máquina virtual" fuerza procesos excepcionales cuando usted intenta acceder a una dirección de memoria indefinida (quizá el registro de estado de un dispositivo equivocado) o cuando demanda al procesador que ejecute una instrucción que no existe en su repertorio. Los procesos excepcionales pueden entonces emular por software el dispositivo o instrucción.

C O N T E N I D O

P A R T E I I I

M O D O S D E D I R E C C I O N A M I E N T O

| | | |
|--------|--|----|
| 3.1 | Introducción. | 42 |
| 3.2 | Codificación de direcciones. | 43 |
| 3.3 | Modos de direccionamiento. | 47 |
| 3.3.1 | Directo a registro de datos. | 47 |
| 3.3.2 | Directo a registro de direcciones. | 47 |
| 3.3.3 | Indirecto a registro de dirección. | 47 |
| 3.3.4 | Indirecto con registro de dirección y post-incremento. | 48 |
| 3.3.5 | Indirecto con registro de dirección y pre-decremento. | 48 |
| 3.3.6 | Indirecto con registro de dirección y desplazamiento. | 49 |
| 3.3.7 | Indirecto con reg. de dir. e índice y desplazamiento. | 50 |
| 3.3.8 | Indirecto con memoria y post-índice. | 51 |
| 3.3.9 | Indirecto con memoria y pre-índice. | 52 |
| 3.3.10 | Indirecto con PC y desplazamiento. | 53 |
| 3.3.11 | Indirecto con PC e índice y desplazamiento. | 53 |
| 3.3.12 | Indirecto a memoria con PC y post-índice. | 54 |
| 3.3.13 | Indirecto a memoria con PC y pre-índice. | 55 |
| 3.3.14 | Absoluto corto. | 56 |
| 3.3.15 | Absoluto largo. | 56 |
| 3.3.16 | Dato inmediato. | 56 |
| 3.4 | Categorías. | 57 |

3.1 INTRODUCCION

En esta parte se describiran los modos de direccionamiento que utiliza la familia 68k. Asi como tambien la codificaci6n de direcciones.

La codificaci6n de direcciones, que es el punto 3.2, describe la forma en que estan distribuidos los campos de los modos de direccionamiento en las palabras de memoria.

Los modos de direccionamiento que utiliza la familia 68k pueden agruparse en siete tipos b6sicos.

Estos siete grupos son :

- 1.- Direccionamiento directo a registro :
 - a) Direccionamiento a registro de datos.
 - b) Direccionamiento a registro de direcciones.
- 2.- Indirecto con registro:
 - a) Indirecto con registro de direcci6n.
 - b) Indirecto con registro de direcci6n y post-incremento.
 - c) Indirecto con registro de direcci6n y pre-decremento.
 - d) Indirecto con registro de direcci6n y desplazamiento.
 - e) Indirecto con registro de direcci6n e indice y desplazamiento de 8 bits *
- 3.- Indirecto con memoria:
 - a) Indirecto con memoria y post-Indice. *
 - b) Indirecto con memoria y pre-Indice. *
- 4.- Indirecto con contador de programa:
 - a) Indirecto con PC y desplazamiento de 16 bits.
 - b) Indirecto con PC e indice y desplazamiento de 8 bits.
 - c) Indirecto con PC e indice y desplazamiento de 16 32 bits. *
- 5.- Indirecto con memoria y con el contador de programa:
 - a) Indirecto a memoria con PC y post-Indice. *
 - b) Indirecto a memoria con PC y pre-Indice. *
- 6.- Absoluto:
 - a) Absoluto corto.
 - b) Absoluto amplio.
- 7.- Inmediato.

Los modos marcados con un asterisco(*) o bien tienen funciones ampliadas en el 68020 o bien existen s6lo en el 68020.

Cada uno de los modos anteriores se describen con m6s detalle en el punto 3.3 de esta parte III.

3.2 CODIFICACION DE DIRECCIONES

Las instrucciones de la familia 68k codifican las direcciones de operandos de una manera específica. La primera palabra especifica la operación y, en muchos casos, la dirección de un operando. Las especificaciones de operandos adicionales van a continuación de esta primera palabra.

En los primeros cuatro miembros de la familia 68k hay, a lo mucho, cuatro palabras de "ampliación". El 68020 permite hasta 10 palabras de ampliación (a causa de los modos adicionales de direccionamiento).

Las direcciones efectivas se codifican con ayuda de un campo modo y un campo registro. Cada uno de esos campos tiene una longitud de 3 bits y están incluidos en la primera palabra de instrucción. Cualquiera de las palabras de ampliación que sigan pueden ser sencillamente o bien direcciones de operandos, en forma de direcciones absolutas, o pueden ser formas bastante complejas con campos de direcciones incluidas o con modos de direccionamiento indirecto a memoria. Los campos de ampliación más complejos incluyen valores para indexación y direccionamiento indirecto, además de para la presencia o no del tamaño del desplazamiento, en su caso.

TABLA 3-1
Campos en los modos de direccionamiento.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

| | | | | | | | | | | | | | | |
|------------------------------|--|--|--|--|--|--|------|--|--|----------|--|--|--|--|
| código operación de la inst. | | | | | | | modo | | | registro | | | | |
|------------------------------|--|--|--|--|--|--|------|--|--|----------|--|--|--|--|

formato de dirección efectiva única

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

| | | | | | | | | | | | | | | |
|-----|-------------|-----|-------------|---|----------------|--|--|--|--|--|--|--|--|--|
| D/A | Reg. índice | W/L | esca- la | 0 | desplazamiento | | | | | | | | | |
|-----|-------------|-----|-------------|---|----------------|--|--|--|--|--|--|--|--|--|

palabra de ampliación

| | | | | | | | | | | | | | | |
|-----|-----------------|-----|-------------|---|----|----|----------------|---|------|--|--|--|--|--|
| D/A | registro índice | W/L | esca- la | 1 | BS | IS | BD ta- maño | 0 | I/IS | | | | | |
|-----|-----------------|-----|-------------|---|----|----|----------------|---|------|--|--|--|--|--|

desplazamiento base (0,1 ó 2 palabras)

desplazamiento externo(0,1 ó 2 palabras)

palabra(s) de ampliación, formato completo(sólo 68020)

- registro registros de datos o direcciones (vease Tabla 3-2)
- modo modo de direccionamiento (vease Tabla 3-2)
- código de operación inst. y posible información sobre modo/registro para el segundo operando.
- desplazamiento valor con signo de 8 bits.
- escala índice de factor de escala (sólo en 68020)
00 = 1X , 01 = 2X , 10 = 4X , 11 = 8X
- I/IS Selección indexado/indirecto(sólo 68020, vease Tabla 3.3).
- BD tamaño Tamaño de desplazamiento base(sólo 68020).
00 reservado 01 desplazamiento nulo
10 desplazamiento de palabra 11 desplazamiento largo
- IS Supresión de índice(sólo 68020, vease Tabla 3-3)
- BS Supresión de base (sólo 68020)
0 evaluar y añadir registro base
1 suprimir registro base
- W/L Tamaño de registro índice
0 palabra extendida en signo
1 palabra larga con signo
- registro índice registro de datos o de direcciones (000-111).
- D/A Tipo de registro índice
0 registro de datos 1 registro de dirección

Tabla 3-2 Codificación modo/registro.

| Modo 000 | Registro # reg | Operación de direccionamiento Directo a registro de datos |
|-------------|-------------------|--|
| 000 | # reg | Directo a registro de datos. |
| 001 | reg # | Directo a registro de dirección. |
| 010 | reg # | Indirecto a reg. de dirección. |
| 011 | reg # | Indirecto a reg. de dir. con post-incremento. |
| 100 | reg # | Indirecto a reg. de dir. con predecremento. |
| 101 | reg # | Indirecto a reg. de dir. con desplazamiento. |
| 110 | reg # | Indirecto a memoria con reg. de dir. con índice.* |
| 111 | 000 | Absoluto corto. |
| 111 | 001 | Absoluto largo. |
| 111 | 010 | Indirecto con el PC y desplazamiento. |
| 111 | 011 | Indirecto a memoria con el PC con índice.* |
| 111 | 100 | Dato Inmediato. |
| 111 | 101-111 | Reservado. |

* Solo el 68020.

Tabla 3-3 Codificación de modos de direccionamiento IS-I/S.

| IS | I/IS | Operación de direccionamiento |
|----|---------|---|
| 0 | 000 | indexado, no indirecto a memoria. |
| 0 | 001 | Indirecto pre-índice, sin ningún desplazamiento externo. |
| 0 | 010 | Indirecto pre-índice con desplazamiento externo de palabra. |
| 0 | 011 | Indirecto pre-índice con desplazamiento externo largo. |
| 0 | 100 | Reservado. |
| 0 | 101 | Indirecto post-índice sin ningún desplazamiento externo. |
| 0 | 110 | Indirecto post-índice con desplazamiento externo de palabra. |
| 0 | 111 | Indirecto post-índice con desplazamiento externo largo. |
| 1 | 000 | Sin índice, no indirecto a memoria. |
| 1 | 001 | Sin índice indirecto a memoria sin desplazamiento externo. |
| 1 | 010 | Sin índice indirecto a memoria con desplazamiento externo de palabra. |
| 1 | 011 | Sin índice indirecto a memoria con desplazamiento externo largo. |
| 1 | 100-111 | Reservado. |

(Sólo en el 68020)

La tabla 3-1 resume los campos del modo de direccionamiento tanto en el interior de la palabra instrucción como en el interior de las palabras de ampliación.

La tabla 3-2 resume los valores modo/registro y sus funciones.

La tabla 3-3 resume los valores de la selección de indexado/indirecto y Supresión de índice, ambos del 68020, solamente.

3.3 MODOS DE DIRECCIONAMIENTO

3.3.1 Directo a registro de datos

En este modo, un registro de datos contiene el operando.

registro de datos :

| |
|----------|
| operando |
|----------|

Sintaxis en ensamblador : Dn

3.3.2 Directo a registro de direcciones

En este modo, un registro de dirección contiene el operando.

Registro de dirección :

| |
|----------|
| operando |
|----------|

Sintaxis en ensamblador : An

3.3.3 Indirecto con registro de dirección.

En este modo, un registro de dirección contiene la dirección del operando ("apunta" al operando) en la memoria.

Registro de dirección :

| |
|----------------------|
| dirección de memoria |
|----------------------|

Memoria :

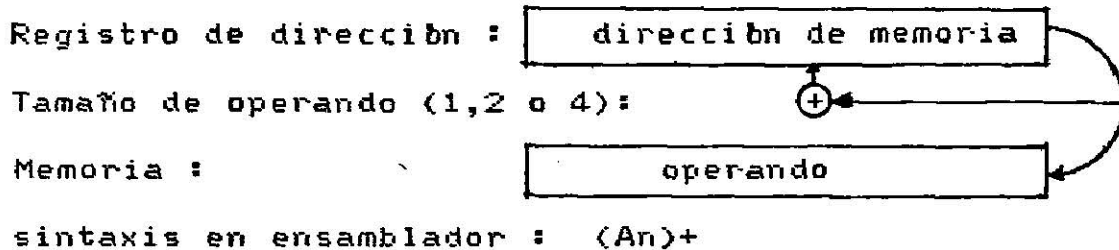
| |
|----------|
| operando |
|----------|

Sintaxis en ensamblador : An

3.3.4 Indirecto con registro de dirección y post-incremento.

En este tipo de modo de direccionamiento, un registro de dirección contiene la dirección donde se encuentra el operando. Sin embargo, después de usar el operando, el procesador suma el registro de dirección, la longitud del operando en byte (esto es, 1 para una operación de 1 byte, 2 para una operación con una palabra o 4 para una operación con una palabra larga)

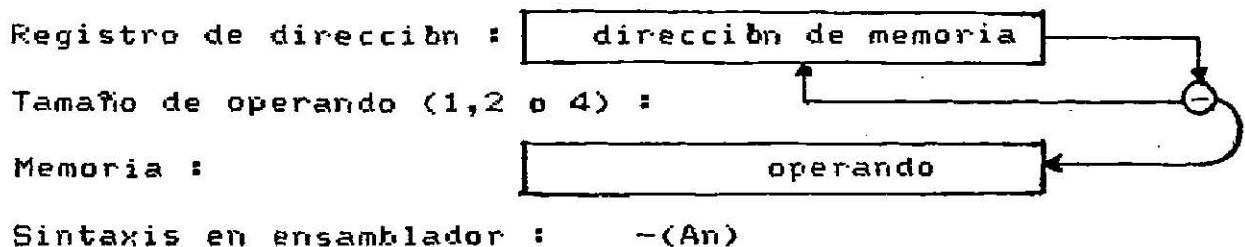
Hay que notar que si el registro de dirección es el apuntador de la pila y la operación es de una longitud de un byte, el procesador automáticamente añadirá 2 al registro. Esto fuerza a la pila a permanecer con un alineamiento en palabra para una máxima eficiencia.



3.3.5 Indirecto con registro de dirección y pre-decremento.

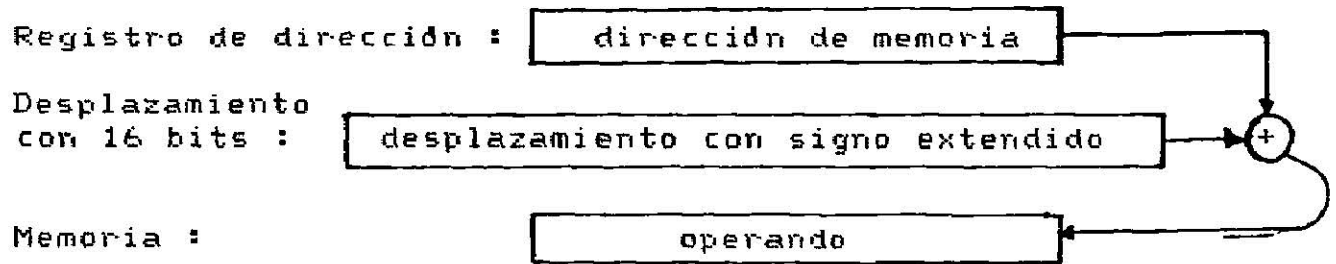
En este modo, un registro de dirección contiene la dirección del operando. Sin embargo, antes de usar el operando, el procesador resta la longitud del operando al registro de dirección en bytes (esto es, 1 para una operación de un byte, 2 para una operación con palabra o 4 para una operación con palabra larga).

Algo muy importante es que si el registro de dirección es el apuntador de la pila y la operación es con un byte, el procesador automáticamente resta 2 del registro. Esto fuerza a la pila a permanecer alineada en palabras para una eficiencia máxima.



3.3.6 Indirecto con registro de dirección y desplazamiento.

En este modo el operando reside en la dirección dada por la suma de los contenidos de un registro de dirección y un desplazamiento de 16 bits. El valor del desplazamiento es trasladado en signo a 32 bits para permitir desplazamientos negativos.

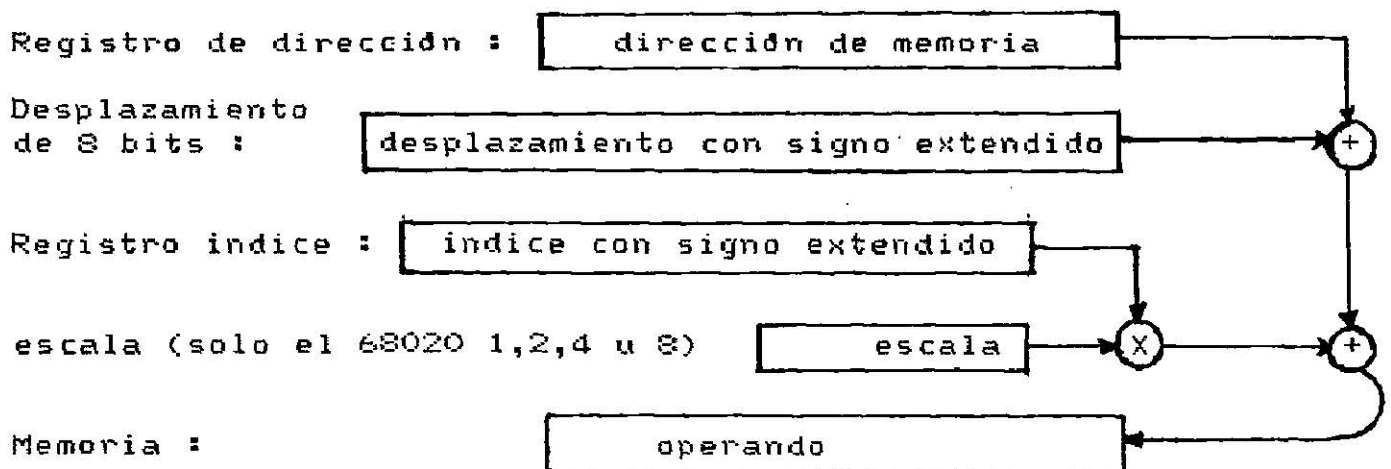


Sintaxis en ensamblador: (d_{16}, An)

3.3.7 Indirecto con registro de dirección e índice y desplazamiento.

En este modo, el operando reside en una dirección dada por la suma de los contenidos de un registro de dirección, un desplazamiento y el producto del registro índice por un factor de escala.

Si la base del desplazamiento es de 8 o 16 bits, está el signo trasladado a 32 bits. (Hay que hacer notar que la versión de 8 bits de este tipo de direccionamiento sólo existe en el 68020). El valor del registro índice sólo puede ser un valor de 16 o 32 bits; si es 16 bits, el procesador traslada el signo a 32 bits antes de añadirlo. El factor de escala puede ser 2 (1), 2 (2), 2 (4) o 2 (8). Sólo el 68020 admite índice escalado.



Sintaxis en ensamblador : (d_e, An, Xn, tamaño)

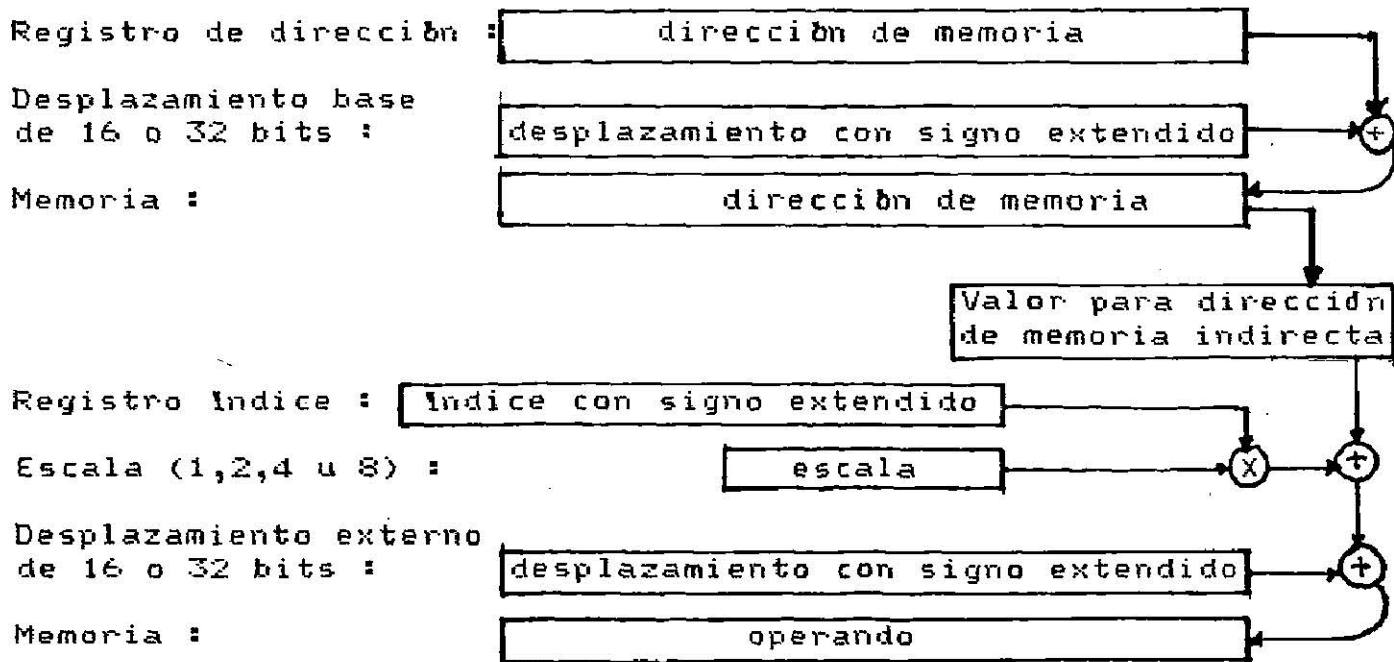
68020 : (d_e, An, Xn, tamaño * escala)

3.3.8 Indirecto con memoria y post-índice.

El modo indirecto con memoria y post-índice es exclusivo, por así decirlo, del 68020; En este modo la dirección del operando se obtiene a partir de cuatro valores: el contenido de un registro de dirección, un desplazamiento base de 16 o 32 bits, el producto del valor en un registro índice por la escala y un segundo desplazamiento (externo).

La base y el desplazamiento externo pueden tener una longitud de 16 o 32 bits; si son de 16 bits, el procesador traslada el signo de los valores antes de añadirlos. El registro índice es también ampliado y es trasladado el signo (cuando se requiera) antes de añadirlo a la dirección efectiva. Los cuatro valores son opcionales; si se omiten, el procesador suma un valor cero de contribución al cálculo de la dirección efectiva.

Al calcular la dirección efectiva, el procesador añade el valor del registro de dirección al desplazamiento base. Después utiliza este valor para obtener un segundo valor de dirección. Para este segundo valor, el procesador añade la escala del registro índice. Finalmente, añade el desplazamiento externo para obtener la dirección del operando.



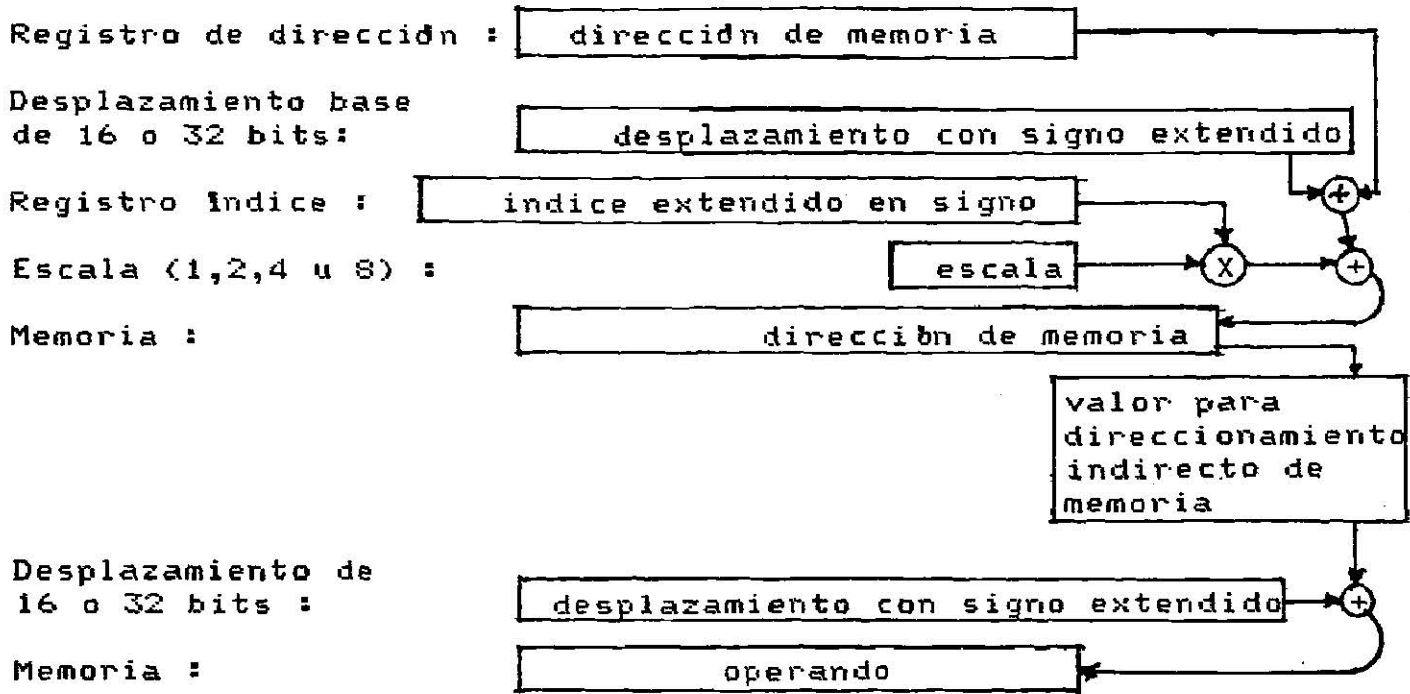
Sintaxis ensamblador : ([db,An],Xn,tamaño * escala, de)

3.3.9 Indirecto con memoria y pre-índice.

Este modo es exclusivo del 68020, la dirección del operando se obtiene a partir de cuatro valores: el contenido de un registro de dirección, un desplazamiento base de 16 o 32 bits, el producto del valor de un registro índice y la escala y un segundo desplazamiento externo.

La base y el desplazamiento externo pueden ser de 16 o 32 bits; si son de 16 bits el procesador traslada el signo antes de sumarlos. El registro índice es también ampliado y su signo trasladado (cuando sea necesario) antes de ser sumado a la dirección efectiva. Los cuatro valores son opcionales; si se omiten, el procesador supone un valor alto para esa contribución a la dirección efectiva.

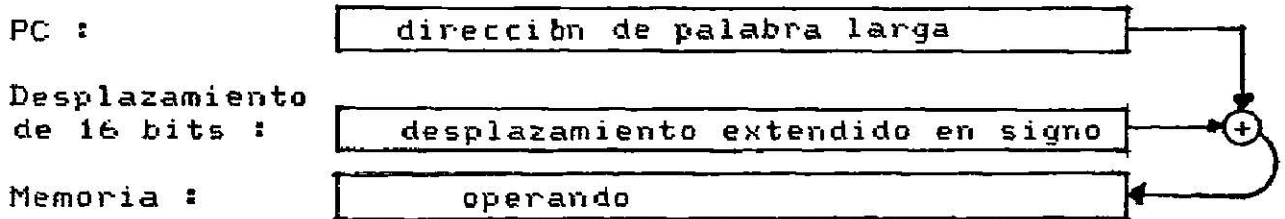
Al evaluar la dirección efectiva, el procesador añade el valor del registro de dirección al desplazamiento base y el valor escala del registro índice. Después utiliza este valor para obtener una segunda dirección. Para calcular esta segunda dirección, el procesador suma el desplazamiento externo consiguiendo así la dirección del operando.



Sintaxis en ensamblador: ([db,An],Xn,tamaño * escala, de)

3.3.10 Indirecto con PC y desplazamiento.

En este modo el procesador genera una dirección efectiva sumando el valor del contador de programa (program counter; PC) y el valor con el signo trasladado de esa palabra de ampliación de 16 bits.

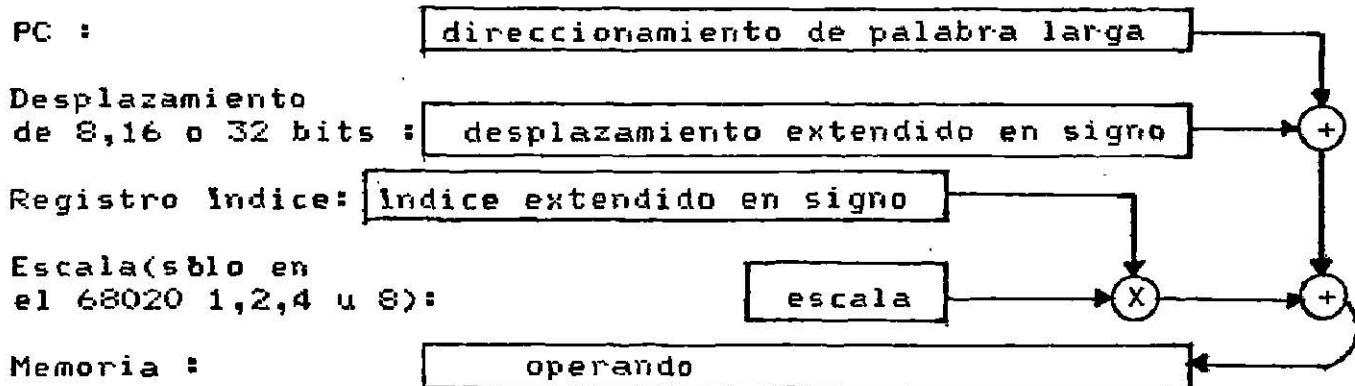


Sintaxis en ensamblador: (d_{16} , PC)

3.3.11 Indirecto con PC e índice y desplazamiento.

En este modo la dirección efectiva es la suma de la dirección en el contador de programa, el valor en la palabra de ampliación y el valor en el registro índice (multiplicado por la escala, en el caso del 68020).

El valor del contador del programa en el momento de la evaluación será la dirección de la palabra de ampliación. El procesador trasladará el signo del desplazamiento si es de 8 o 16 bits (la versión de 8 bits es sólo del 68020). El índice del valor en el registro índice puede ser trasladado, si es necesario, así como el del valor escalado, en el procesador 68020.

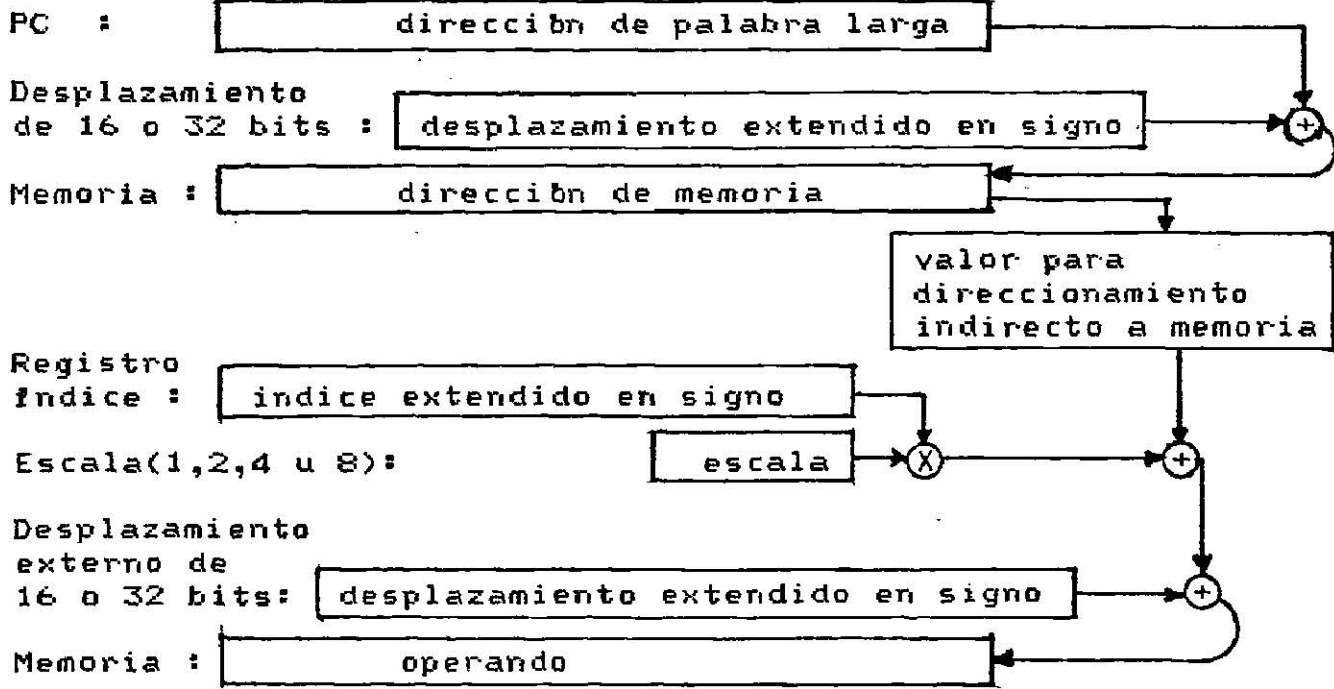


Sintaxis en ensamblador : (d_n , PC, $X_n \cdot \text{tamaño}$)
 68020 : (d_n , PC, $X_n \cdot \text{tamaño} * \text{escala}$)

3.3.12 Indirecto a memoria con PC y post-índice.

En este modo la dirección efectiva se calcula a partir de cuatro valores. La suma del PC y un desplazamiento base se utiliza como una dirección. El procesador suma el valor de dicha dirección con una escala en el registro índice y con el desplazamiento externo, obteniendo así la dirección del operando.

El valor del contador de programa es el de la dirección de la palabra de ampliación. La base y el desplazamiento externo pueden ser de 16 o 32 bit; si la longitud es de 16 bits, el procesador traslada los signos antes de usar los valores. El valor en el registro índice puede ser 1,2,4 u 8. Todos los componentes utilizados en el cálculo de la dirección del operando son opcionales; si se omiten, el procesador asume para ellos el valor cero.



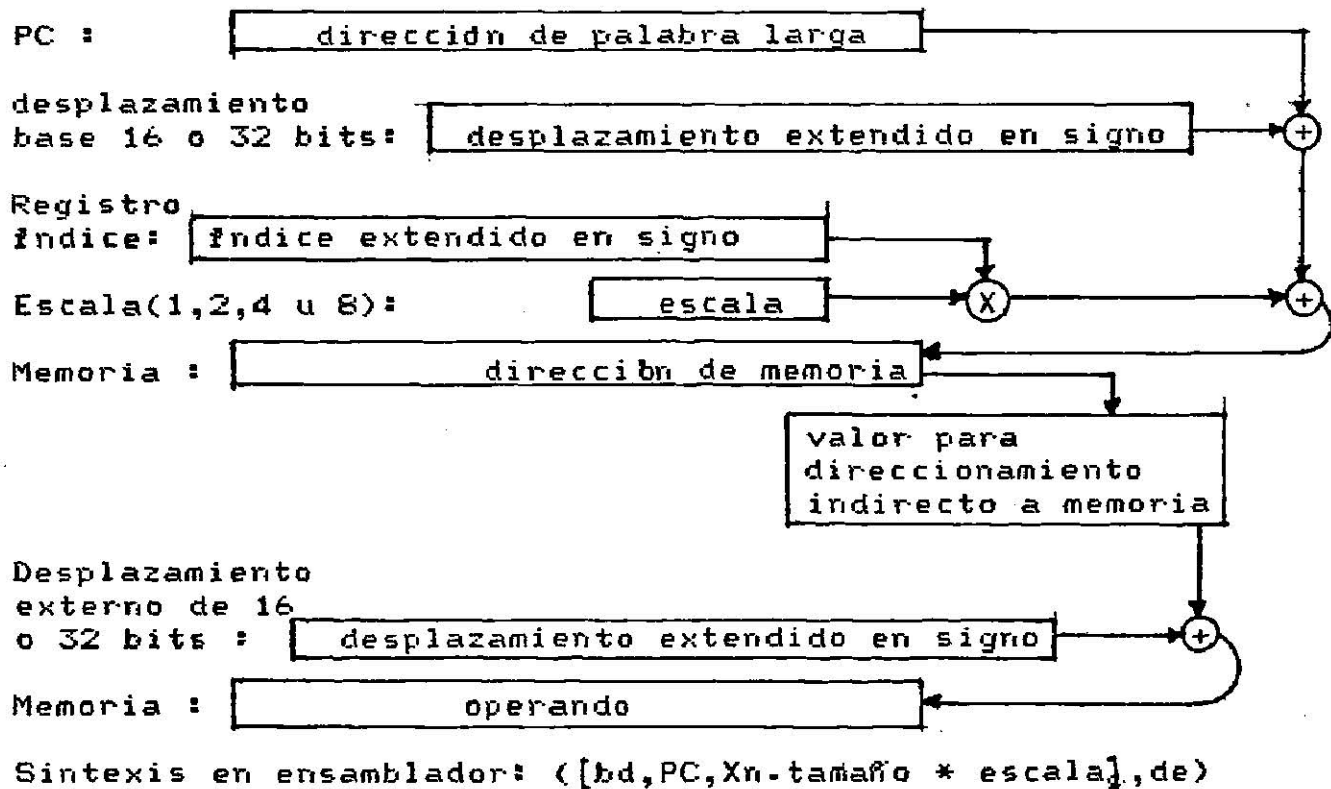
Sintaxis en ensamblador: ([bd,PC],Xn.tamaño * escala, de)

3.3.13 Indirecto a memoria con PC y pre-índice.

En este modo el cálculo de la dirección efectiva se hace con cuatro valores. La suma del PC, con el desplazamiento base y con la escala en el registro índice, se usa como una dirección de partida. El procesador suma el valor de esa dirección con el desplazamiento externo para dar la dirección del operando.

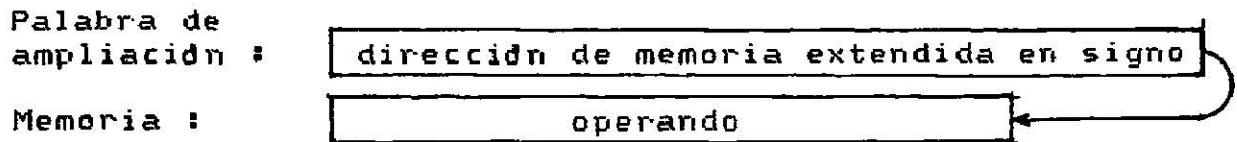
El valor del contador de programa es la dirección de la palabra de ampliación. Los desplazamientos base y externo pueden ser de 16 o 32 bits; si es de una longitud de 16 bits, el procesador traslada el signo antes de usar el valor.

El valor en el registro índice puede estar en un byte, una palabra o una palabra larga y su signo se traslada si es necesario. El valor de escala puede ser 1,2,4 u 8. Todos los componentes utilizados en el cálculo de la dirección del operando son opcionales; si se omiten, el procesador supone un valor cero.



3.3.14 Absoluto corto.

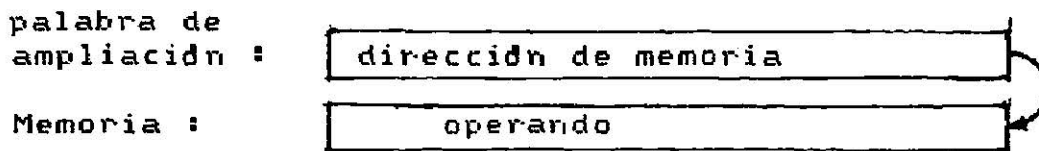
En este modo la direccidn del operando es el valor, con el signo trasladado, de la palabra de 16 bits de ampliacion que sigue a la instruccin.



Sintaxis en ensamblador: xxxx.W

3.3.15 Absoluto largo.

En este modo la direccidn del operando estd en la palabra de ampliacion (32 bits) que sigue a la palabra instruccin.



Sintaxis en ensamblador : xxxx.L

3.3.16 Dato inmediato.

En este modo el propio operando sigue a la propia instruccin. Dependiendo del tamao del operando, la instruccin necesita una o dos palabras de ampliacion.



Sintaxis en ensamblador: #xxxx.tamafio

3.4 CATEGORIAS

Los modos de direccionamiento son altamente compatibles en todos los miembros de la familia 68k. Y se pueden dividir en cuatro categorias principales:

DATO: Si un modo de direccionamiento debe referirse a un operando dato.

MEMORIA: Si un modo de direccionamiento debe ser usado para referirse a operandos en memoria.

CONTROL: Si el procesador no necesita saber del tamaño del operando previo para calcular la direccion efectiva.

ALTERABLE: Si el modo de direccionamiento debe referir a un operando escrito.

Después de leer las cuatro categorias se puede observar que cualquier modo dado debe ser clasificado en más de una categoria, como se puede ver en la tabla 3.4.

Estos modos proveen una substancial cantidad de flexibilidad y potencia o poder.

Tabla 3.4 Modos de direccionamiento.

| MODO | Sintaxis | Datos | Memoria | Control | Alterable |
|---|-----------|-------|---------|---------|-----------|
| Directo a registro de datos | Dn | † | | | † |
| Directo a registro de direccion | Rn | | | | † |
| Indirecto a registro de direccion | (Rn) | † | † | † | † |
| Indirecto a registro de direccion con post-incremento | (Rn)+ | † | † | | † |
| Indirecto a registro de direccion con pre-decremento | -(Rn) | † | † | | † |
| Indirecto a registro de direccion con desplazamiento | d(Rn) | † | † | † | † |
| Indirecto a registro de direccion con indice | d(Rn, Rx) | † | † | † | † |
| Absoluto corto | xxx.W | † | † | † | † |
| Absoluto largo | xxx.L | † | † | † | † |
| PC relativo con desplazamiento | d(PC) | † | † | † | |
| PC relativo con indice | d(PC, Rx) | † | † | † | |
| Inmediato | #xxx | † | † | | |

Dn significa cualquiera de los ocho registros de datos de 32 bits, D0 - D7.

Rn significa cualquiera de los ocho registros de direccion de 32 bits, R0 - R7.

Rx significa cualquier registro de 32-bits, D0-D7 o R0-R7.

d significa desplazamiento de signo extendido de 8 o 16 bits.

+, - significa el automatico post-incremento/pre-decremento de un registro de direccion especifico por 1, 2 o 4 dependiendo del tamaño del operando (1, 2 o 4 bytes).

C O N T E N I D O

P A R T E I V

R E P E R T O R I O D E I N S T R U C C I O N E S

| | | |
|------|--|----|
| 4.1 | Introducción. | 60 |
| 4.2 | Transferencias de datos. | 61 |
| 4.3 | Aritmética entera. | 62 |
| 4.4 | Aritmética booleana. | 64 |
| 4.5 | Desplazamientos y rotaciones. | 65 |
| 4.6 | Manipulación de bits individuales. | 66 |
| 4.7 | Manipulación de campos de bits. | 67 |
| 4.8 | Manipulación en Decimal Codificado Binario (BCD). | 68 |
| 4.9 | Control del flujo de programa. | 69 |
| 4.10 | Control del sistema. | 70 |
| 4.11 | Comunicaciones Multitarea/Multiprogramación | 71 |
| 4.12 | Precaptación de instrucción/segmentación/ enlazamiento/aceleración. | 72 |

4.1 INTRODUCCION.

El repertorio de instrucciones del 68K tiene más de 300 instrucciones individuales. Sin embargo, muchas de esas instrucciones son similares; sus diferencias radican en los tipos de datos con que operan o en los modos de direccionamiento que utilizan. Realmente el 68K dispone de 55-60 instrucciones básicas; el número depende del miembro de la familia.

El formato básico de todas las instrucciones es el mismo tanto internamente como en los mnemónicos del ensamblador. Internamente, el código de operación es de una palabra, con el modo o los modos de dirección incluidos en el código de operación. Ciertos modos de direccionamiento requieren ampliaciones de la palabra de instrucción para especificar el operando. Dependiendo del modo, la instrucción total puede variar de una palabra a once palabras a lo mucho.

Motorola ha construido la familia 68K compatible hacia arriba a nivel de código objeto. Esto significa que todo código del 68000 es ejecutable en el 68020. Sin embargo, lo opuesto no se verifica, el código de máquina del 68020 no necesariamente funcionará en el 68000.

Las instrucciones pueden agruparse en diez tipos distintos:

- * Transferencias(o movimientos) de datos.
- * Aritmética entera.
- * Aritmética booleana.
- * Desplazamientos y rotaciones.
- * Manipulación de bits individuales.
- * Manipulación de campos de bits.
- * Manipulación en decimal codificado binario (BCD)
- * Control de flujo de programa.
- * Control del sistema.
- * Comunicaciones multi/area/multiprocesador.

Cada uno de ellos será tratado en esta parte.

Observe que no hay instrucciones para E/S (I/O). A diferencia de algunos otros microprocesadores, la familia 68K utiliza E/S "direccionadas como memoria" en lugar de puertas de E/S. Esto significa que el procesador se comunica con los dispositivos de E/S exactamente como si fueran posiciones de memoria, añadiendo simplicidad al repertorio de instrucciones para el programador y para el ingeniero de interfaces hardware.

4.2 TRANSFERENCIA DE DATOS.

Las instrucciones de transferencias de datos facilitan el desplazamiento de datos entre registros, entre un registro y una posición de memoria y directamente entre dos posiciones de memoria. Estas instrucciones permiten las transferencias de datos, de direcciones, intercambio de datos en registros, carga y almacenamiento de registros múltiples y enlazado y desenlazado de estructuras de pila.

Dos observaciones deben hacerse con las instrucciones de transferencia de datos.

Primera, no hay instrucciones generales PUSH/POP ("insertar/extraer"). Puede realizar operaciones de pila con una instrucción más general (MOVE) utilizando direccionamiento con auto-incremento y auto-decremento con (A7)+ y (A7)-.

Segunda, a diferencia de algunos microprocesadores, no hay una instrucción de transferencia de bloques. Pueden, de nuevo, transferirse bloques con velocidad, utilizando el modo de direccionamiento con auto-incremento o auto-decremento en conjunción con una instrucción de decremento y ramificación.

Tabla 4-1.
Instrucciones de transferencia de datos.

| Mnemónicos | Operación |
|------------|-------------------------------------|
| EXG | Intercambio de registros |
| LEA | Carga dirección efectiva |
| LINK | Enlazar ubicar pila |
| MOVE | Transferir fuente a destino |
| MOVEA | Transferir fuente a registro dir. |
| MOVEC | Transferir registro de control |
| MOVEM | Transferencia múltiple de registros |
| MOVEP | Transferencia a periférico |
| MOVEQ | Transferir datos cortos a destino |
| MOVES | Transferir espacio de direcciones |
| PEA | Insertar en pila dirección efectiva |
| UNLK | Desenlazar pila |

4.3 ARITMETICA ENTERA

El 68K proporciona cuatro funciones aritméticas enteras básicas: suma, resta, multiplicación y división. También suministra ayudas para las operaciones aritméticas: comparación de dos enteros, hacer cero un entero, cambiar de signo un entero y se pueden efectuar operaciones en aritmética(ampliada) en múltiple precisión. Usted (lector) puede sumar, comparar y restar tanto direcciones como datos; la manipulación con direcciones está restringida a valores de 16 y 32 bits mientras que la manipulación de datos puede incluir valores de 8 bits.

Las operaciones de multiplicación y división pueden efectuarse con signo o sin signo, según sea necesario. Todos los miembros 68K pueden multiplicar dos enteros de 16 bits generando un producto de 32 bits y pueden dividir un dividendo de 32 bits por un divisor de 16 bits produciendo un cociente de 16 bits y un residuo de 16 bits. El 68020 permite multiplicar operandos de 32 bits generando un producto de 64 bits y divisores de 32 bits, dando residuos y cocientes de 32 bits.

Tabla 4-2

Instrucciones de aritmética entera

| Mnemónico | Operación |
|-----------|---|
| ADD | Sumar fuente a destino |
| ADDA | Sumar fuente a registro de dirección |
| ADDI | Sumar dato inmediato a destino |
| ADDQ | Sumar dato corto a destino |
| ADDX | Sumar con bit extendido a destino |
| CLR | Poner a cero el operando |
| CMP | Comparar fuente con destino |
| CMPA | Comparar fuente c/reg. de dirección |
| CMPI | Comparar dato inmediato con destino |
| CMPM | Comparar memoria |
| CMP2* | Comparar registro para fronteras superior/ inferior |
| DIVS | Dividir con signo |
| DIVU | Dividir sin signo |
| DIVSL* | División larga con signo |
| DIVUL* | División larga sin signo |
| EXT | Extensión del signo |
| EXTB | Byte extensión del signo |
| MULS | Multiplicación con signo |
| MULU | Multiplicación sin signo |
| NEG | Negar (Complementar) |
| NEGX | Negar con extensión |
| SUB | Restar fuente al destino |
| SUBA | Restar fuente al registro de dirección |
| SUBI | Resta inmediata al destino |
| SUBQ | Resta corta al destino |
| SUBX | Resta con bit extendido al destino |

4.4 ARITMETICA BOOLEANA

Los MP's 68K disponen de las operaciones booleanas (lógicas) AND, OR, Exclusive OR y NOT. También incluida dentro de este tipo de instrucciones está una instrucción de comprobación (test) que compara un entero con cero, y fija los indicadores del registro del código de condición en relación con dicha comparación. La instrucción Scc comprueba los códigos de condición y pone el contenido de una posición de memoria toda a unos o a ceros, según el estado de los códigos.

Tabla 4-3

Instrucciones booleanas

| Mnemónico | Operación |
|-----------|---|
| AND | AND entre fuente y destino |
| ANDI | AND entre dato inmediato y destino |
| EOR | Exclusive OR entre fuente y destino |
| EORI | Exclusive OR entre dato inmediato y destino |
| NOT | NOT del destino (negación) |
| OR | OR entre fuente y destino |
| ORI | OR entre dato inmediato y destino |
| Scc | Comparar los códigos de condición y poner a uno el operando |
| TST | Comparar operando y poner a uno los códigos de condición |

4.5 DESPLAZAMIENTOS Y ROTACIONES

Los MP's 68K permiten desplazamientos aritméticos y lógicos (tanto a izquierda como a derecha) y rotaciones incluyendo o no al signo (también a izquierda o derecha).

Los desplazamientos aritméticos difieren de los desplazamientos lógicos en su actuación sobre los códigos de condición. También el desplazamiento aritmético a la derecha incluye el signo o bit más significativo, mientras que los desplazamientos lógicos a la derecha insertan un cero en el bit más significativo.

Una rotación sin extensión o ampliación sencillamente transfiere los bits de uno de los terminales del entero al terminal opuesto (final). La rotación con extensión difiere en que los bits que se transfieren externamente a la palabra lo hacen con la intervención del indicador de ampliación del registro de código de condición, con dicho bit de ampliación completando el límite opuesto de la palabra.

Tabla 4-4

Instrucciones de desplazamiento y rotación

| Mnemónico | Operación |
|-----------|--|
| ASL | Desplazamiento aritmético a izq. |
| ASR | Desplazamiento aritmético a der. |
| LSL | Desplazamiento lógico a izq. |
| LSR | Desplazamiento lógico a der. |
| ROL | Rotación izquierda |
| ROR | Rotación derecha |
| ROXL | Rotación izq. con bit de extensión |
| ROXR | Rotación der. con bit de extensión |
| SWAP | Intercambiar palabras de una palabra larga |

4.6 MANIPULACION DE BITS INDIVIDUALES

El 68K permite desde programa comprobar, poner a cero, poner a uno y cambiar los bits individuales de un valor entero. Las operaciones de puesta a cero, a uno y de cambio son utiles en sistemas multitarea, ya que combinan en la misma instruccion una lectura y una escritura, haciendo posible que un programa controle el acceso a una estructura de datos.

Estas instrucciones no son, sin embargo, "indivisibles", un programa no deberia depender de ellas para inmovilizar a otro procesador en un sistema multiprocesador, ya que la ejecucion del procesador podria ceder el bus a un segundo procesador dentro de la ejecucion de la instruccion. Vea las instrucciones incluidas bajo el titulo "Comunicaciones multitarea/multiprocesador" en donde se citan las instrucciones indivisibles.

Tabla 4-5

Instrucciones de manipulacion de bits

| Mnemonic | Operacion |
|----------|------------------|
| BCHG | Cargar bit |
| BCLR | Poner bit a cero |
| BSET | Poner bit a uno |
| BTST | Comprobar bit |

4.7 MANIPULACION DE CAMPOS DE BITS

El 68020 permite a un programa modificar una cadena o "campo" de bits consecutivos así como a bits individuales. Las instrucciones pueden operar en un campo de hasta 32 bits de longitud. Un programa puede insertar bits, extraer bits, encontrar cadenas de bits, poner bits a uno, poner bits a cero, comprobar bits y cambiar bits.

Tabla 4-6

Instrucciones de campos de bits

| Mnemdnico | Operación |
|-----------|---|
| BFCHG | Cargar campo de bit |
| BFCLR | Poner a cero campo de bit |
| BFEXTS | Extraer y extender en signo el campo bit |
| BFEXTU | Extraer y extender a ceros el campo bit |
| BFFFO | Encontrar el primer bit 1 en el campo bit |
| BFINS | Insertar campo bit |
| BFSET | Poner a uno el campo bit |
| BFTXT | Comprobar el campo bit |

4.8 MANIPULACION EN DECIMAL CODIFICADO BINARIO (BCD)

De igual forma a como opera con enteros, el 68K permite programar la suma y resta de cadenas en código binario codificado decimal (BCD). El 68020 incluye instrucciones para convertir cadenas de BCD entre las formas de empaquetado y desempaquetado.

Tabla 4-7

Instrucciones decimal codificado binario (BCD)

| Mnemónico | Operación |
|-----------|--------------------------------|
| ABCD | Sumar fuente al destino |
| NBCD | Negar destino |
| *PACK | Empaquetar fuente a destino |
| SBCD | Restar fuente al destino |
| *UNPK | Desempaquetar fuente a destino |

4.9 CONTROL DEL FLUJO DEL PROGRAMA

Las instrucciones del flujo del programa permiten a un programa hacer ramificaciones condicionales e incondicionales a cualquier punto del programa en código de máquina. Las instrucciones permiten realizar bifurcaciones referidas al PC (Program Counter) y bifurcaciones absolutas. Todos los miembros 68K permiten llamadas a subrutinas y retornos utilizando la pila, además, el 68020 permite estructuras formales en la pila para transferir parámetros a las subrutinas.

Tabla 4-8

instrucciones de flujo del programa

| Mnemónico | Operación |
|-----------|---|
| Bcc | Bifurcación condicional |
| BRA | Bifurcación incondicional |
| BSR | Bifurcación a subrutina |
| *CALLM | Llamada a módulo |
| DBcc | Comprobar, decrementar y saltar |
| JMP | Saltar a dirección |
| JSR | Saltar a subrutina |
| NOP | No operativa |
| RTD ** | Retornar y recuperar la pila |
| RTE + | Retornar de excepción |
| *RTM | Retornar de módulo |
| RTR | Retornar y restaurar códigos de condición |
| RTS | Retornar de subrutina |
| | |
| * | solo en el 68020 |
| ** | solo 68010-68020 |
| + | instrucción privilegia |

4.10 CONTROL DEL SISTEMA

Las instrucciones de control del sistema incluyen operaciones privilegiadas (que solo pueden ejecutarse en modo supervisor), así como instrucciones que permiten la interconexión entre programas en modo usuario y en modo supervisor, e instrucciones para manipular el byte código de condición del registro de estado.

Tabla 4-9

Instrucciones de control del sistema

| Mnemónico | Operación |
|-----------|---|
| ANDI | AND inmediato con el registro estado / registro código de condición |
| BKPT | Intercepción de punto de ruptura |
| CHK | Intercepción de operando con frontera superior errónea |
| CHK2* | Intercepción de operando con fronteras erróneas |
| EORI | Exclusive OR inmediato con estado |
| ILLEGAL | Intercepción de instrucción ilegal |
| MOVE | Transferencia de/hacia registro de estado/registro de código de condición |
| MOVEC+ | Transferencia de/hacia registro de control |
| MOVES+ | Transferencia de/hacia espacio de direcciones |
| RESET+ | Validar línea RESET(reiniciar) |
| STOP+ | Detener el procesador |
| TRAP | Intercepción incondicional |
| TRAPcc* | Intercepción bajo condiciones |
| TRAPV | Intercepción si hay desbordamiento |

* solo el 68020
+ instrucción privilegiada

4.11 COMUNICACIONES MULTITAREA/MULTIPROCESADOR

En un sistema multitarea y multiprocesador, una tarea dada o un procesador dado debe ser capaz de bloquear otra tarea o procesador. Sin esta capacidad, un programa necesitaría una gran l6gica para prevenir que un segundo programa acceda a la misma posici6n de memoria cuando el primero est6 accediendo a ella. Los procesadores 68K permiten realizar unos ciclos indivisibles "leer - modificar - escribir" que previenen al procesador de ceder el bus durante la ejecuci6n de una instrucci6n.

Tambi6n en este grupo se incluyen instrucciones para interfaz con el coprocesador. Un coprocesador puede ser una unidad en punto flotante, una unidad de manejo de memoria, una unidad de control de E/S o elementos similares.

El 68020 es el 6nico miembro de la familia que admite esta interfaz hardware para coprocesador.

Tabla 4-10
Instrucciones multitarea/multiprocesador

| Mnem6nico | Operaci6n |
|------------|---|
| CAS* | Comparar e intercambiar con operando |
| CAS2* | Comparar e intercambiar con operandos |
| cpBcc* | Ramificar bajo condici6n del coprocesador |
| cpDBcc* | Comprobar coprocesador, decrementar y ramificar |
| cpGEN* | Instrucci6n general de coprocesador |
| cpRESTORE* | Restaurar estado del coprocesador |
| cpSAVE* | Salvaguardar el estado del coprocesador |
| cpScc* | Comprobar situaci6n del coprocesador |
| cpTRAPcc* | Interceptar con situaciones del coprocesador |
| TAS | Comprobar y poner a uno operando |

* solo el 68020

4.12 PRECAPTACION DE INSTRUCCION / SEGMENTACION / ENLAZAMIENTO / ACELERACION

Segmentación ("pipelining")

La familia de procesadores 68K implanta distintos grados de "segmentación". La segmentación implica la captación y ejecución concurrente (al mismo tiempo) de instrucciones. Debido a este solapamiento, un procesador con esta posibilidad puede ejecutar instrucciones más rápidamente que aquellos procesadores que captan y ejecutan instrucciones secuencialmente (en serie).

Las unidades 68000, 68008, 68010 y 68012 utilizan una precaptación de dos palabras. El procesador comienza captando la palabra instrucción. Cuando la decodificación comienza, el procesador capta o captura la siguiente palabra de una dirección de memoria, de forma que cuando se inicia la ejecución de la instrucción, han sido ya captadas o cargadas dos palabras de memoria. Y a medida que el procesador va utilizando palabras, va captando o cargando nuevas palabras de la memoria.

Observe que si la instrucción causa una ramificación, la palabra que sigue a la instrucción (ya almacenada internamente en el procesador) no es utilizable. En una ejecución normal, sin embargo, el incremento en velocidad debido a la precaptación es mucho mayor que las pérdidas debidas a las captaciones que no son utilizadas.

El 68020 opera similarmente, excepto que usa una precaptación de tres palabras, que permite operaciones simultáneas de hasta tres palabras de una instrucción única o de tres instrucciones consecutivas de una palabra.

Instrucciones enlazables

Tabla 4-11 Instrucciones enlazables(68010,68012)

| | | | | | |
|------|------|------|------|------|------|
| ABCD | ASL | EOR | NEG | ROR | SUBA |
| ADD | ASR | LSL | NEGX | ROXL | SUBX |
| ADDA | CLR | LSR | NOT | ROXR | TST |
| ADDX | CMP | MOVE | OR | SBCD | |
| AND | CMPA | NBCD | ROL | SUB | |

Las unidades 68010 y 68012 mejoran la idea de segmentación. Como el procesador precapta dos palabras en avance a la ejecución. Motorola añade a sus procesadores la "astucia" de comprobar si una de las instrucciones es DBcc. En ciertos casos, el procesador puede ejecutar la instrucción lazo sin recaptar la instrucción de memoria.

La instrucción DBcc tiene tres operandos: un contador de lazo, una condición de ramificación y un desplazamiento de ramificación. En el caso especial de que una instrucción de una única palabra esté seguida por una instrucción DBcc que ramifica hacia atrás a dicha instrucción de una sola palabra, el procesador no necesita ninguna captación de memoria mientras se esté en el interior del lazo.

Acelerador de instrucciones

El 68020 tiene un acelerador de memoria de 256 byte localizado en el propio circuito integrado. El procesador copia instrucciones en la memoria rápida (acelerador) cuando inicia la captación de ellas de la memoria. El procesador mantiene un seguimiento de las direcciones (instrucciones) que tiene en su acelerador, así que antes de captar de nuevo de memoria, puede captar la instrucción del acelerador.

Esto incrementa la velocidad de ejecución del procesador ya que no necesita consumir un ciclo de memoria para captar la instrucción. En un entorno multiprocesador esto puede incrementar la velocidad de todos los procesadores, ya que mejorará (disminuirá) el uso del bus del sistema para la captación de instrucciones.

Observe que, por eficiencia, el procesador no introduce operandos en el acelerador. También un programa supervisor puede optar por inhibir el acelerador. Esto puede ser útil en el interior de una rutina de excepción, ya que un programa de usuario puede ser ejecutado fuera del acelerador durante la excepción.

CONTENIDO

PARTE V

DESCRIPCION DE SEÑALES.

| | |
|--|----|
| 5.1 Introducci3n. | 75 |
| 5.2 Señales para el 68000,68008,68010 y 68012. | 76 |
| 5.3 Señales para el 68020. | 83 |

5.1 INTRODUCCION

En esta parte V se discutirán las señales de entrada y salida requeridas por todos los miembros de la familia 68K. Aunque los miembros de la familia son muy parecidos, en el siguiente punto se discutirán todos los miembros que únicamente utilizan un bus de datos de 8 o 16 bits y luego en el siguiente punto los que utilizan un bus de datos de 32 bits (únicamente el 68020).

En los puntos posteriores se darán figuras de cada miembro de la familia 68K que van a mostrar los grupos funcionales de las señales de cada uno de ellos.

Los grupos funcionales de señales los constituyen los siguientes señales :

- 1.- Bus de datos
- 2.- Bus de direcciones
- 3.- Dato valido
- 4.- validaciones de datos superior/inferior
- 5.- Lectura/escritura
- 6.- Reconocimiento de transmisión de datos
- 7.- Códigos de función
- 8.- Señales de interrupción
- 9.- Error en bus
- 10.-Parada de procesador
- 11.-Reiniciar
- 12.-Reloj
- 13.-Señales de arbitraje del bus
- 14.-Señales de la familia 6800

5.2 SEÑALES PARA EL 68000, 68008, 68010 Y 68012.

Las figuras 5-1, 5-2, 5-3 y 5-4 muestran los grupos funcionales de las señales en cada uno de los microprocesadores de 16-bits.

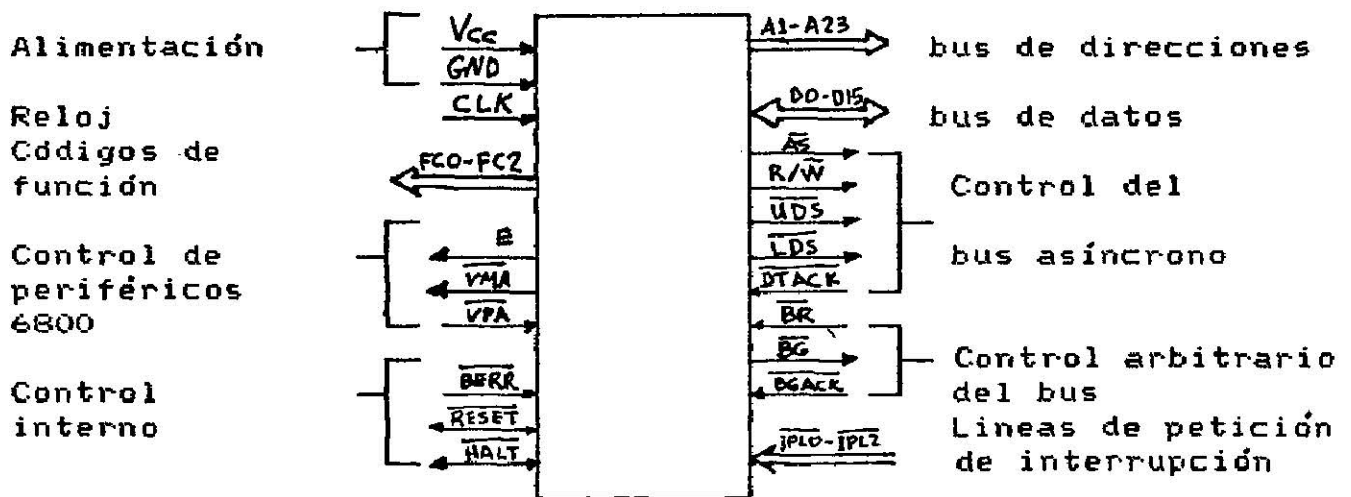


Figura 5-1.
Los grupos funcionales de las señales del 68000.

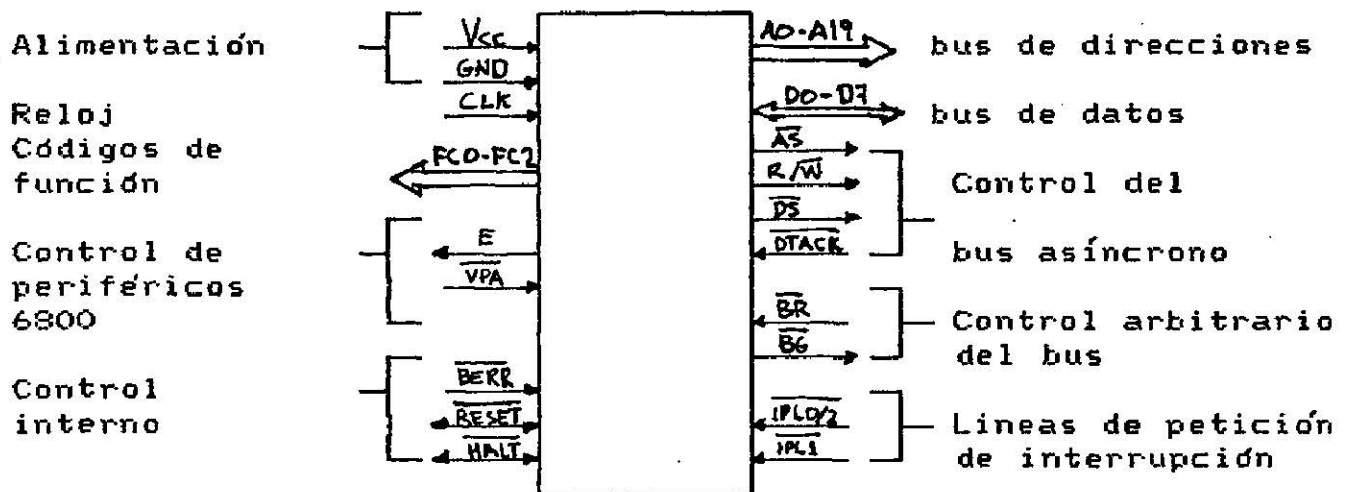


Figura 5-2.
Los grupos funcionales de las señales del 68008.

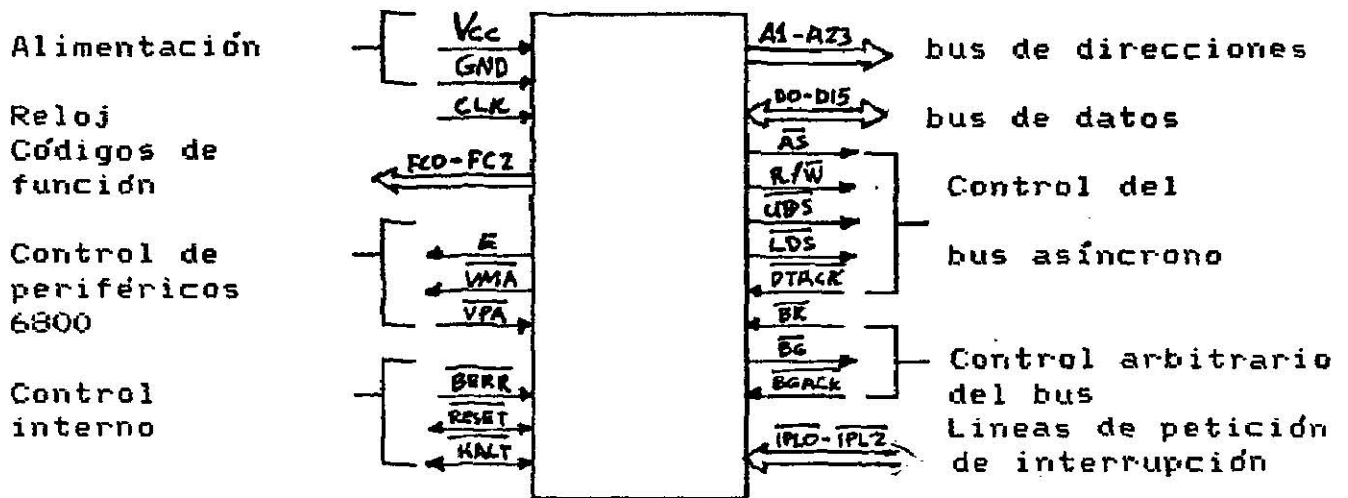


Figura 5-3. Los grupos funcionales de las señales del 68010.

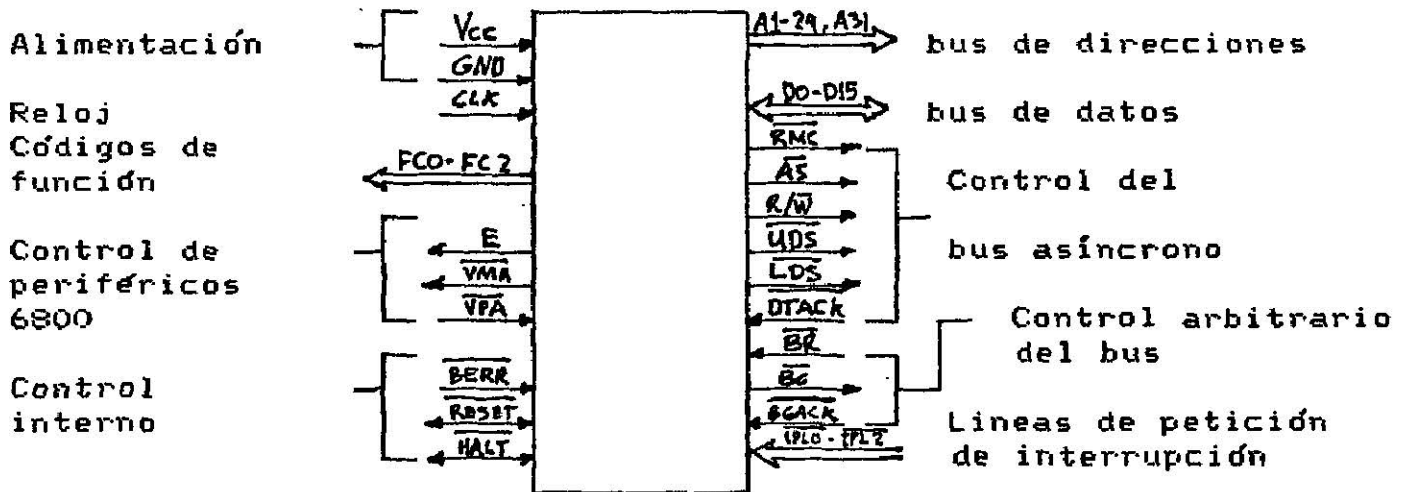


Figura 5-4. Los grupos funcionales de las señales del 68012.

BUS DE DATOS

Las líneas D0-D15 constituyen el bus bidireccional de datos (en el 68008 este es D0-D7).

BUS DE DIRECCIONES

Las líneas A1-A23 constituyen el bus de salida de direcciones (A0-A19 en el 68008; A1-A29 y A23 en el 68012). Debido al encapsulado, el conjunto de procesadores 68K no requiere multiplexado de las líneas de datos y direcciones, como sucede en algunos otros procesadores conocidos (Intel 8086, 28000, etc.).

DATO VALIDO

La señal dato valido (DS o "Data Strobe") indica que el bus de datos del 68008 está en uso.

VALIDACIONES DE DATOS SUPERIOR/INFERIOR

Los procesadores 68000, 68010 y 68012 utilizan las validaciones de datos superior (UDS o "Upper Data Strobe") e inferior (LDS o "Lower Data Strobe") para determinar si los datos que están siendo transferidos en un momento dado son los superiores (byte más significativo), los inferiores (byte menos significativo) o ambos bytes del bus de datos de 16 bits.

La tabla 5.1 define el significado de UDS y LDS en relación con el bus de datos.

Tabla 5.1

UDS y LDS en el 68000, 68010 y 68012.

| UDS | LDS | R/W | D8-D15 | D0-D7 | Operacion |
|------|------|------|-------------------|-------------------|-----------------|
| Alto | Alto | Alto | Bit de datos 8-15 | Bit de datos 0-7 | Lectura de pal. |
| Bajo | Bajo | Alto | Bit de datos 8-15 | Bit de datos 0-7 | Lectura de byte |
| Alto | Bajo | Alto | Bit de datos 8-15 | Bit de datos 0-7 | Lectura de byte |
| Bajo | Alto | Alto | Bit de datos 8-15 | Bit de datos 0-7 | Escritu de pal. |
| Bajo | Bajo | Bajo | Bit de datos 8-15 | Bit de datos 0-7 | Escritu de byte |
| Alto | Bajo | Bajo | Bit de datos 0-7 | Bit de datos 0-7 | Escritu de byte |
| Bajo | Alto | Bajo | Bit de datos 8-15 | Bit de datos 8-15 | Escritu de byte |

Dato de entrada o salida no valida

Para acceder a un byte de datos par, el procesador valida UDS y niega LDS, la transmisión de datos ocurre en los bits 0-7 del bus de datos. Para acceder a un byte de datos impar, el procesador valida LDS y niega UDS; la transmisión de datos ocurre en los bits 0-7 del bus de datos. Para acceder a una palabra completa, el procesador valida LDS y UDS; la transmisión de datos ocurre en los 16 bits del bus de datos.

Este esquema de codificación está en relación con la restricción de que el acceso a las palabras de datos ocurran sólo con direcciones límites pares.

La validación de la señal de dirección (AS o "Address Strobe") es una salida del procesador activo que indica que los datos situados en el bus de direcciones y las validaciones de datos son validados.

LECTURA/ESCRITURA

La señal leer/escribir (R/W o "Read/Write") indica si la transmisión a través del bus corresponde a un ciclo de lectura o a uno de escritura. Esta señal es alta para una lectura y baja para una escritura.

RECONOCIMIENTO DE TRANSMISION DE DATOS

DTACK ("Data Transfer Acknowledge") es la señal de entrada de reconocimiento de transmisión de datos. La lógica externa (memoria, por ejemplo) debe validar DTACK para indicar al procesador que se ha recibido el dato del bus de datos (para una escritura) o para indicar que ha sido situado el dato solicitado en el bus de datos (para una lectura). Si es necesario, el procesador automáticamente inserta "estaos de espera" en este ciclo hasta recibir DTACK. Esto permite acceder a periféricos de diversas velocidades.

CODIGOS DE FUNCION

FC0, FC1 y FC2 son las salidas de código de función. Estas salidas (correctas sólo cuando AS las valida) identifican el tipo de actividad en curso en el bus, encargada por el 68K, como resume la Tabla 5-2.

Tabla 5-2
Resumen de codigos de funcion.

| FC2 | FC1 | FC0 | TIPO DE CICLO MAQUINA |
|--|-----|-----|---|
| 0 | 0 | 0 | |
| 0 | 0 | 1 | Acceso a la memoria de datos del usuario |
| 0 | 1 | 0 | Acceso a la memoria del programa del usuario |
| 0 | 1 | 1 | |
| 1 | 0 | 0 | |
| 1 | 0 | 1 | Acceso a la memoria de datos del supervisor |
| 1 | 1 | 0 | Espacio de CPU (recomocimiento de interrupcion) |
| 1 | 1 | 1 | Acceso a la memoria del programa del supervisor |
| <div style="border: 1px solid black; display: inline-block; width: 20px; height: 15px; vertical-align: middle;"></div> Reservado, actualmente indefinido | | | |

Como puede ver en la tabla, las salidas del código de función separan los accesos a memoria con el criterio de accesos de datos frente a accesos de programa y de usuario frente a accesos de supervisor. Esto permite a una unidad de manejo de memoria (tal como el circuito 68451) controlar el acceso a memorias protegidas y sin protección.

Una quinta codificación de los códigos de función es definida como "espacio CPU". Hay que señalar que la documentación del 68000 y del 68008 denomina a este código "reconocimiento de interrupción". En general, el procesador utiliza el espacio CPU para comunicaciones periféricas además de para las instrucciones normales de lectura y escritura, por ejemplo para reconocer una interrupción.

El 68010 y el 68012 permite a un programa definir el código de función de salida con una instrucción MOVE a través de los registros de código de función (SFC y DFC).

SENALES DE INTERRUPCION

Cuando un periférico desea interrumpir el proceso, el periférico valida una o más líneas de nivel de interrupción IPL0, IPL1 e IPL2. Estas tres entradas representan el valor binario del nivel de la interrupción solicitada.

Recordará de una discusión anterior que el registro de estado mantiene una máscara de interrupción de 3 bits que determina los niveles de interrupción que pueden iniciar un procesamiento excepcional. Para

iniciar una interrupción, un periférico valida (transmite) el nivel de prioridad en las líneas de interrupción. Si la máscara de interrupción está fijada de forma que acepta este nivel reconocerá la interrupción a través de las líneas de código de función. Los procesos excepcionales se discuten con más detalle en la parte VI.

El 68008 combina IPL0-IPL2 en una señal. Esto limita a este procesador los niveles de interrupción a 0,2,5 y 7.

ERROR EN BUS

BERR es la señal de entrada de error en bus ("Bus Error"). Cuando esta señal es validada, el 68K inicia una secuencia de procesamiento excepcional. El objetivo de BERR es informar al procesador que un dispositivo externo no responde a una operación de lectura o escritura (esto es, no ha validado la señal DTACK). BERR también puede usarse por una unidad de manejo de memoria (MMU o "Memory Management Unit") para indicar que el procesador tiene intención de acceder a la memoria protegida.

Para generar BERR en el primer caso, el sistema debe tener circuitos externos que generan dicha señal. En el último caso, la línea de entrada BERR normalmente debería ser conectada a la línea de fallo de página o línea equivalente del MMU.

El BERR produce resultados ligeramente diferentes en los procesadores de memoria virtual, 68010 y 68012. La parte VI presenta esta información en detalle.

PARADA DEL PROCESADOR

La señal HALT ("parada") tiene diversas funciones. Es tanto señal de entrada como de salida y puede actuar sola o en conjunción con otras líneas. Cuando se valida sola como línea de entrada, HALT fuerza al procesador a introducirse en un estado inactivo donde permanece hasta que la señal HALT es negada.

HALT puede también actuar como una línea de salida indicando que el procesador ha suspendido la ejecución de una instrucción -- por ejemplo, a causa de un doble fallo del bus --. Los circuitos externos pueden así detectar esta circunstancia.

HALT y BERR pueden usarse juntas. Cuando HALT

es validada junto con BERR, el procesador reejecuta el último ciclo.

REINICIAR

La señal RESET ("reiniciar"), como HALT, es bidireccional. Si el procesador ejecuta una instrucción RESET, valida esta línea. RESET también puede funcionar como línea de entrada cuando se usa en conjunción con la línea HALT. Cuando se validan juntas, el 68K se inicializa (incluyendo una intercepción a través de vectores de reinicio).

RELOJ

Para funcionar, el procesador requiere una señal de temporización. La entrada reloj("clock") CLK, suministra dicha señal.

SENALES DE ARBITRAJE DEL BUS

Las señales de arbitraje del bus son BR ("Bus Request" o petición del bus), BG ("Bus Grant" o cesión del bus), BGACK("Bus Grant Acknowledge" o reconocimiento de cesión del bus). Estas señales se usan en sistemas donde otros dispositivos, tales como controladores DMA, pueden actuar como controladores del bus, solicitando dicho control. Los dispositivos externos solicitan el acceso al bus del sistema validando la señal BR. El 68K, después de haber terminado el ciclo de bus en curso, lo cederá.

El procesador da como salida la señal BG para indicar al dispositivo solicitante que el bus se le cederá al final del ciclo en curso. Después el dispositivo conforma que el bus le ha sido cedido por el 68K validando la señal BGACK. El dispositivo mantiene la señal BGACK a uno hasta que se hace con el bus. El 68008 no dispone de la señal BGACK.

SENALES DE LA FAMILIA 6800

Puesto que la familia 68K es un pariente lejano de la serie inicial de MP's 6800, Motorola decidió incluir tres señales de bus de forma que los diseñadores pudiesen construir sistemas utilizando la amplia gama disponible de circuitos 6800. Estas señales son: señal disponible (E o "Enable"), señal dirección de periférico válida (VPA o "Valid Peripheral Address") y dirección de memoria válida (VMA o "Valid Memory Address").

A diferencia del conjunto 68K, el 6800 utiliza una interfaz de bus síncrona en lugar de asíncrona. Esto significa que las transmisiones de datos a través del sistema están sincronizadas a través de una señal de reloj. La señal de salida E del 68K proporciona esta señal. La frecuencia de E es igual a una décima parte de la señal de entrada CLK en el 68K; el período de E es igual a 10 CLK períodos, E está baja durante seis ciclos CLK y está alta durante cuatro ciclos CLK.

La señal VPA es usada por los dispositivos de la familia 6800 incluidos en el sistema para informar al 68K que requieren una transmisión de datos del tipo 6800. El sistema debe contener circuitos lógicos externos que determinen qué direcciones corresponden a periféricos de la familia 6800. Esta lógica debe validar la señal VPA.

Cuando el 68K recibe la señal VPA, altera su temporización en la transmisión de datos, sincronizándose con la señal E. El procesador después valida la señal VMA y la transmisión de datos comienza.

El 68008 no dispone de señal VMA.

5.3 SEÑALES 68020

A causa de sus potentes características, el 68020 utiliza algunas ampliaciones y modificaciones en su conjunto de señales de entrada y salida. Revisando las señales 68020 se encuentran muchas semejanzas en los nombres de la señal y en las funciones las principales diferencias existentes radican en que los buses datos y de direcciones son de 32-bits completos, y la carencia de las líneas de la familia 6800.

La figura 5-5 muestra los grupos funcionales de las señales en el 68020.

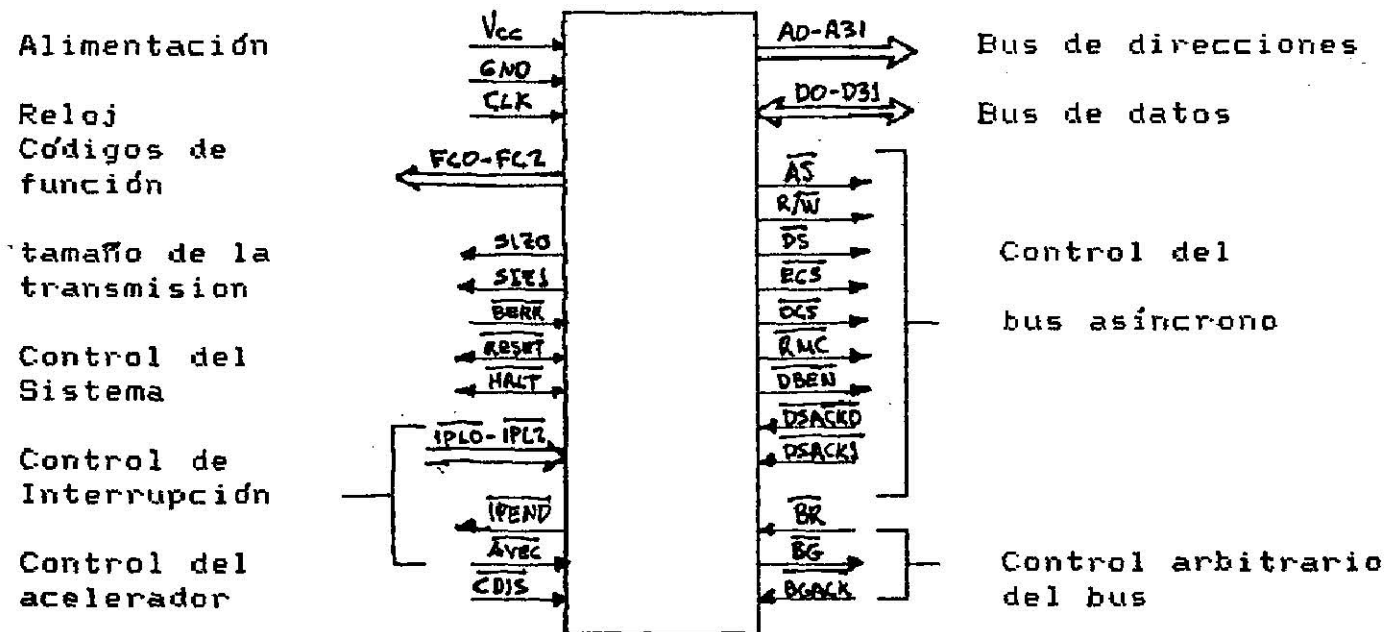


Figura 5-5

Grupos funcionales de señales del 68020.

BUS DE DATOS

El bus bidireccional de datos es de 32-bits de ancho (D0-D31). El 68020 es capaz de ajustar dinámicamente el ancho de bus, esto es, pueden transferirse datos de 1, 2, 3 o 4 bytes de ancho, utilizando cualquier parte del bus de datos.

BUS DE DIRECCIONES

El bus de direcciones es un bus de 32 bits completos (A0-A31). Observe la presencia de A0 (recuerde que algunos de los primeros modelos no usaban A0). Este gran bus da un espacio potencial de direcciones de 4 gigabyte.

TAMANO DE TRANSFERENCIA

El medio de controlar la anchura del bus de datos para una transferencia dada viene dado por las señales de tamaño (size) de transferencia SIZ0 y SIZ1. La combinación de estas señales indica el número de bytes utilizados en la transferencia de datos.

LEER/ESCRIBIR

Como otros miembros de la familia el 68020 utiliza varias señales de control de bus para manejar la interfaz asíncrona con los periféricos. La señal leer/escribir (R/W o "Read/Write") indica la dirección de la transferencia de datos. Observe que esta línea tiene dos estados de validación nivel alto significa leer y nivel bajo significa escribir.

DATO HABILITADO

La línea de dato habilitado (DS o "Data Strobe") indica que el bus de datos está en uso. Si se valida durante un ciclo de lectura, DS indica que el periférico debe cargar el bus de datos; si es validada durante un ciclo de escritura, indica que el periférico puede captar el dato presente en el bus.

DIRECCION HABILITADA

Dirección habilitada (AS o "Address Strobe") es una señal de salida que indica que el procesador ha sido situado, en las líneas correspondientes, códigos de función válidos, información de dirección, tamaño e información del estado de lectura/escritura.

CODIGOS DE FUNCION

Las señales de código de función (FC0, FC1 y FC2) identifican el estado en curso del procesador y el espacio de dirección requerido. La combinación FC0-FC2 (como se muestra en la figura 5-2) indica si el espacio de dirección (código) es de datos o programa y si dicho espacio es de usuario o de supervisor.

Una quinta combinación implica "espacio CPU". Bajo el espacio CPU se incluyen cuatro subtipos, incluyendo reconocimiento de punto de ruptura, operación módulo, comunicación con procesador y reconocimiento de interrupción. El procesador utiliza el bus de dirección para distinguir entre esos modos así como para pasar a su vez información adicional a los periféricos. La figura 5-6 muestra los códigos del bus de direcciones para el espacio CPU.

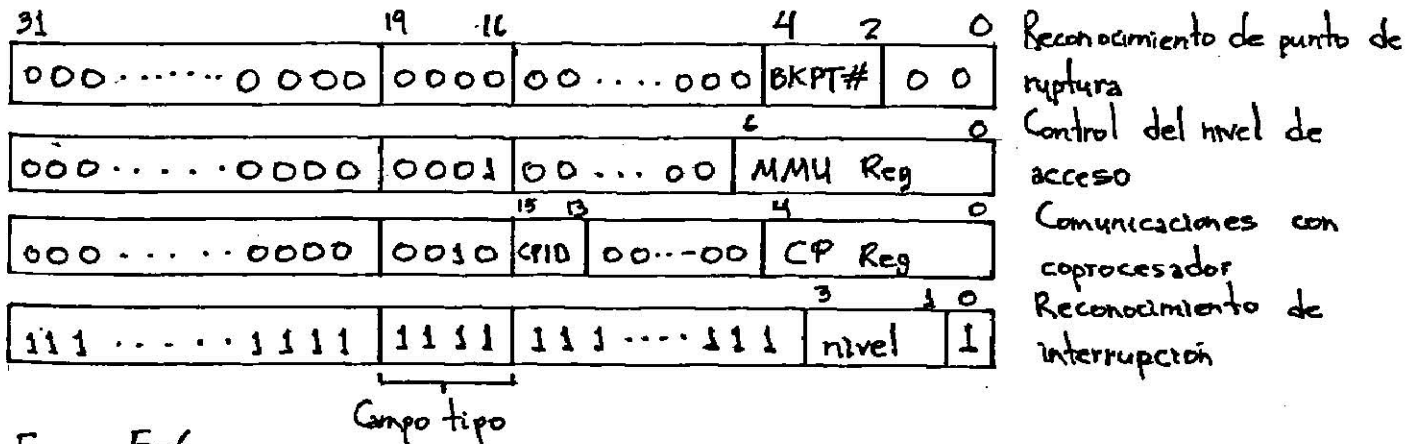


Figura 5-6.
Códigos de función para espacio de CPU.

Hay que indicar que una rutina del supervisor puede requerir un espacio específico de direcciones para todas las instrucciones MOVES, suministrando el valor de código de función de los registros fuente y destino del código de función alternativo (SFC y DFC).

RECONOCIMIENTO DE TRANSMISION DE DATOS Y TAMAÑO

Las señales de entrada de reconocimiento de transmisión de datos y tamaño ("Data Transfer and Size Acknowledge" o DSACK0 Y DSACK1) indican al procesador que se ha completado una transmisión de

datos. Para un ciclo de lectura, esto significa que el periférico ha ubicado datos en el bus y el procesador debe captarlos. Para un ciclo de escritura, esto señala que el periférico ha leído los datos y el procesador puede continuar.

COMIENZO DE CICLO

Con objeto de controlar mejor la comunicación entre procesador y periféricos, el 68020 incluye dos señales de salida de comienzo de ciclo (ECS o "External Cycle Start") durante el inicio de la primera mitad de la señal de reloj de todo ciclo de bus. La señal de inicio de ciclo de operando (OCS u "Operand Cycle Start") actúa similarmente, excepto que el procesador la valida sólo durante el primer ciclo de bus de una transmisión de operando.

SEÑAL LECTURA/MODIFICACION/ESCRITURA

La señal lectura/modificación/escritura (RMC o "Read-Modify-Write") indica que el bus está retenido por un control externo. Las únicas instrucciones del 68020 que actúan sobre esta señal son la de comprobar y fijar (TAS) y la de comparar e intercambiar (CAS y CAS2). Esta posibilidad de bloquear el bus asegura la integridad de los datos en un sistema multiprocesador.

REGISTRO DE DATOS HABILITADO

La señal de salida de registro de datos habilitado (DBEN o "Data Buffer Enable") pone en condiciones los registros de datos externos para transmisión de datos. Permite cambios en la señal R/W sin la posibilidad de contención del registro externo.

INHABILITACION DEL ACELERADOR DE INSTRUCCIONES

La señal de entrada de inhabilitación del acelerador (CDIS o "Cache Disable Input Signal") hace que no se pueda utilizar el acelerador de instrucciones del circuito integrado.

PETICION DE INTERRUPCION

El 68020 soporta siete niveles de interrupción. Las líneas de entrada de petición de interrupción (IPLO, IPL1 e IPL2) indican el deseo de un periférico de interrumpir al procesador. Una vez recibida una petición de interrupción, el procesador compara esas líneas con la máscara de prioridad de

interrupciones en el registro de estado.

INTERRUPCIONES PENDIENTES

La salida de interrupción pendiente (IPEND) indica que el procesador ha recibido y decodificado un nivel de interrupción (a través de IPLO-IPL2) mayor que el nivel almacenado en la máscara de interrupción del registro de estado o que el procesador ha recibido una interrupción no enmascarable.

AUTOVECTOR

Después que el procesador reconoce una señal de interrupción, el dispositivo que interrumpe envía un número de vector indicando la entrada en la tabla de vectores donde el procesador puede encontrar la adecuada rutina de interrupción. El 68020 también permite la técnica de "autovectores" a través de la línea de entrada autovector (AVEC).

Cuando AVEC es validada, el procesador capta el vector entre los vectores automáticos de la tabla, en lugar de esperar a que el dispositivo le especifique uno. Los procesadores primeros del conjunto 68K utilizaban VPA para solicitar la función autovector.

ERROR EN BUS

BERR es la señal de entrada de error en bus (o "Bus Error"). Cuando esta señal es válida, el 68020 inicia una secuencia de procesamiento excepcional. El objetivo de BERR es informar al procesador que un dispositivo externo ha fallado en responder a una operación de lectura o escritura (esto es, no ha valido DSACK0/DSACK1). BERR puede también usarse para una unidad de manejo de memoria (MMU o "Memory Management Unit") para indicar que el procesador ha intentado un acceso a la memoria protegida.

El sistema debe tener hardware externo para generar la señal BERR en el primero de los casos. En el último de los casos, la línea de entrada BERR normalmente estaría conectada a la línea del fallo de página, u otra equivalente, de MMU.

PARADA DEL PROCESADOR

La señal HALT (o "parada") realiza diversas funciones. Es tanto una señal de entrada como de salida, y puede actuar sola o en conjunción con otras líneas. Cuando se valida individualmente como

línea de entrada, HALT fuerza al procesador a entrar en un estado inactivo donde permanece hasta que la señal HALT es negada.

HALT puede también funcionar como señal de salida indicado que el procesador ha cesado la ejecución de instrucción -- por ejemplo, a causa de una situación de doble fallo de bus --. La lógica externa, puede detectar esta circunstancia.

HALT y BERR pueden usarse conjuntamente. Cuando HALT es validada simultáneamente con BERR, el procesador reejecuta el último ciclo de bus.

REINICIAR

RESET ("reiniciar"), como HALT, es una señal bidireccional. Si el procesador ejecuta una instrucción RESET, valida esta línea. RESET puede también actuar como una línea de entrada, que, cuando se valida, provoca que el 68020 intercepte con el vector reiniciar.

RELOJ

Para funcionar el procesador requiere una señal de temporización. La señal de entrada de reloj ("clock"), CLK, realiza esta función.

SENALES DE ARBITRAJE DE BUS

Las señales BR (petición de bus), BG (cesión de bus) y BGACK (reconocimiento de cesión de bus) son las señales de arbitraje del bus. Estas señales se usan en sistemas donde otros dispositivos, tales como controladores DMA, pueden actuar como controladores de bus requiriendo el control del bus. Los dispositivos externos solicitan acceder al bus del sistema validando la señal BR. El 68020 cederá el bus después de acabar el ciclo de bus en curso.

El procesador proporciona la señal BG para indicar al dispositivo solicitante que el bus le será cedido al final del ciclo en curso. El dispositivo después reconocerá que el 68020 le ha traspasado el control del bus validando la señal BGACK. El dispositivo mantiene BGACK validado hasta haber terminado con el bus.

CONTENIDO

P A R T E V I

P R O C E S A M I E N T O S E X C E P C I O N A L E S

| | |
|---|-----|
| 6.1 Introducci3n. | 91 |
| 6.2 Modos de Operar. | 92 |
| 6.3 Tipos de excepciones. | 93 |
| 6.4 Prioridades de excepci3n. | 95 |
| 6.5 Tabla de vectores de excepci3n. | 97 |
| 6.6 Estructuras de pila. | 99 |
| 6.7 Secuencias de procesamiento de excepciones. | 105 |
| 6.8 Intercepci3n("trap") de instrucciones. | 105 |
| 6.9 Instrucciones Ilegales/no existentes. | 106 |
| 6.10 Error de direccionamiento. | 106 |
| 6.11 Traza. | 107 |
| 6.12 Puntos de ruptura. | 107 |
| 6.13 Error de direccionamiento. | 108 |
| 6.14 Interrupciones. | 109 |
| 6.15 Error en bus. | 110 |
| 6.16 Reinicio("Reset"). | 111 |

6.1 INTRODUCCION

En esta parte, se va a tratar de los procesos excepcionales. Un procesamiento excepcional es el proceso de ejecución que ocurre a causa de eventos especiales internos al microprocesador. Los "eventos especiales" que pueden iniciar un procesamiento de excepción incluyen errores en direccionamiento, fallos en el bus, intentos de ejecutar instrucciones privilegiadas estando en modo usuario e intentan dividir por cero. También pueden provocar procesamientos excepcionales, eventos no relacionados con errores. Por ejemplo, interrupciones desde periféricos, reinicios hardware e interrupciones programadas son causas del inicio de un procesamiento excepcional.

Los puntos que siguen trataran en particular las causas del inicio, su secuencia, y todo lo relacionado con un procesamiento excepcional.

6.2 MODOS DE OPERAR

Antes de proceder a describir el sistema de procesamientos excepcionales, recordaremos los modos de funcionamiento del 68K, vistos en la Parte II, ya que estos afectan a los procesamientos excepcionales. Como se mencionó anteriormente, el 68K puede operar o funcionar en modo supervisor o en modo usuario. Cuando el 68K es reiniciado con una entrada RESET, comienza a funcionar en modo supervisor. El procesador permanece en este modo hasta que se ejecuta una de las siguientes instrucciones:

- * RET - retorno de excepción
- * MOVE - transferir hacia registro de estado
- * ANDI - "and" inmediato con el reg. de estado
- * EORI - "exclusive or" inmediato con el registro de estado

Ninguna de esas instrucciones provoca automáticamente la transición al modo usuario de operación, más bien en ellas son capaces de cambiar el estado del bit S del registro de estado, con el cual si se reinicia ("reset") se fuerza al 68K a comenzar a operar en el modo usuario.

Una vez que el 68K está operando en modo usuario, la única cosa que puede hacerle volver al modo supervisor es una excepción. Cuando el procesador termina el procesamiento excepcional, éste ejecuta una instrucción retorno de excepción (RTE) para regresar al modo usuario.

6.3 TIPOS DE EXCEPCIONES

Las excepciones se producen de varias formas, que pueden dividirse en dos categorías generales: excepciones generadas internamente que resultan de la ejecución de ciertas instrucciones o de errores detectados internamente y excepciones generadas externamente, que incluyen errores en bus, reinicios y peticiones de interrupciones.

Excepciones generadas internamente

También se pueden dividir las excepciones generadas internamente, a las que el 68K responde, en tres categorías: errores internos, intercepciones de instrucciones y la función traza. Los siguientes son errores detectables internamente que causarán el inicio por el 68K de un proceso excepcional.

- * Errores de direccionamiento. Todas las instrucciones deben estar dispuestas en fronteras pares. Además, en todos los procesadores, excepto el 68020, los operandos palabra y palabra larga deben encajar en fronteras palabra. Un intento de acceder a operandos o instrucciones no alineadas da lugar a un proceso excepcional.

- * Violaciones de privilegios. Como recordaremos ciertas instrucciones están reservadas a rutinas en modo supervisor. Estas instrucciones facilitan el acceso a recursos que pueden romper la ejecución si se utilizan desde un programa de usuario. La ejecución por un usuario de estas instrucciones privilegiadas da lugar a un procesamiento excepcional.

instrucciones ilegales y no pertenecientes al repertorio: Como las instrucciones tienen una longitud de 16 bits, cierto número de combinaciones de 16 bits no corresponde a ninguna instrucción. Si un programa intenta ejecutar una instrucción no existente, se reinicia un procesamiento excepcional. Dos prefijos de instrucciones son definidos como no pertenecientes al repertorio en lugar de como ilegales; si los bits 15-12 son 1010 (llamados "línea A") o 1111 (llamados "línea F"), el procesador inicia un procesamiento excepcional diferente al que se efectúa con las instrucciones ilegales.

Las intercepciones de instrucciones son excepcionales originadas por la ejecución de instrucciones del programa. Hay una instrucción TRAP estandar que es usada frecuentemente por los sistemas operativos para dar a los programas en modo usuario acceso limitado a los recursos del modo supervisor. Hay otras instrucciones que inician procesamientos de excepción si se detectan ciertas condiciones, tales como desbordamientos o divisiones por cero.

El tercer tipo de excepción generada internamente ocurre cuando el 68K está operando en modo traza. Si el(los) bit(s) T del registro de estado está(n) a uno, potencialmente puede producirse una excepción después de cualquier instrucción. La función traza es útil para depuración de programas, ya que puede usted analizar la ejecución del programa parando después de cualquier instrucción o después de la ejecución clave.

EXCEPCIONES GENERADAS EXTERNAMENTE

Hay tres tipos diferentes de excepciones generadas externamente: errores en bus, reinicio y peticiones de interrupción.

Los errores en bus se identifican por la validación de circuitos externos de la señal BERR.

Reinicio, que resulta de la validación de circuitos externos de RESET.

Las peticiones de interrupción, que resultan de la validación por circuitos externos de una o más líneas de petición de interrupción (IPLO - IPL2).

6.4 PRIORIDADES DE EXCEPCION

El procesador aplica prioridades a los diferentes tipos de procesamiento excepcionales. Los tipos de excepciones de prioridad mayor son reinicio, error en bus y error en direccionamiento. Cualquiera de esas excepciones provoca la terminación inmediata de la instrucción en curso, incluso durante un ciclo de bus. El siguiente grupo de excepciones: traza, petición de interrupción, instrucciones ilegales/no existentes y violaciones privilegiadas, permiten la terminación de la instrucción en curso antes de iniciar el procesamiento excepcional. Las peticiones de interrupción incluyen unas prioridades adicionales, que dependen de la combinación de líneas de petición de interrupción validadas.

Las excepciones de menor prioridad corresponden a las que son causadas por instrucciones de tipo intercepción ("trap"). Estas instrucciones pueden iniciar procesamientos excepcionales como parte de su ejecución normal. Todas las excepciones de instrucciones de intercepción tienen igual prioridad ya que es imposible que dos de ellas generen excepciones simultáneamente.

La tabla 6-1 lista los tipos de excepciones de acuerdo con sus prioridades relativas y también define cuándo el procesamiento de cada tipo comienza.

TABLA 6-1

Prioridades de excepciones.

| Grupo/Prioridad | Características |
|---|--|
| 0.0 Reinicio | Cancelar todo procesamiento (no se salvan contenidos previos). |
| 1.0 Error de dirección | Suspende el procesamiento (se salvan contenidos previos). |
| 1.1 Error en bus | |
| 2.0 BKPT, CALLM, CHK, CHK2, inst. cpMid, violación cpProtocol, Dividir por cero, RTE, RTM, TRAP, TRAPV. | Procesamiento excepcional que es parte de la ejecución de una inst. |
| 3.0 Instrucción ilegal, línea-F, línea-A, violación de privilegio, preinstrucción cp. | El procesamiento excepcional comienza antes de que la inst. sea ejecutada. |
| 4.0 Post-instrucción. | El procesamiento excepcional comienza cuando la inst. en curso o el procesamiento excepcional previo concluyen |
| 4.1 Traza | |
| 4.2 Interrupción | |

6.5 TABLA DE VECTORES DE EXCEPCION

El núcleo de la secuencia de procesos excepcionales del 68K es una tabla de vectores que ocupa 1,024 bytes de memoria. En el 68000, esta tabla ocupa las direcciones de memoria de 0000 a 03FF. En el último procesador, la posición de esta tabla es definible por el usuario a través del registro base de vectores (VBR). Después de un reinicio hardware, el registro VBR se pone a cero, de forma que la tabla empieza en la dirección 0000.

La tabla consta de 256 vectores de cuatro bytes. Cada vector (excepto el primero) es una dirección de 32 bits que será cargada en el contador de programa como parte de la secuencia del proceso excepcional. Los primeros vectores son reservados para el contador del programa (PC) y el apuntador de la pila de reinicio. La tabla 6-2 lista los contenidos de la tabla de vectores. Observe que todos los vectores son captados del espacio de datos del supervisor, con la excepción de los vectores de reinicio (vectores #'s 0 y 1).

Como puede ver, el número de entrada en la tabla de vectores sirve para definir los tipos de excepciones que hemos discutido. Ciertos vectores sólo tienen sentido en los últimos miembros de la familia 68K. En los primeros modelos eran definidos como "reservados". Los primeros 64 vectores tienen usos predefinidos o reservados; esto deja 192 vectores disponibles para uso general, tal como peticiones de interrupción externas.

TABLA 6-2

Asignaciones de la tabla de excepciones.

| # DE VECTOR | DESPLAZAMIENTO | ASIGNACION |
|-------------|----------------|--|
| 0 | 000 | Reiniciar : Apuntador de pila de INT-INICIAL |
| 1 | 004 | Reiniciar : contador de programa inicial |
| 2 | 008 | Error en bus |
| 3 | 00C | Error de dirección |
| 4 | 010 | Instrucción ilegal |
| 5 | 014 | División por cero |
| 6 | 018 | Instrucción CHK,CHK2 |
| 7 | 01C | Instrucción cpTRAPcc, TRAPcc, TRAPV |
| 8 | 020 | Violación de privilegio |
| 9 | 024 | Traza |
| 10 | 028 | Emulador linea-A |
| 11 | 02C | Emulador linea-F |
| 12 | 030 | Reservado |
| 13 | 034 | Violación del protocolo de coprocesador |
| 14 | 038 | Error de formato |
| 15 | 03C | Interrupción inicializada |
| 16-23 | 040-05C | Reservados |
| 24 | 060 | Interrupción |
| 25 | 064 | Autovector (nivel 1) |
| 26 | 068 | Autovector (nivel 2) |
| 27 | 06C | Autovector (nivel 3) |
| 28 | 070 | Autovector (nivel 4) |
| 29 | 074 | Autovector (nivel 5) |
| 30 | 078 | Autovector (nivel 6) |
| 31 | 07C | Autovector (nivel 7) |
| 32-47 | 080-0BC | TRAP #0-15 (Intercepciones) |
| 48 | 0C0 | FPCP ramificación/poner a 1 cond. no ord. |
| 49 | 0C4 | FPCP resultado inexacto |
| 50 | 0C8 | FPCP dividir por cero |
| 51 | 0CC | FPCP infradesbordamiento |
| 52 | 0D0 | FPCP error de operando |
| 53 | 0D4 | FPCP desbordamiento |
| 54 | 0D8 | FPCP señalando NAN |
| 55 | 0DC | Reservado |
| 56 | 0E0 | PMMU configuración |
| 57 | 0E4 | PMMU operación ilegal |
| 58 | 0E8 | PMMU nivel de acceso |
| 59-63 | 0EC-0FC | Reservados |
| 64-255 | 100-3FC | Vectores definidos por el usuario |
| | | FPCP coprocesador de pto. flotante |
| | | PMMU Unidad Manejo de Pag. de memoria |

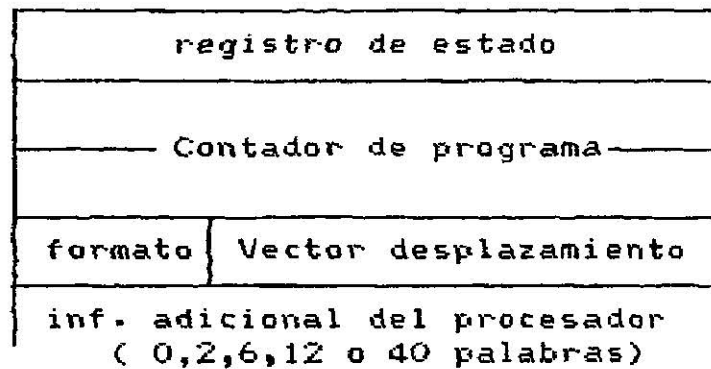


Figura 6-1
Formato general de estructura de pila.

6.6 ESTRUCTURAS DE PILAS.

Dependiendo del procesador y del tipo de excepción encontrado, el procesador puede insertar de 0 a 44 palabras de datos en la pila del supervisor como parte de la secuencia del procesamiento excepcional. Este bloque de datos es llamado "estructura de pila". La figura 6-1 muestra el formato general de la estructura de pila de excepción.

Observe en la figura 6-1 el rotulo "formato". El 68010 y 68020 tienen varias estructuras de pila que son identificadas por el campo formato. El 68000 y el 68008 son menos formales en lo referente a estructuras de pila ya que no incluyen un campo formato en los datos insertados.

La tabla 6-3 resume las definiciones de formatos de estructura de pila.

| FORMATO | T I P O |
|-----------|--|
| 0000 | Formato corto (4 palabras) |
| 0001 | Relleno (4 palabras) |
| 0010 | Excepción de instrucción (6 palabras) |
| 0011-0111 | Reservado |
| 1000 | Fallo en bus 68010/012 (29 palabras) |
| 1001 | Inst. de mediación con procesador (10 Pal) |
| 1010 | Fallo en bus corto 68020 (16 palabras) |
| 1011 | Fallo en bus largo 68020 (44 palabras) |
| 1100-1111 | Reservado |

Tabla 6-3.
Definiciones de formato de estructura de pila.

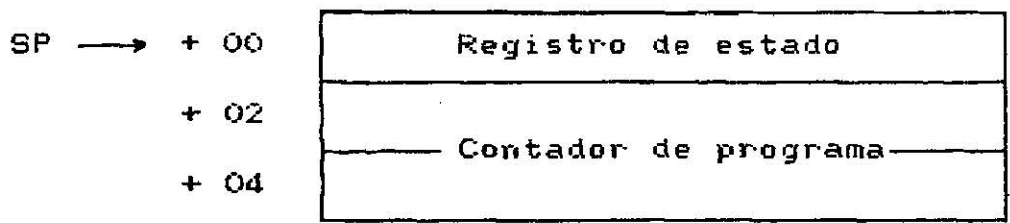


Figura 6-2 Estructura de pila corta 68000/68008.

La fig. 6-2 muestra una estructura de pila 68000/008 creada por una traza, TRAP, inst. ilegal/no existente, violación privilegiada o peticiones de interrupción. La figura 6-3 muestra la estructura de pila 68000/008 creada por un error excepcional en bus o dirección.

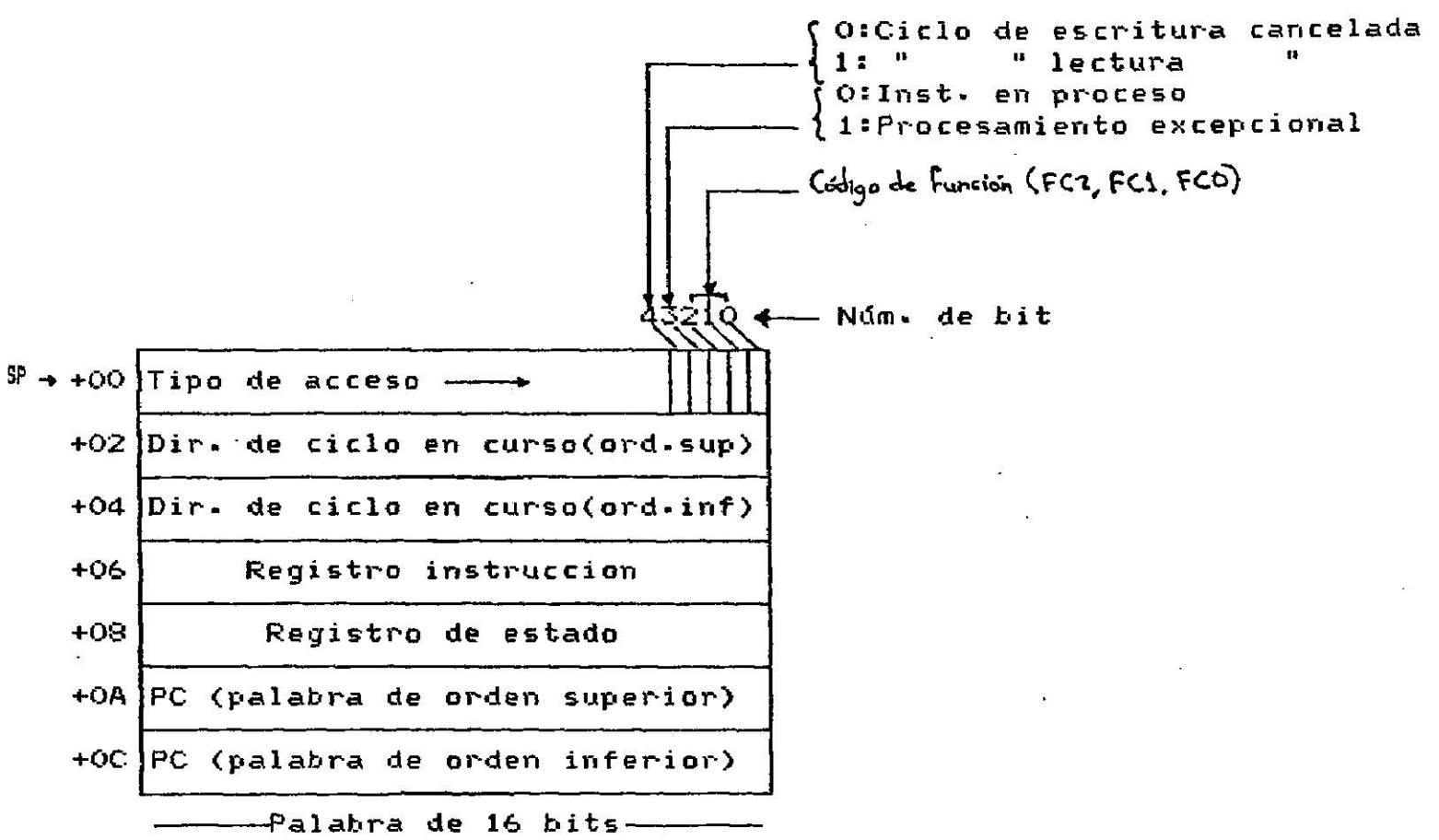


Figura 6-3. Estructura de pila en error en bus y de dirección en el 68000/68008.

La figura 6-4 muestra el formato \$0 de la estructura de pila. El 68010, 68012 y el 68020 utilizan este formato para interrupciones, errores de formato, instrucciones TRAP, intercepciones línea A y línea F, violaciones privilegiadas y excepciones de preinstrucciones para coprocesador.

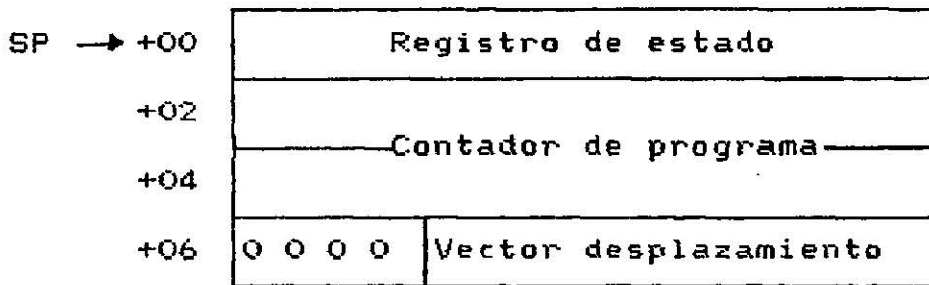


Figura 6-4 Formato \$0.

La figura 6-5 muestra el formato \$1. También llamado estructura de pila de "malgaste", el formato \$1 es utilizado por el 68020 cuando recibe una interrupción si el bit de control del registro de estado está a uno.

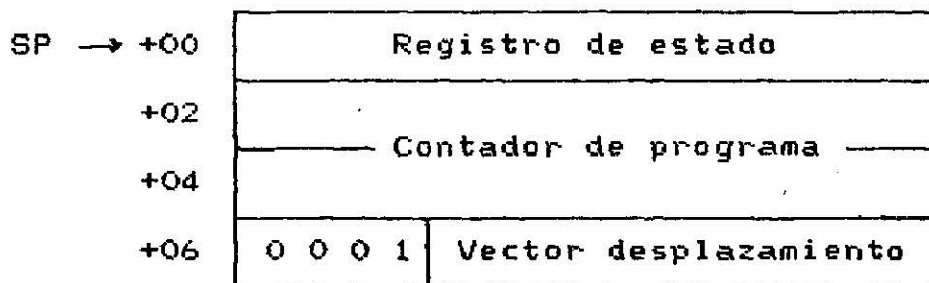


Figura 6-5 Formato \$1.

La figura 6-6 muestra el formato \$2. El 68020 utiliza esta estructura para excepciones relacionadas con postinsts. de coprocesador, CHK, CHK2, cpTRAPcc, TRAPcc, TRAPV, traza y división por cero.

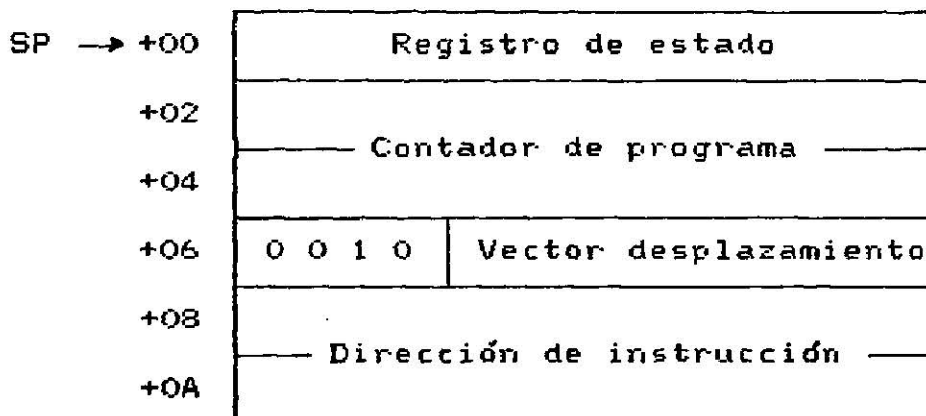


Figura 6-6 Formato \$2.

La figura 6-7 muestra el formato \$8. El 68010 y el 68012 utilizan esta estructura de pila para errores en bus y direcciones.

| | | |
|----------|-------------------------------|-----------------------|
| SP → +00 | Registro de estado | |
| +02 | Contador de programa | |
| +04 | | |
| +06 | 1 0 0 0 | Vector desplazamiento |
| +08 | Palabra estado | |
| +0A | Dirección fallada | |
| +0C | | |
| +0E | Reservado | |
| +10 | Registro de salida de datos | |
| +12 | Reservado | |
| +14 | Registro de entrada de datos | |
| +16 | Reservado | |
| +18 | Registro de entrada de insts. | |
| +1A | información interna(16 pals) | |
| . | . | . |
| . | . | . |
| . | . | . |
| . | . | . |
| +28 | . | . |

Figura 6-7 Formato \$8.

La figura 6-8 muestra el formato \$9. El 68020 utilizaba esta estructura para excepciones en medio de instrucciones coprocesador.

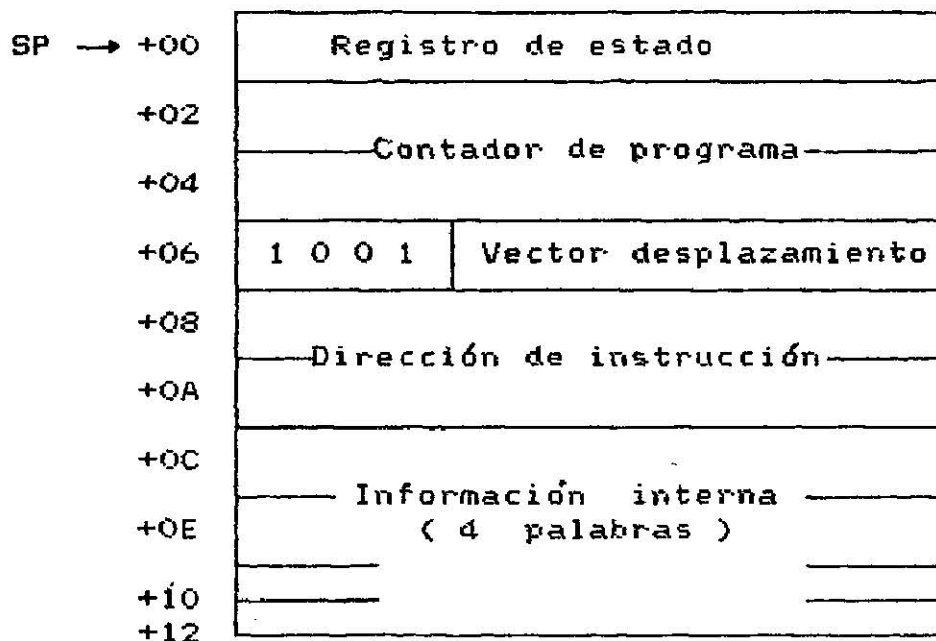


Figura 6-8

Formato \$9.

La figura 6-9 muestra el formato \$A. El 68020 utiliza esta estructura cuando encuentra un error de ciclo de bus mientras esta en la frontera de una instrucción.

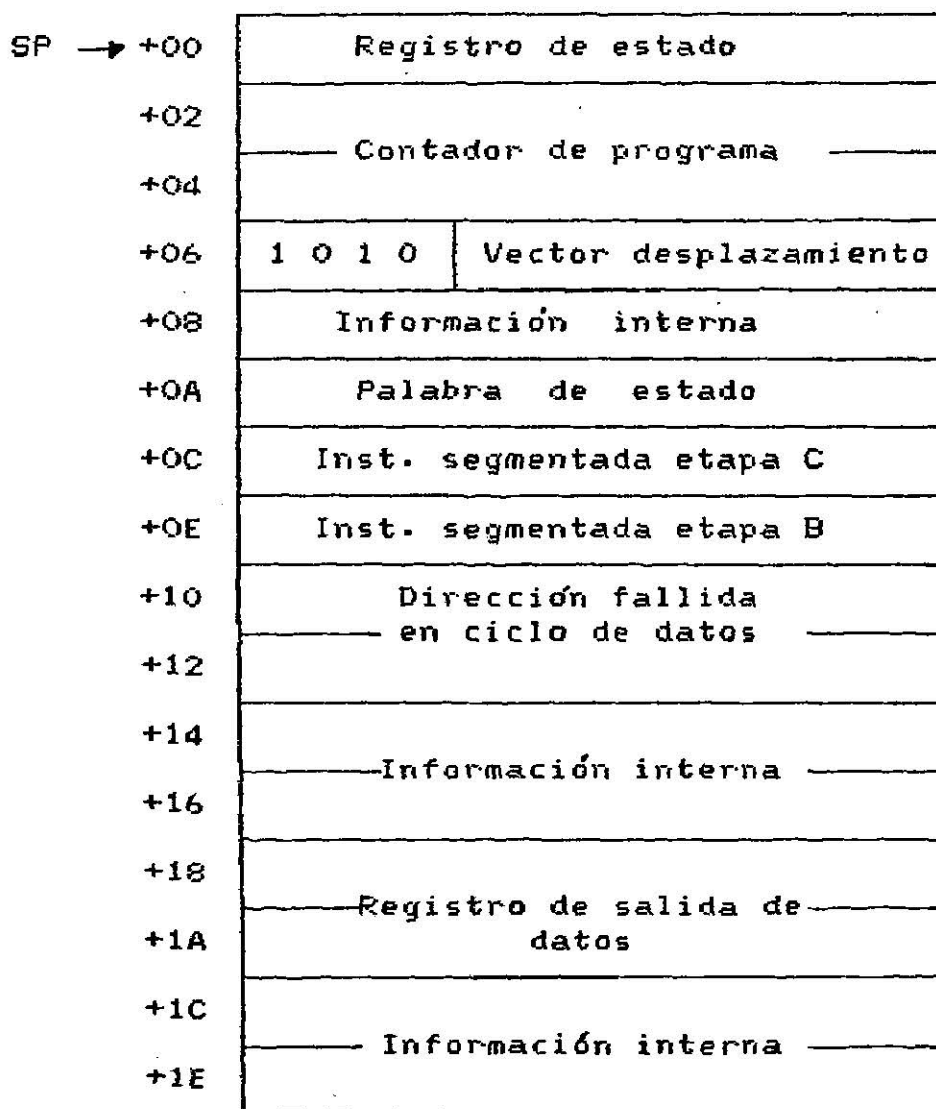


Figura 6-9

Formato \$A.

La figura 6-10 muestra el formato \$B. El 68020 utiliza esta estructura cuando encuentra un error en ciclo de bus en medio de la ejecución de una instrucción. Esta condición difiere de la anterior en que deben salvarse mayor cantidad de datos internos (pertenecientes a la ejecución de la inst.), con objeto de conservar la situación del procesador.

| | | |
|----------|-------------------------------------|-----------------------|
| SP → +00 | Registro de estado | |
| +02 | Contador de programa | |
| +04 | | |
| +06 | 1 0 1 1 | Vector desplazamiento |
| +08 | Información interna | |
| +0A | Palabra de estado | |
| +0C | Inst. segmentada etapa C | |
| +0E | Inst. segmentada etapa B | |
| +10 | Dirección fallida en ciclo de datos | |
| +12 | | |
| +14 | Información interna | |
| +16 | | |
| +18 | Registro de datos de salida | |
| +1A | | |
| +1C | Información interna | |
| +1E | | |
| +20 | | |
| +22 | | |
| +24 | Dirección etapa B | |
| +26 | | |
| +28 | Información interna | |
| +2A | | |
| +2C | Registro de entrada de datos | |
| +2E | | |
| +30 | Información interna (22 palabras) | |
| . | . | |
| . | . | |
| . | . | |
| +5A | 104 | |

Figura 6-10
Formato \$B.

6.7 SECUENCIA DEL PROCESAMIENTO DE EXCEPCIONES

Cada uno de los tipos de excepción sigue la misma secuencia general de eventos cuando comienza. Esta secuencia general es:

- 1.-Hacer una copia interna del registro de estado y poner a uno el bit del supervisor, poner a cero el(los) bit(s) de traza y (para interrupciones) modificar la máscara de interrupción.
- 2.-Determinar la entrada a la tabla de vectores, ya sea para leer de ella un dispositivo de interrupción o ya sea para utilizar los números de entrada fijos asociados con los otros tipos de procesamiento excepcional.
- 3.-Insertar el dato pertinente en la pila del supervisor. El tipo de datos insertados depende tanto del tipo de excepción como del miembro de la familia 68K.
- 4.-Cargar el contador de programa con la dirección procedente de la tabla de vectores y comenzar la ejecución.

6.8 INTERCEPCION ("TRAP") DE INSTRUCCIONES

Las intercepciones de instrucciones ocurren como resultado de la ejecución de instrucciones específicas. La instrucción normalmente puede causar procesamientos excepcionales, como con la instrucción TRAP, o puede causar procesamientos excepcionales sólo cuando ocurren situaciones anormales durante su ejecución, como sucede con las instrucciones TRAPcc, TRAPV, cpTRAPcc, CHK, CHK2, DIVS, DIVU, CALLM o RTM.

En el 68000 y 68008 el procesador construye una pila estructurada de tres palabras, como muestra la figura 6-2. En el 68010 y el 68012 el procesador construye una pila estructurada de cuatro palabras (formato \$0), como se muestra en la figura 6-4. El tipo de pila estructurada construido por el 68020 depende del tipo de instrucción que sea: una instrucción TRAP crea un formato de pila estructurada \$0 (mostrado en la figura 6-4) mientras que cualquier otra instrucción da lugar a un formato de pila estructurada \$2 (mostrado en la figura 6-6).

6.9 INSTRUCCIONES ILEGALES/NO EXISTENTES

Los procesos excepcionales por instrucciones ilegales o no existentes actúan similarmente a como lo hacen las instrucciones del punto anterior(6.8).

El 68000 y el 68008 utilizan la pila estructurada de tres palabras (fig. 6-2), mientras el resto de los miembros de la familia 68K utilizan el formato \$0 de la pila estructurada(fig. 6-4).

Observe la diferencia entre instrucciones ilegales y no existentes. Cualquier código de instrucciones que comience con la forma binaria 1010 ("línea A") o 1111 ("línea F") se considera instrucción no existente. Las instrucciones Línea A y Línea F tienen cada una de ellas su propia entrada en la tabla. Esto es conveniente para emular instrucciones no existentes en el repertorio.

El 68020 considera las instrucciones línea F como instrucciones potenciales para coprocesador. Después de decidir que no puede ejecutar la instrucción por sí mismo, el procesador realiza un ciclo de bus espacio CPU (ver fig. 5-6), codificando la identificación del coprocesador en el bus de dirección. Si el coprocesador no responde, la lógica externa indica un error en bus y el procesador inicia un procesamiento excepcional a través del vector de la línea F.

Si usted está emulando una instrucción no existente vía software, debería tener en cuenta que el contador de programa almacenado en la pila estructurada apunta a la instrucción ilegal. Después de acabar de emular las instrucciones, debe asegurarse de modificar el PC de forma que apunte a la siguiente instrucción a la ilegal.

6.10 ERROR DE DIRECCIONAMIENTO

Como lo indicamos anteriormente, las instrucciones deben enmascarse en fronteras de palabra. Si el procesador intenta acceder a una instrucción dispuesta a una dirección impar, se cancela e inicia un procesamiento de excepción. El 68000 y el 68008 construyen una pila estructurada de siete palabras, como muestra la fig. 6-3. El 68010 y el 68012 crean un formato \$8 de pila estructurada (fig. 6-7) y el 68020 utiliza un formato \$A de pila estructurada (fig. 6-9).

Observe que si durante un proceso excepcional por error en bus o por error de direcci3n o por reinicio, se encuentra un error de direcci3n, el procesador se introduce en un estado de parada y debe ser reiniciado externamente.

6.11 TRAZA

Los miembros de la familia 68K realizan las trazas paso a paso. Con una traza paso a paso, el procesador inicia un procesamiento excepcional despu3s de la ejecuci3n de cualquier instrucc3n. El bit de traza (del 68000 al 68012) o el bit T1 (en el 68020) del registro de estado, cuando est3 a uno, indica al procesador que efectu3 la traza despu3s de cada instrucc3n.

Para prevenir un procesamiento recursivo, una de las primeras cosas que es hecha por el procesador durante el procesamiento excepcional es poner a cero el (los) bit(s) de traza. El 68000 y el 68008 crean la pila estructurada de tres palabras (fig. 6-2), el 68010 y el 68012 crean el formato #0 (fig. 6-4) y el 68020 crea el formato #2 (fig. 6-6).

El 6802 permite una forma m3s selectiva de traza, una traza en cambio de flujo. Cuando el bit T0 del registro de estado es uno, el procesador act3a normalmente (sin traza) hasta que se ejecute una instrucc3n que interrumpa el procesamiento secuencial de instrucciones. Estas instrucciones son Bcc, JSR, BSR y sucesivas, y despu3s de haber sido ejecutadas provocan un procesamiento excepcional.

Observe que el intento de ejecuci3n de una instrucc3n ilegal o no existente no provocar3 una traza. Esto es importante si est3 emulando instrucciones. Su rutina de emulaci3n deber3a comprobar el bit de traza del registro de estado accesado en la pila antes de efectuar el retorno; si es uno, su rutina de emulaci3n tambi3n debe emular la traza.

6.12 PUNTOS DE RUPTURA

En algunos sistemas, puede ser conveniente insertar alg3n tipo de "punto de ruptura" en un programa. Para ser completamente 3til en un sistema emulador hardware, el procesador debe tener alguna forma de indicar al hardware externo que ha alcanzado tal punto de ruptura. En el 68000 y el

68008, un programa puede insertar una instrucción ilegal como punto de ruptura: el hardware externo puede monitorizar las líneas de dirección esperando un acceso a un vector de instrucción ilegal en la tabla de vectores.

En el 68010, 68012 y 68020, esto no funcionará, ya que disponen del registro base de vectores (VBR), que permite a un programa cambiar dinámicamente la posición física de la tabla de vectores. Estos definen un conjunto especial de instrucciones ilegales como instrucciones "punto de ruptura".

Cuando la ejecución se efectúa en el 68010 o el 68012, el procesador primero lanza un "ciclo de bus punto de ruptura". Como se muestra en la Tabla 5-2, éste está clasificado como un ciclo de bus espacio CPU. Después de emitir el ciclo de bus punto de ruptura, el procesador ejecuta una intercepción a través del vector de instrucción ilegal y comienza normalmente la ejecución de la instrucción ilegal.

En el 68020, el procesador no solo emite un ciclo de bus punto de ruptura, sino que también realiza una lectura del espacio CPU durante este ciclo. Si esta lectura termina con un error en bus, el procesador intercepta a través de un vector de instrucción ilegal. Sin embargo, el hardware externo puede situar una nueva instrucción en el bus de datos y acaba el ciclo con una señal DSACKx. En este caso, el procesador reemplaza la instrucción punto de ruptura (en la unidad de segmentación) con la instrucción en el bus de datos y comienza la ejecución con aquella instrucción.

6.13 ERROR DE FORMATO

El 68020 tiene tres instrucciones que efectúan una comprobación del formato de datos, la llamada a módulo (CALLM), retorno de módulo (RTM) y restablecer coprocesador (cpRESTORE). Estas instrucciones sirven para encontrar datos específicos en pilas estructuradas, descriptores de módulo y campos de formato. Si encuentran que el dato no es correcto, lo interceptarán a través del vector de excepción de error de formato utilizando el formato \$0 de la pila estructurada (fig. 6-4).

6.14 INTERRUPCIONES

Los dispositivos externos pueden interrumpir el flujo normal del procesador validando una o más líneas de petición de interrupción (IPLO-IPL2). El procesador reconoce la petición dada a él, entre instrucciones normales y, en el caso del 68020, durante el transcurso de ciertas instrucciones de coprocesador (cp). Después de recibir la petición de interrupción, el procesador compara el valor binario indicado por IPLO-IPL2 con la máscara de interrupción en el registro de estado. Si IPLO-IPL2 muestra una prioridad mayor que la situada en la máscara, el procesador inicia un procesamiento excepcional. En caso contrario, el procesador ignora la petición y continúa con su procesamiento normal.

Si el procesador desea reconocer la interrupción, ejecuta un ciclo de bus espacio CPU con la salida de nivel de prioridad de interrupción en las líneas A1-A3 del bus de dirección (vease tabla 5-2).

El dispositivo puede elegir enviar el número de entrada de la tabla de vectores validando DTACK y cargar el número de vector en el bus de datos.

El dispositivo puede, alternamente, desear que el procesador utilice la entrada de autovector. En los modelos 68000-68012, debe validarse VPA; y en el 68020 se debe validar AVEC. El procesador entonces interceptará con la entrada de autovector asociada con el nivel de interrupción indicados.

Si ningún dispositivo responde al ciclo de reconocimiento de interrupción, el hardware externo debe validar BERR.

La mayoría de los nuevos dispositivos periféricos 68K tienen número de vectores de interrupción programables que envían al procesador un reconocimiento de interrupción. Implícitamente estos dispositivos envían el vector \$0F si no han sido inicializados a través de software. Este vector es definido como la entrada al vector no inicializado, y en la documentación inicial se le clasifica como "reservado".

La pila estructurada creada por excepciones de interrupción varía entre procesadores. En el 68000 y 68008, el procesador construye la pila estructurada de tres palabras (fig. 6-2). En el 68010 y 68012, el procesador construye la pila estructurada del

formato \$0 (fig. 6-4). En el 68020, la pila estructurada depende de la acción que estuviese realizando el procesador cuando se le interrumpió: si estaba entre instrucciones, el procesador elabora un formato \$0 de la pila estructurada de la pila supervisor; si fuese interrumpido durante una instrucción de coprocesador, el procesador construye un formato \$9 de pila estructurada (fig. 6-8).

La posición de la pila estructurada en el 68020 depende del estado del bit M en el registro de estado. Si es cero, el procesador construye la pila estructurada en la pila de interrupción. Si es uno, el procesador construye la pila estructurada en la pila principal y después construye una "pila estructurada de malgaste" (formato \$1) en la pila de interrupción.

6.15 ERROR EN BUS

En respuesta a una petición de leer o escribir, la lógica externa puede presentar o aceptar los datos y validar DTACK o también puede responder con una señal de error en bus (BERR). La lógica externa puede validar BERR si la memoria no está físicamente presente o si la memoria requerida no corresponde al espacio de direcciones del usuario. Los sistemas frecuentemente hacen uso de unidades de manejo de memoria para determinar la disponibilidad de una tarjeta de memoria.

En el 68000 y 68008, al recibir la señal de error en bus, el procesador construye una pila estructurada de siete palabras (fig. 6-3).

El 68010, 68012 y 68020 permiten manejar "memoria virtual". En un sistema con memoria virtual el espacio total de memoria de las tareas en el sistema puede ser mayor que el espacio de memoria física disponible. El sistema operativo puede elegir almacenar ciertos datos y trozos del programa en memoria secundaria mientras que otras secciones del programa o datos no disponibles, el sistema operativo debe ser capaz de copiar rápidamente los datos no esenciales en disco (memoria secundaria) y leer del disco el material solicitado.

Para disponer de una memoria virtual eficiente el procesador debe ser capaz de reconocer la ausencia de la memoria solicitada, cargar los datos desde disco y continuar la ejecución del programa. Los circuitos lógicos externos pueden reconocer la no existencia de la memoria requerida y validar BERR. El 68000 y 68008, no salvaguardan suficiente

información en la pila para capacitar al procesador a reiniciar la instrucción una vez que los datos estén disponibles en memoria.

El 68010, 68012 y 68020 crean pilas estructuradas más elaboradas cuando se inicia un procesamiento excepcional de error en bus. El 68010 y 68012 utilizan el formato \$B de la fig. 6-7 y el 68020 usa ya sea el formato \$A de la fig. 6-9 (para errores en bus durante la captación de instrucción) o bien el formato \$B de la fig. 6-10 (para errores en bus durante la ejecución de instrucción).

Los datos adicionales en la pila contienen información interna al procesador. Como parte de la rutina de excepción, el procesador puede evaluar la información de la pila y determinar si el error fue causado por una solicitud de información almacenada en disco. Si así es, puede transpasar datos hacia afuera de la memoria y transpasar la información requerida de disco a memoria. Después de actualizar los registros de manejo de memoria adecuados, el sistema operativo puede dejar al programa fallido probar de nuevo el acceso.

Para realizar esto, el procesador emite una instrucción de retorno de excepción (RTE). La instrucción RTE examina el formato de la pila estructurada y determina si los datos en la pila es el resultado de un error en bus y si debería restablecer los datos internos almacenados en la pila y reanudar la ejecución del programa en donde indica el contador de programa (PC). Observe que este apuntador de continuación puede estar en medio o al comienzo de una instrucción.

En todos los procesadores 68K, si, mientras transcurre la captación del vector u operación pila, el procesador encuentra un segundo fallo de bus, el procesador entra en estado de parada. Esto se denomina "doble fallo de bus" y la única forma de salir de esta situación es un reinicio ("reset") externo.

6.16 REINICIO ("RESET")

La excepción reinicio es la única frente a los diferentes tipos de excepciones de que no hay que tomarse la molestia de guardar información alguna en una pila.

Cuando se valida un reinicio externo ocurre lo siguiente:

- 1.- El proceso en curso se cancela.

- 2.- En el registro de estado, el bit de supervisor se hace uno, el bit de control se hace cero (sólo en el 68020), el (los) bit(s) de traza se ponen a cero, la máscara de interrupción se fija al nivel 7, el registro base de vectores se pone a \$0000 (68010 - 68020) y el registro de control de acelerador se hace cero (68020).
- 3.- El procesador carga el apuntador de pila del supervisor (ISP en el 68020) con la palabra larga con desplazamiento \$0000 en la tabla de vectores y carga el contador de programa con la palabra larga con desplazamiento \$0004. Observe que este vector se introduce en el espacio del programa supervisor mientras que todos los demás vectores de excepción se dan a través del espacio de datos del supervisor.
- 4.- El procesador valida RESET para reiniciar los dispositivos externos y comenzar la ejecución en un nuevo valor PC (Contador del Programa).

CONTENIDO

PARTE VII

MISCELANEA

| | | |
|-------|--|-----|
| 7.1 | Introducción. | 114 |
| 7.2 | Interfaz con la familia 6800. | 115 |
| 7.3 | Interfaz con un coprocesador. | 116 |
| 7.4 | Sus diferencias entre sus modelos. | 118 |
| 7.5 | El acelerador "CACHE" del 68020. | 121 |
| 7.5.1 | Introducción. | 121 |
| 7.5.2 | Como opera el acelerador(cache) del 68020. | 121 |
| 7.5.3 | Control del acelerador. | 122 |
| 7.5.4 | Inhabilitación del acelerador. | 123 |

7.1 INTRODUCCION

En esta Parte VII y ultima lleva por titulo "miscelanea" ya que tiene cuatro puntos a tratar de distinto concepto cada uno de ellos, los cuales tratan sobre la familia 68000.

- 7.2.- su interfaz con la familia 6800.
- 7.3.- su interfaz con un coprocesador(CP).
- 7.4.- sus diferencias entre sus modelos.
- 7.5.- El acelerador "CACHE" del 68020.

En el punto 7.2 se da una explicación del porque se incluyeron estos modelos de la familia 68000 interfaz con la anterior familia.

En el punto 7.3 se habla de lo que es un coprocesador, como funciona el 68020 con él, y se compara con los dispositivos que no son coprocesadores.

En el punto 7.4 se hace un resumen de las diferencias que se vieron en los anteriores partes entre los modelos de la familia 68000.

Y por último en el punto 7.5 se habla del acelerador ("cache") de memoria del 68020, en particular, ya que es el unico modelo de la familia 68000 que maneja el acelerador de memoria.

7.2 INTERFAZ CON LA FAMILIA 6800

Cuando Motorola anunció el procesador 68000, había pocos circuitos integrados periféricos disponibles para buses de datos de 16 bits y con control de bus asíncrono. Había, sin embargo, bastantes periféricos disponibles para la anterior familia de procesadores, el 6800. Con el objeto de facilitar a los diseñadores desarrollar sistemas basados en el 68000 sin esperar a periféricos completamente compatibles.

Motorola añadió tres señales al 68000 y a los 3 modelos siguientes (68008,68010,68012) de manera que pudiesen interconectarse a los dispositivos de la familia 6800.

Como recordará, los procesadores 68K utilizan señales de conformidad asíncronas para orquestar las transferencias de datos hacia y desde los periféricos. La familia 6800, por su parte, requiere transferencias de datos síncronas. Las señales de la familia 6800 incluidas en el 68000 permiten a este procesador emular las señales de conformidad síncronas.

Las señales requeridas son la señal de dirección de memoria válida (VMA), la señal de dirección de periférico válida (VPA) y la señal de habilitación (E).

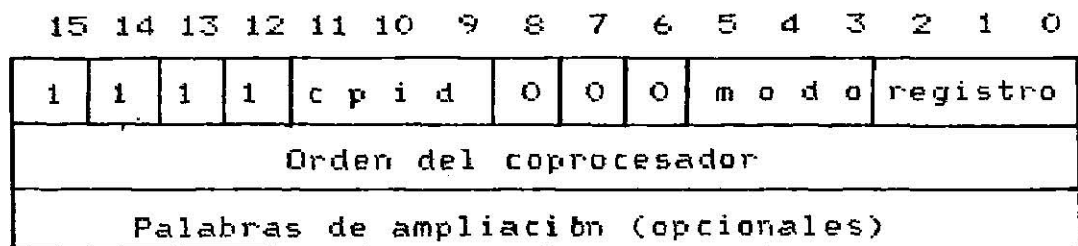
Hoy en día, la mayoría de los periféricos necesarios para la familia 68000 están disponibles en formato de transferencia asíncrono. De aquí que los nuevos microprocesadores de esta familia (68K) no tengan necesidad de utilizar las señales de interfaz de la familia 6800. Un ejemplo de ello es el modelo 68020 que no incluye estas señales.

7.3 INTERFAZ CON COPROCESADOR

Un coprocesador (cp) es un procesador de uso específico que actúa como una ampliación del procesador principal.

Los coprocesadores difieren de los periféricos corrientes en su interfaz con el procesador. Mientras el procesador se comunica con el periférico a través de instrucciones de programación normales, las comunicaciones con el coprocesador ocurren transparentemente al programador del lenguaje ensamblador, es decir, él no se da cuenta de esta comunicación entre el procesador (MP) y el coprocesador (cp). El coprocesador aparece así como parte integral del procesador principal.

Las instrucciones del cp aparecen como instrucciones línea F según se ilustra en la siguiente figura.



Cuando el procesador principal encuentra una instrucción línea F, emite un ciclo espacio CPU. Como parte de este ciclo, el procesador codifica el número de identificación del procesador requerido (extrayéndolo a partir del código de instrucción). Si el procesador no está presente, los circuitos externos validan BERR y el procesador intercepta a través del vector línea F.

Si el cp está presente, el procesador principal comienza una serie de transferencias bidireccionales, es decir, de entrada y salida. El procesador principal indica al cp qué ordenes requiere, el cp solicita al procesador principal que capte cualquier operando necesitado, entonces él ejecuta la orden. Después de ejecutarla, devuelve cualquier dato requerido al procesador principal. Mientras está ejecutando la orden, el procesador principal continuará sondeando al coprocesador hasta que detecte lo que se ha realizado. En ese momento, el procesador puede continuar la ejecución de la instrucción.

El procesador principal utiliza una serie de instrucciones estandar para comunicarse con un periférico. Las instrucciones ejecutadas por un interfaz coprocesador están enteramente en microcódigo; lo cual no requiere captaciones externas de instrucciones por lo que pueden ejecutarse más rápidamente. Como están en microcódigo, el programador no necesita conocer nada concerniente con los detalles de la interfaz, sino sólo lo relativo a la sintaxis de las instrucciones del coprocesador.

El 68020 es el único miembro de la familia 68K que tiene el microcódigo disponible para la interfaz con coprocesador.

7.4 Resumen de las diferencias entre la familia 68K.

| | 68000 | 68008 | 68010 | 68012 | 68020 |
|----------------------------------|---|---------|---------------------------|--------------|-----------------------------------|
| Bus de datos | 16 | 8 | 16 | 16 | 8,16,32 |
| Bus de dir. | 24 | 20 | 24 | 30 | 32 |
| Acelerador "Cache"/anidamientos. | ninguno | ninguno | 3 palabras de anidamiento | | 128-pal. de memo. "cache" |
| Memo/Maq virtual | NO | NO | SI | SI | SI |
| Alineamiento de memoria | Alineamiento par para palabras, palabras largas, pilas. | | | | Alineamiento par solo para insts. |
| Registro de control | ninguno | ninguno | SFC,DFC, VBR | SFC,DFC, VBR | SFC,DFC, VBR,CACR, CAAR. |
| Apuntadores de pila | USP,SSP | USP,SSP | USP,SSP | USP,SSP | USP,SSP (MSP,ISP) |

Ademas:

Modos de direccionamiento adicionales del 68020:

- Indirecto memoria.
- Indice escalado.
- Desplazamientos largos.

Instrucciones adicionales del 68020:

| | |
|-----------|---|
| Bcc | admite desplazamientos de 32 bits. |
| BFxx | nuevo tipo de instrucción. |
| BKPT | admite operaciones externas de sustitución. |
| BRA | admite desplazamientos de 32 bits. |
| BSR | admite desplazamientos de 32 bits. |
| CALLM | nueva instrucción. |
| CAS,CAS2 | nueva instrucción. |
| CHK | admite operandos de 32 bits. |
| CHK2 | nueva instrucción. |
| CMPI | nuevos modos de direccionamiento. |
| CMP2 | nueva instrucción. |
| cpxx | nuevo tipo de instrucción. |
| DIVS,DIVU | admite operandos de 32 y 64 bits. |
| EXTB | admite 8 bits extendidos a 32 bits. |

| | |
|-----------|-------------------------------------|
| LINK | admite desplazamientos de 32 bits. |
| MOVEC | admite nuevos registros de control. |
| MULS/MULU | admite operandos de 32 bits. |
| PACK | nueva instrucción. |
| RTM | nueva instrucción. |
| TST | nuevos modos de direccionamiento. |
| TRAPcc | nueva instrucción. |
| UNPK | nueva instrucción. |

Comparación de patas-de-salida(pin-out).

| | 68000 | 68008 | 68010 | 68012 | 68020 |
|------------------------|-------------------|-------|-------|-------|-------|
| líneas de dirección | 24 | 20/22 | 24 | 31 | 32 |
| Rango de dir. virtual | 16 Mb | 1a4Mb | 16 Mb | 2 Gb | 4 Gb |
| líneas de datos | 16 | 8 | 16 | 16 | 32 |
| Bus de protocolo | a s i n c r o n o | | | | |
| ciclos de reloj/bus | 4 | 4 | 4 | 4 | 3 |
| # de datos validados | 2 | 1 | 2 | 2 | 1 |
| # transfer. reconocida | 1(DTACK) | 1 | 1 | 1 | 2 |
| Lin. código de función | 3 | 3 | 3 | 3 | 3 |
| Lin. arbitraria de bus | 3 | 2/3 | 3 | 3 | 3 |

Una comparación de los recursos del supervisor.

| | 68000/008 | 68010/012 | 68020 |
|-----------------------------------|-----------|-----------|-------|
| Registro de estado | | | |
| Hardware trace bits | 1 | 1 | 2 |
| bit estado del supervisor | SI | SI | SI |
| bits mask Interrupción | 3 | 3 | 3 |
| Apuntadores de pila | | | |
| pila de interrupción | SI | SI | SI |
| pila maestra o principal | NO | NO | SI |
| Registro de Vector Base (VBR) | NO | SI | SI |
| Registro de control de acelerador | NO | NO | SI |
| Códigos de función Alternativos | NO | SI | SI |

Comparación del conjunto de instrucciones.

| | 68000/008 | 68010/012 | 68020 |
|----------------------------------|-----------|-----------|-------|
| Movimientos de datos | 7 | 9 | 9 |
| Aritméticos/lógicos | 17 | 17 | 18 |
| Código decimal binario | 5 | 5 | 7 |
| Operando simple | 9 | 9 | 9 |
| Cambiar y rotar | 8 | 8 | 8 |
| manipulaciones de bits | 4 | 4 | 4 |
| manipulaciones de campos de bits | 0 | 0 | 8 |
| Salto (16 condiciones) | 3 | 3 | 3 |
| Excepción-relatada | 5 | 7 | 8 |
| Control | 12 | 13 | 20 |
| Insts. generadas a coprocesador | 0 | 0 | 7 |
| Total | 70 | 75 | 101 |

Comparación de la transferencia de los signos del bus.

| Nombre de la función de signos | 68000 | 68008 | 68010/012 | 68020 |
|--|---------|-------|-----------|-----------|
| Validación de dirección | AS | AS | AS | AS |
| Validación de datos | UDS/LDS | DS | UDS/LDS | DS |
| Reconocimiento de transmisión de datos | DTACK | DTACK | DTACK | DSACK 0/1 |
| tamaño de operando | n/a | n/a | n/a | SI20/1 |

7.5 EL ACELERADOR ("CACHE") DEL 68020.

7.5.1 INTRODUCCION.

Una gran cantidad de tiempo de ejecución de procesador se invierte en la ejecución de buses. Una forma usual de mejorar las prestaciones del procesador es usar un acelerador ("cache") para hacer que las instrucciones de dichos lazos sean más fácilmente disponibles al procesador. Una vez leídas las instrucciones por la memoria aceleradora, el procesador puede ejecutarlas mucho más rápidamente que si tuviese que captar las instrucciones de la memoria externa.

La velocidad de ejecución del procesador se incrementa considerablemente omitiendo los estados de espera durante los ciclos de lectura. El 68020 va más allá de esto porque incorpora un acelerador de instrucción "en el interior del circuito integrado". El acceso a este acelerador puede realizarse sin ningún tipo de captación en bus.

7.5.2 COMO OPERA EL ACELERADOR("CACHE") DEL 68020.

El acelerador interior del circuito integrado puede mantener hasta 64 palabras largas procedentes de la memoria. Cada palabra consiste en un campo "etiqueta", un bit de "validez" y los 32 bits de instrucción. El campo "etiqueta" contiene los 24 bits superiores (A8-A1) de la dirección y el valor FC2 (que se diferencia entre el espacio de usuario y espacio del supervisor).

Siempre que el 68020 necesite captar una instrucción primero comprueba el acelerador para ver si la instrucción está presente en él. Hace esto indexando en el acelerador con los bits A2-A7 de la dirección de instrucción. Después comprueba el campo etiqueta y el código FC2 asociado a esta palabra. Si ellos casan, y el bit de validez es 1, se produce un "acierto" en el acelerador. El bit A1 selecciona el desplazamiento entre la palabra larga de entrada y el procesador puede comenzar a ejecutar las instrucciones sin realizar ninguna captación externa.

Si el campo etiqueta no casa, o si el bit de validez no es 1, se produce un "fallo" en el acelerador. Y el procesador capta la instrucción de memoria utilizando un ciclo de lectura normal. Según lee la instrucción, la inserta en el acelerador, actualiza el campo etiqueta y pone a uno el bit de validez.

Observe que el 68020 utiliza el acelerador solo para instrucciones, la captación de datos siempre requiere ciclos de memoria externos.

7.5.3 CONTROL DEL ACELERADOR

No siempre se necesita el acelerador de instrucciones o incluso resulta inútil en ciertos casos. Por ejemplo, durante una emulación hardware, el sistema puede requerir captaciones externas para facilitar algún aspecto de la emulación. El 68020 puede controlar el acceso al acelerador a través del registro de control del acelerador (CACR o "Cache Control Register") y el registro de dirección del acelerador ("CAAR o "Cache Address Register"). El CAAR es utilizado sólo para poner a cero las palabras del acelerador.

El CACR es un registro de 32 bits con cuatro bits definidos, como muestra la figura siguiente:



Registro de control del acelerador (CACR).

El procesador accesa al CACR con la instrucción de transferencia de control (MOVEC). Observe que esta instrucción es privilegiada. Los bits de CACR se definen como sigue:

E Habilidad del acelerador

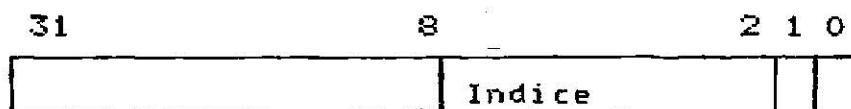
Este bit, cuando es uno, habilita las funciones normales del acelerador. Cuando es cero, inhabilita el acelerador y fuerza a que todas las instrucciones se capturen por medio de ciclos externos. El hardware siempre pone a cero este bit.

F Retención del acelerador

Este bit, cuando es uno, mantiene los contenidos del acelerador. Puede leerse (producirse "aciertos"), pero los "fallos" del acelerador no producen remplazamientos de las palabras del acelerador. El hardware siempre pone a cero este bit.

CE Poner a cero las palabras.

Este bit, cuando es uno, pone a cero el bit de validez de la palabra del acelerador asociada con el índice dado por el CAAR. Esta función ocurre sólo una vez, cuando se escribe en el CACR o con la instrucción MOVEC (el bit sólo es escrito). El formato del CAAR se muestra en la siguiente figura:



Registro de dirección del acelerador (CAAR).

C Limpiar el acelerador.

Este bit de sólo escribir, cuando es uno es una instrucción MOVEC a CACR, invalida todo acceso al acelerador (por ejemplo, como parte de un conmutador en el contexto de una tarea).

7.5.4 Inhabilitación del acelerador.

La entrada de inhabilitación del acelerador (CDIS) dinámicamente inhabilita al acelerador. Después de que la lógica externa valida CDIS, toda captación sucesiva de instrucciones se hace de la memoria externa, independientemente del estado del bit de habilitación del acelerador del CACR. Cuando la lógica externa niega CDIS, el acelerador es rehabilitado para la captación de las siguientes instrucciones.

CONCLUSIONES

C O N C L U S I O N E S

Con el microprocesador 68000 muchos conceptos de su arquitectura provienen de las minicomputadoras. Sus características son un controlador multinivel - microprogramable, operaciones pipelined-paralelo, con prefetching, y buses asincronos no-multiplexados. Su arquitectura y conjunto de instrucciones que son altamente regulares que soportan lenguajes de alto nivel y restaurador rapido de programas. Su completa habilidad para condiciones de procesamientos excepcionales que permiten ayudar al programador para tratar con hardware inadecuado y problemas de software en una manera limpia y consistente. Esta habilidad y un estado supervisor proveen un alto grado de proteccion de error.

De todas las consideraciones dadas para el diseño del 68000, se tomó como alta prioridad la "performance". El objetivo fue en desarrollar un microprocesador con mejor "performance" que su antecesor el 6800 de 8-bits por un orden de magnitud. Este objetivo fue alcanzado con mucho éxito. Cuando una elección se tiene que hacer entre la "performance" y algun otro factor, para la mayor parte la "performance" queda fuera. Por ejemplo, multiplexando los buses de direcciones y de datos daría como resultado en un pequeño empaquete pero que sería también reducida la "performance" por mucho en un 30%, por eso los buses no fueron multiplexados en la familia 68000.

La segunda consideración más importante dada en el 68000 fue la consistencia, con respecto al hardware y software, igualmente. La consistencia no es solamente hacer a la maquina más flexible y poderosa, es también hacerla facil de usar. Todos los registros de datos funcionan idénticamente, por ejemplo, como hacer todos los registros direcciones, y todos los registros de direcciones y de datos deben servir como registros índices.

El conjunto de instrucciones del 68000 consiste de un número relativamente pequeño de tipos de instrucciones de proposito general. Más instrucciones operan sobre bytes, palabras de 16-bits, y palabras largas de 32-bits, el tamaño del dato especificado en el mnemónico. Consecuentemente la consistencia en las operaciones sobre diferentes tamaños de datos es mantenida. La consistencia en el conjunto de condiciones de códigos también se encuentra en el 68000. Varias instrucciones diferentes afectan los mismos códigos de condición. Por ejemplo, ejecutando una instrucción TST (la cual resta cero desde un operando) afecta los códigos de condición exactamente en la misma manera como una instrucción SUB#0.

El énfasis fue también sobre el soporte de lenguajes de alto nivel. Especial atención fue enfocada sobre operaciones encontradas frecuentemente que normalmente requieren varias líneas de código. Varias de tales instrucciones fueron implementadas en el 68000 como simples instrucciones, de este modo se incrementó la eficiencia de compilación y se decrementó los requerimientos de memoria al mismo tiempo.

B I B L I O G R A F I A

- 1.- Titulo : Microprocessors.
Autores : Laurence Altman.
Editorial : Electronics Book Series McGraw Hill 1975
- 2.- Titulo : Introducción a la ciencia de las computadoras.
Autores : Jean Paul Tremblay & Richard B. Bunt.
Editorial : McGraw Hill 1981.
- 3.- Titulo : Manual del Microprocesador 68000.
2da. Edición.
Autores : William Cramer y Gerry Kane.
Editorial : Osborne McGraw Hill 1986.
- 4.- Titulo : Introduction to microprocessor system design.
Autores : Harry Garland.
Editorial : New York McGraw Hill 1979.
- 5.- Titulo : Microprocessor and Microcomputers
One-chip Controllers to High-end Systems.
Autores : Raymond P. Capace; Managing Editor(technical)
and John G. Posa, Solid State Editor(Electronics).
Editorial : Electronics Magazine Book McGraw Hill 1981.
- 6.- Titulo : Organizaci6n de computadoras.
Segunda Edici6n.
Autores : V. Carl Hamacher
Zvonko G. Vranesic
Safwat G. Zaky
Editorial : McGraw Hill 1986.
- 7.- Titulo : Folletos de Motorola Corp.
y algunas revistas de byte y Magazine PC.
Fechas : de los meses de Febrero de 1986 a Noviembre de 1987.

