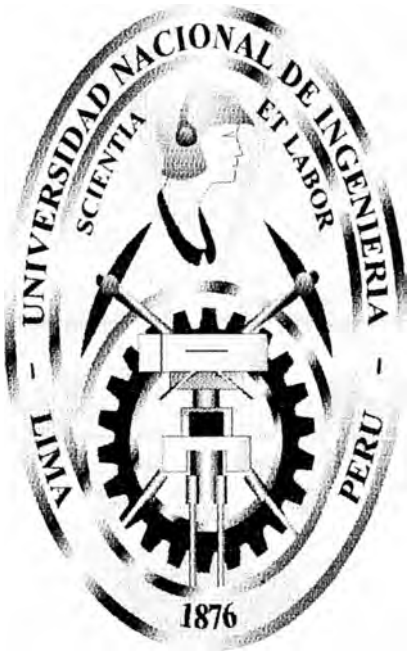
UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



CONTROL DE UN PROCESO DE ELEVACIÓN Y SELECCIÓN DE PAQUETES, EMPLEANDO EL MICROCONTROLADOR 80C535

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO

OSWALDO ALBERTO WATERS TORRES

PROMOCIÓN 1978-II

LIMA-PERÚ

2 001

A mi esposa Elisa y a
mis hijos Oswaldo y Edwin
por su apoyo, estímulo,
comprensión y cariño

CONTROL DE UN PROCESO DE ELEVACIÓN Y SELECCIÓN DE PAQUETES, EMPLEANDO EL MICROCONTROLADOR 80C535

SUMARIO

En la presente tesis se realiza un proyecto de aplicación tecnológica del microcontrolador 80C535 al campo de control de procesos de manufactura, específicamente, a un proceso de elevación y selección de paquetes. La aplicación de este microcontrolador es en forma alternativa a la lógica cableada basada en relés o a la lógica programada basada en PLCs o PCs con la finalidad de tener costos mínimos de equipamiento, consumo mínimo de energía y pérdidas mínimas por disipación.

La ejecución del proyecto se ha realizado en el laboratorio de Control y Automatización de la FIEE empleando como planta a controlar el proceso neumático de elevación y selección de paquetes y la tarjeta 80535 compuboard que contiene al microcontrolador 80C535, para utilizarlo como controlador. Así mismo se ha elaborado un programa con la lógica de control correspondiente, el mismo que ha funcionado satisfactoriamente.

Para la adquisición de datos de planta y para el accionamiento de dispositivos finales de control se ha diseñado y construido tarjetas que procesan las señales de entrada y de salida del microcontrolador, con un costo reducido.

INDICE

PROLOGO

La aparición del microprocesador propició un cambio decisivo en las técnicas de diseño de los equipos de instrumentación y control entre otros. Los microprocesadores requerían de un conjunto de circuitos integrados (C.I.) que resolverían las necesidades de memoria, de entradas/salidas, temporizadores, conversores, etc. con los que se podían construir sistemas.

Luego los microprocesadores se integrarán junto con los subsistemas que anteriormente formaban unidades especializadas e independientes. A este nuevo C.I. se le denominó microcomputador monopastilla. A las monopastillas que son especializados en aplicaciones industriales se le denomina microcontroladores (μc).

Para su ejecución la tesis se ha dividido en cuatro capítulos.

En el capítulo I se describe el proceso, se enuncia el problema de control y se presenta la lógica de control.

En el capítulo II se presenta las características generales del microcontrolador 80C535, tales como la organización de la memoria del microcontrolador, el control de periféricos, puertos de entrada y salida y su programación.

En el capítulo III se presenta el software y el hardware para el proceso de elevación y separación de paquetes. Se indican los dispositivos de entrada y

de salida, se realiza la programación de la lógica operacional en lenguaje assembler en base a la lógica de control en diagrama escalera.

En el capítulo IV se presentan los costos del hardware y software del proyecto.

Adicionalmente a estos cuatro capítulos se presentan las conclusiones derivadas de este trabajo y al final la bibliografía del caso.

Finalmente deseo agradecer al Ing. Javier Donayre por su valiosa información sobre el microcontralador 80C535. Agradezco en forma muy especial a mi asesor el Ing. Rubén Aquize Palacios, quién generosamente empleó su tiempo y su talento ayudándome con sus valiosos comentarios y sugerencias en la realización de este proyecto. También agradezco al profesor Armando Cajahuaringa Camaco por su ayuda en el manejo de diversos softwares utilizados en la elaboración de esta tesis.

CAPITULO I
DESCRIPCION DEL PROCESO

1.1 Introducción

El proceso de elevación y separación de paquetes que se controlará, pertenece al Laboratorio de Control y Automatización de la FIEE y se emplea para la enseñanza de la neumática y el control de sistemas secuenciales basados en PLCs. La Figura 1.1 muestra las vistas frontal y superior del proceso, construido con material de acero inoxidable y perfiles de fierro, en donde están montados tres cilindros neumáticos de doble efecto (A, B y C). También se puede observar que posee un alimentador vertical y tres transportadores a polines. En este proceso se montan sensores de posición mecánico, interruptores de fin de carrera, para los vástagos de los cilindros A y B, y sensores magnéticos para el vástago del cilindro C. Asimismo, frente al transportador a polines de alimentación de paquetes se monta un sensor de presencia óptico y junto a los cilindros B y C se montan sensores de presencia inductivo y óptico, respectivamente, para la selección de paquetes.

1.2 Enunciado del problema

Los paquetes que llegan por el primer transportador a la plataforma que está al final del vástago del cilindro A, son detectados por un sensor de presencia óptico; luego empujados por dicho cilindro a otro nivel donde se

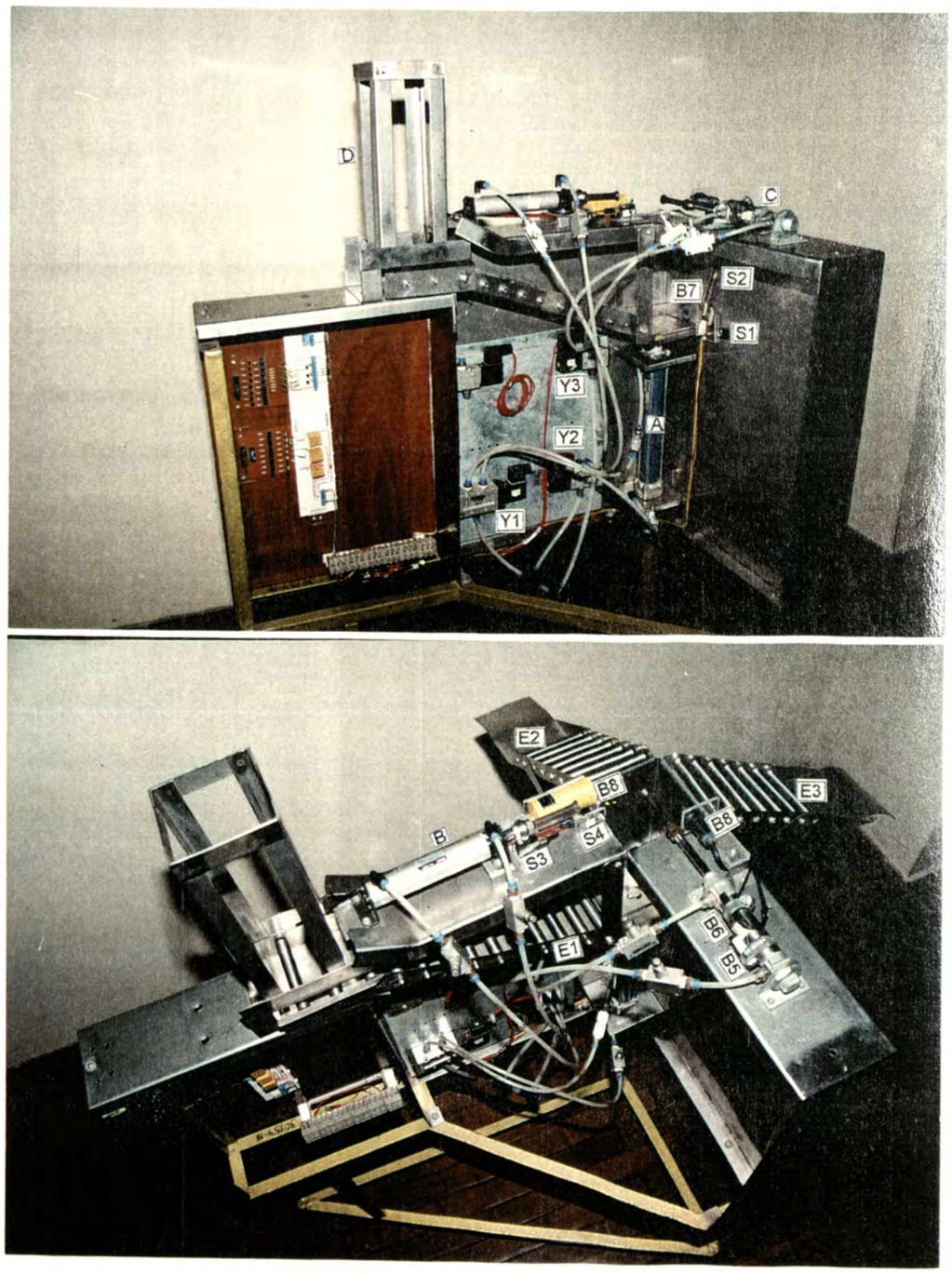


Figura 1.1.: Vista frontal y superior de la estructura metálica

LEYENDA:

A: Cilindro de doble efecto para elevación de paquete.

B y C: Cilindros de doble efecto para selección de paquetes.

Y1: Electroválvula tipo 5/2, para control del cilindro A.

Y2: Electroválvula tipo 5/2, para control del cilindro B.

Y3: Electroválvula tipo 5/2, para control del cilindro C.

S1 y S2: Interruptores de fin de carrera mecánicos para el control del cilindro A.

S3 y S4: Interruptores de fin de carrera mecánicos para el control del cilindro B.

B5 y B6: Sensores de Presencia Magnéticos para controlar el cilindro C.

B7: Sensor de presencia óptico para detectar presencia de paquete sobre plataforma del vástago del cilindro A.

B8: Sensor de presencia inductivo para seleccionar paquetes metálicos.

B8: Sensor de presencia óptico para seleccionar paquetes por el tamaño o por color.

D: Alimentador Vertical.

E1: Transportador de polines para transportar paquete al cilindro A.

E2 y E3: Transportador de polines para separación de paquetes.

encuentran ubicados los cilindros B y C, de donde serán separados de acuerdo a:

1. ***Tamaño de paquete***

Si el paquete elevado es grande, será empujado por el cilindro B y derivado por un transportador a polines hacia una faja transportadora. Si el paquete es chico, será empujado por el cilindro C y derivado por un tercer transportador a polines a una segunda faja transportadora.

Para este caso la discriminación de tamaño es realizada por el sensor de presencia óptico montado al costado del cilindro C.

2. ***Tipo de material***

Si el paquete elevado es de metal, será empujado por el cilindro B y derivado por un transportador a polines hacia una faja transportadora. Si el paquete elevado no es de metal, será empujado por el cilindro C y derivado por un transportador a polines hacia una faja transportadora.

La discriminación por el tipo de material es realizada por el sensor de presencia inductivo montado al costado del cilindro B.

3. ***Colores diferentes***

Si el paquete elevado es de color claro, será empujado por el cilindro B y derivado por un transportador a polines hacia una faja transportadora. Si el paquete es de color oscuro, será empujado por el cilindro C y derivado por un transportador a polines hacia una faja transportadora.

La discriminación de acuerdo al color, será realizada por el sensor de presencia tipo óptico montado al costado del cilindro C.

1.3 Lógica de control

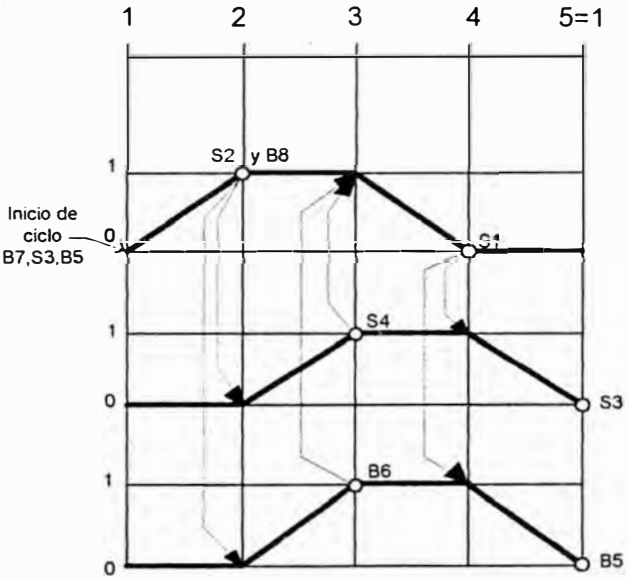
Para la elaboración de la lógica de control con el diagrama escalera, emplearemos los siguientes diagramas de espacio-fase.

1. Separación de paquetes por tamaño

Tal como se muestra en la figura 1.2.

Separación de paquetes por tamaño

Elemento de trabajo	Elemento de señal
A	S1
	S2
B Para paquetes grandes	S3
	S4
C Para paquetes pequeños	B5
	B6



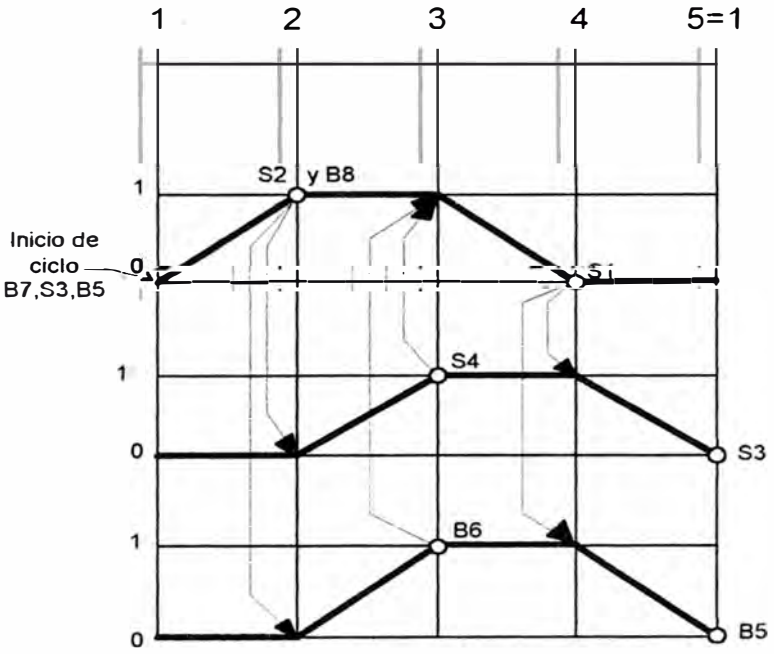
B8: Sensor de presencia tipo óptico

Figura 1.2: Diagrama de espacio-fase para separación de paquetes por tamaño.

2. Separación de paquetes por tipo de material

Tal como se muestra en la figura 1.3.

Elemento de trabajo	Elemento de señal
A	S1
	S2
B Para paquetes metálicos	S3
	S4
C Para paquetes no metálicos	B5
	B6



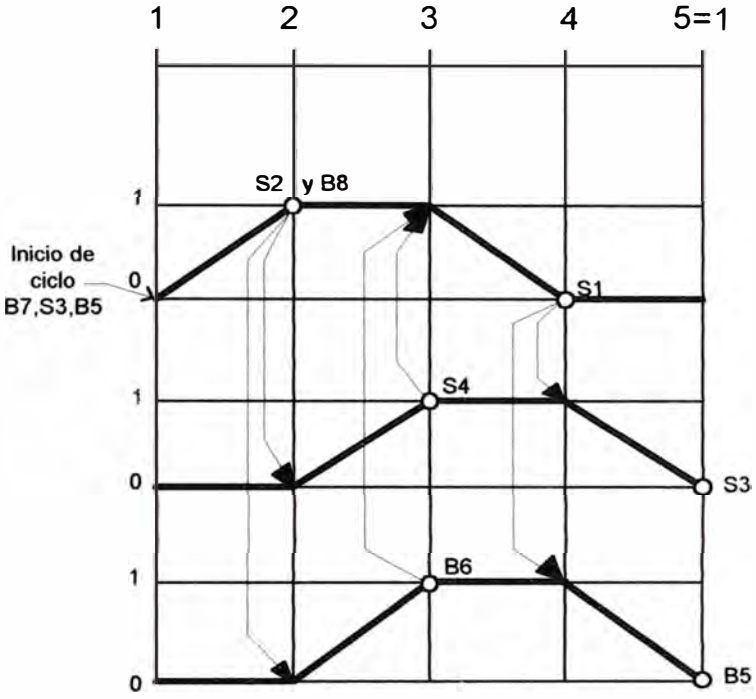
B8: Sensor de presencia tipo inductivo

Figura 1.3: Diagrama de espacio-fase para separación de paquetes por tipo de material.

2. *Separación de paquetes por color*

Tal como se muestra en la figura 1.4.

Elemento de trabajo	Elemento de señal
A	S1
	S2
B Para paquetes claros	S3
	S4
C Para paquetes oscuros	B5
	B6



B8: Sensor de presencia tipo óptico

Figura 1.4: Diagrama de espacio-fase para separación de paquetes por color

Tomando en consideración los diagramas de espacio-fase de las figuras 1.2, 1.3 y 1.4 se construye la lógica de control en diagrama escalera mostrada en el capítulo III, sección 3.4

CAPITULO II
CARACTERÍSTICAS GENERALES DEL MiCROCONTROLADOR 80C535

2.1 Introducción

El microcontrolador SAB 80C535 es un poderoso miembro de la familia de microcontroladores de 8 bits del SAB 8051 de SIEMENS. Está diseñado con tecnología CMOS y basado en la arquitectura del 8051, por lo que es totalmente compatible a dicha familia. El rendimiento de sus periféricos se han incrementado significativamente. Algunos de sus periféricos han sido agregados en el interior del chip de 8 bits sin perder compatibilidad con el 8051. Incorpora varias mejoras con funciones adicionales, las cuales incrementan la flexibilidad de diseño y un mayor campo de aplicación.

Las propiedades de bajo consumo de energía de la tecnología CMOS permite aplicaciones donde el consumo de energía y la disipación son críticos. Además tiene dos modos de actividad reducida seleccionables mediante software para una mayor reducción de energía: modo en reposo y modo de bajo consumo.

Al SAB80C535 le falta en el chip la memoria de programas y es fabricado en un chip de plástico de 68 pines.

Las principales funciones del microcontrolador SAB80C535 son las siguientes:

- 256 Bytes de RAM interno

- 3 Timers/Counter de 16 bits, uno de ellos con PWM.
- Conversor A/D, con 8 entradas analógicas de 8 bits (expansible mediante software, a 10 bits)
- Generador de baudrate programable.
- 4 Registros Capture/Compare de 16 bits.
- 12 señales de interrupción con 4 niveles de prioridad.
- Watchdog-Timer de 16 bits, vigila el funcionamiento óptimo del CPU.
- 6 Puertos paralelos de I/O de 8 bits (Port 0, ..., Port 5).
- Hardware Power-Down.
- Slow-Down modus (Idle).
- Procesador booleano.

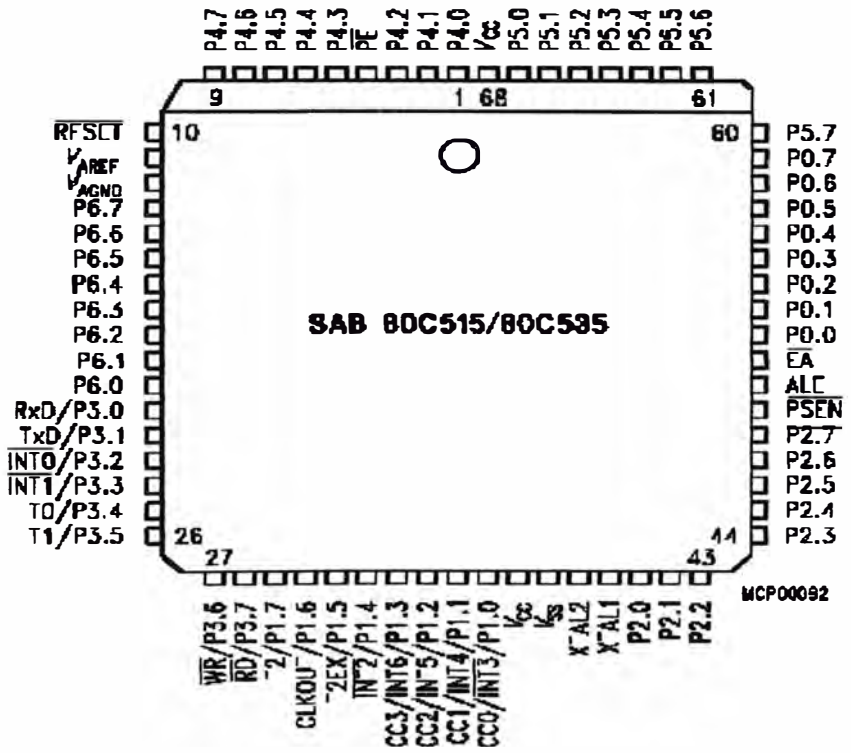


Figura 2.1: pines del SAB 80C515/80C535

La tarjeta 80C535 Compuboard es una tarjeta que contiene un microcontrolador 80C535 de SIEMENS y es empleada no sólo para el aprendizaje sino también en todo tipo de aplicaciones, sean estas industriales, de laboratorio, o sistemas encapsulados.

Las características de esta tarjeta son:

- Microcontrolador 80C535 (SIEMENS).
- 12 MHZ CLOCK
- 32 KB CMOS RAM
- 32 KB EPROM
- Interface RS-232.
- Botón RESET.
- Alimentación +5V DC.
- Dimensión: 115x68mm.

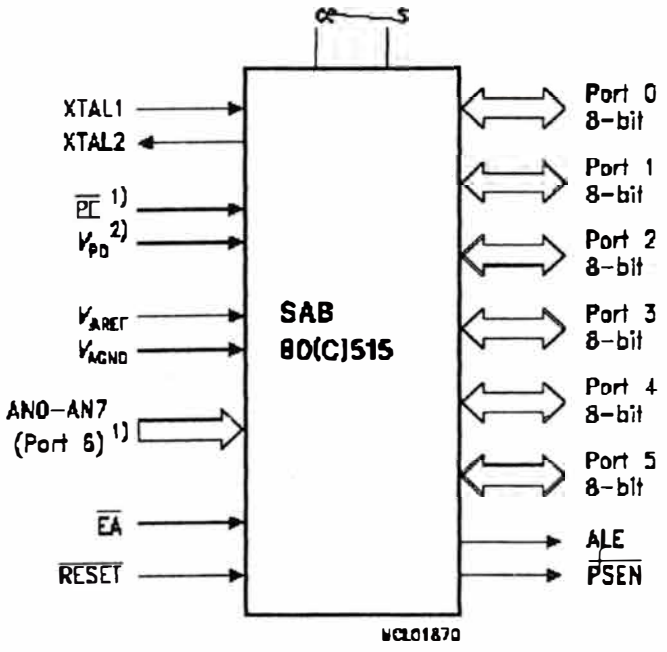


Figura 2.2: Símbolo lógico

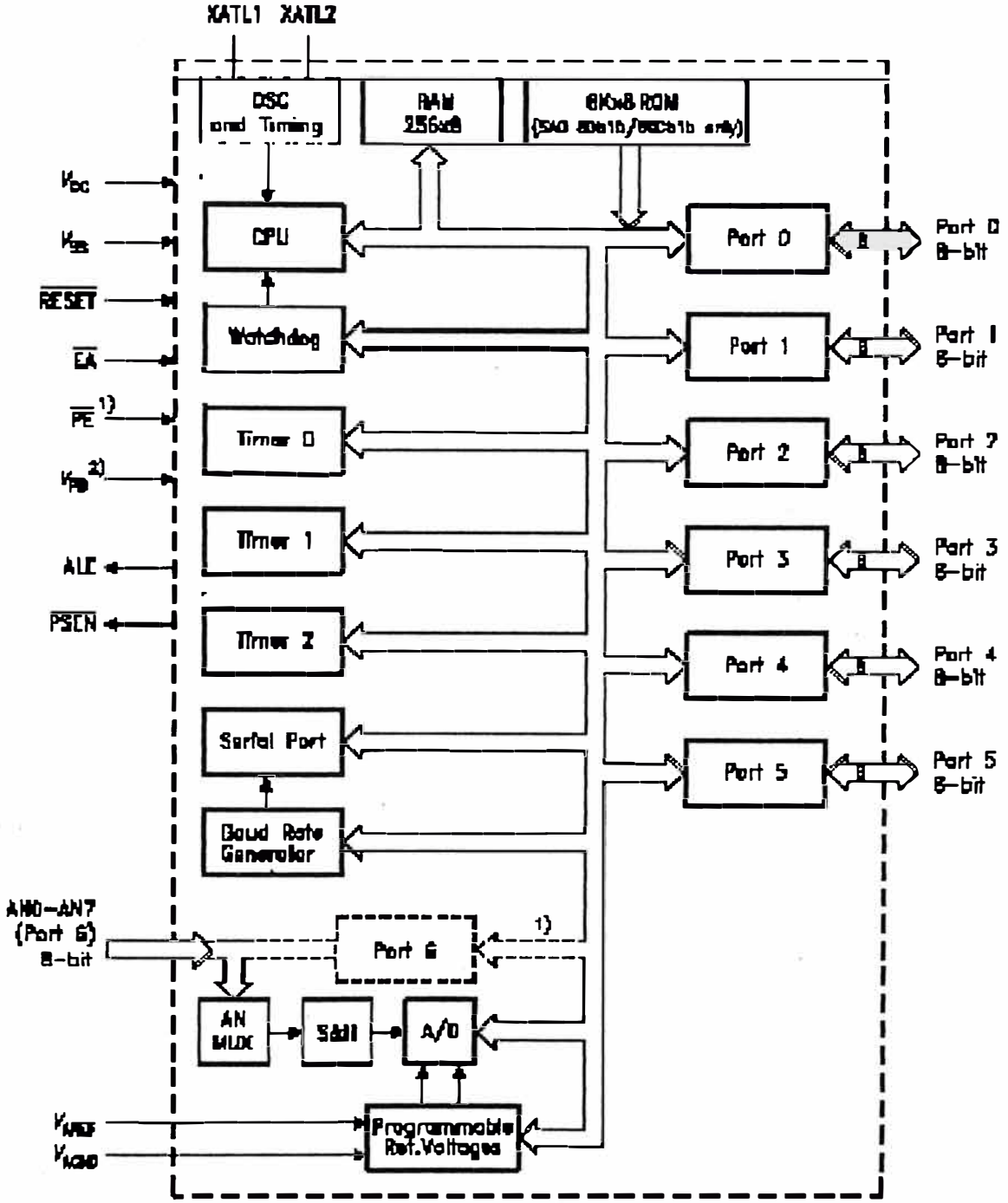


Figura 2.3: Diagrama de bloque del SAB 80C535

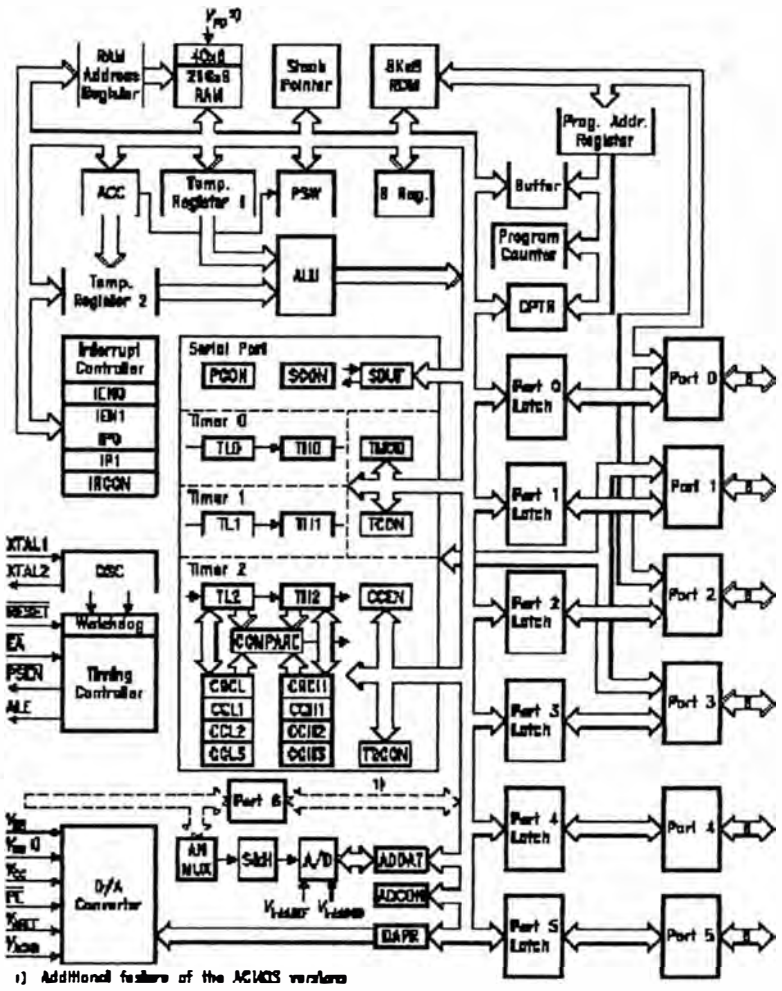


Figura 2.4: Diagrama de bloque detallado del SAB 80C535

2.2 Organización de la memoria del microcontrolador

El SAB80C535 tiene separado los espacios de direcciones para memoria de programas y memorias de datos y manipula operandos en los siguientes cuatro espacios de dirección:

- Hasta 64 KBytes de Memoria de Programa.
- Hasta 64 KBytes de Memoria de Datos Externa.
- 256 Bytes de Memoria de Datos Interna.
- Unos 128 Bytes de área de Registros de Funciones Especiales.

2.2.1 Memoria de programas

La memoria de Programas sólo puede ser leída y tiene como máximo 64 KB. El SAB80C535 no tiene ningún ROM interior; es decir, no tiene Memoria Interna de Programas. Una EPROM es su Memoria Externa de Programas. El pin $\overline{EA}$ debe mantenerse bajo al usar el 80C535; con ello se extrae todas las instrucciones de la Memoria Externa de Programas. El contador de programa, de 16 bits, se direcciona mecánicamente.

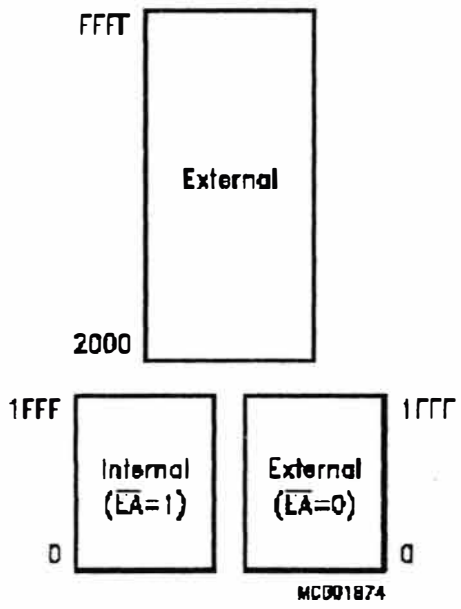


Figura 2.5 : Espacio de las direcciones de la memoria de programa

2.2.2 Memoria de datos

La memoria de datos admite operaciones de lectura y escritura. El espacio de las direcciones de la memoria de datos consiste de una porción de memoria interna y una porción de memoria externa, y puede direccionar hasta 64 KB.

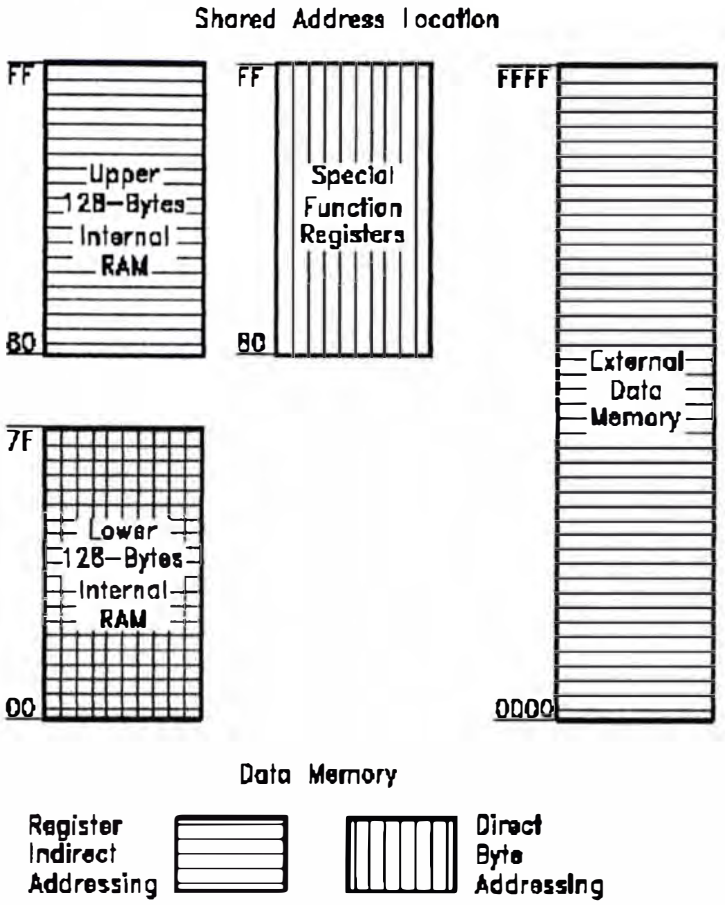


Figura 2.6: Mapa de la porción inferior de la memoria interna de datos

MEMORIA INTERNA DE DATOS

La memoria interna del SAB 80C535 es de 256 bytes de RAM interno. El espacio de las direcciones de la memoria interna de datos está dividida físicamente en tres bloques distintos: los 128 bytes inferiores del RAM, los 128 bytes del área superior del RAM, y los 128 bytes del área de Registros de Funciones Especiales (SFR). Puesto que el último área de SFR y el área superior del RAM comparten los mismos espacios de dirección, ellos deben ser accedidos a través de diferentes modos de direccionamientos. El mapa de la figura 2.6 y el siguiente cuadro muestra los modos direccionamiento usados para los diferentes espacios del RAM/SFR.

Espacio de Direcciones	**Localización**	**Modo de Direccionamiento**
128 bytes inferiores del RAM	00H al 7FH	Directo-Indirecto
128 bytes superiores del RAM	80H al 0FFH	Indirecto
Registros de Función Especial	80H al 0FFH	Directo

Los 128 bytes inferiores del RAM interna son a su vez agrupados en tres espacios de direcciones (ver figura 2.5):

1) Un área de registros de propósito general se localizan desde la dirección 00H a 1FH (32 bytes).
2) Los siguientes 16 bytes, localización del 20H hasta el 2FH. Cada uno de los 128 bits de este segmento se puede direccionar directamente (00H a 7FH).

Programando información.- Estos bits de direccionamiento pueden ser referidos de dos formas diferentes, bien por sus direcciones (bits 00H hasta bits 7FH), o por bytes que los contienen (20H a 2FH).

Esto es, los bits 0 al 7 pueden ser referidos como los bits 20.0 a 20.7, y del 08H al 0FH, como 21.0 a 21.7, etc. Cada uno de los 16 bytes en este segmento también pueden ser direccionados como un byte.

3) Las localizaciones del 30H al 7FH pueden ser usados como un área de memoria "scratch pad", el cual se entiende como la memoria de un block de notas de rápido acceso pero de escasa capacidad. Esta es la memoria de trabajo (RAM) del usuario.

Usando el registro SP (Stack Pointer) indicador de la memoria de pila, es un registro de 8 bits. Inicializado el microcontrolador por la operación Reset,

por defecto el SP se carga con la dirección 07H (SP ← 07). Al ejecutar la primera instrucción PUSH o CALL el SP se carga con 08H, apuntando a dicha posición de memoria.

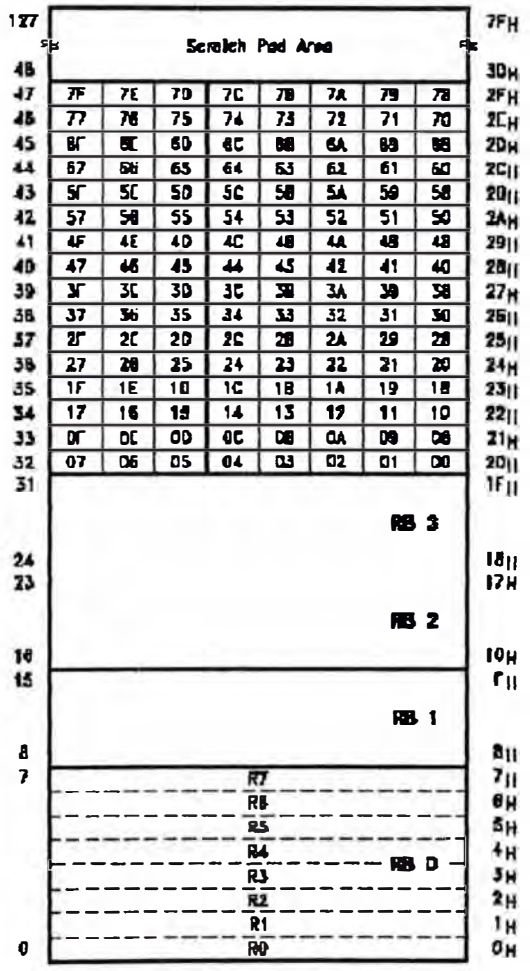


Figura 2.7: Mapa de la porción inferior de la memoria interna de datos

MEMORIA EXTERNA DE DATOS

Las figuras 2.6 y 2.7 contienen el mapa de memoria que ilustra la memoria interna/externa de datos. El 80C535 puede direccionar hasta un máximo de 64 Kbytes de memoria externa de datos, puede ser accedida por la instrucción MOVX que utiliza 16 bits de dirección.

Symbol	Name	Address
P0	Port 0	80H
SP	Stack pointer	81H
DPL	Data pointer, low byte	82H
DPH	Data pointer, high byte	83H
PCON	Power control register	87H
TCON	Timer control register	88H
TMOD	Timer mode register	89H
TL0	Timer 0, low byte	8AH
TL1	Timer 1, low byte	8BH
TH0	Timer 0, high byte	8CH
TH1	Timer 1, high byte	8DH
P1	Port 1	90H
SCON	Serial channel control register	98H
SBUF	Serial channel buffer register	99H
P2	Port 2	0A0H
IEN0	Interrup enable register 0	0A8H
IP0	Interrup priority register 0	0A9H
P3	Port 3	0B0H
IEN1	Interrup enable register 1	0B8H
IP1	Interrup priority register 1	0B9H
IRCON	Interrup request control register	0C0H
CCEN	Compare /capture enable register	0C1H
CCL1	Compare /capture register 1, low byte	0C2H
CCH1	Compare /capture register 1, high byte	0C3H
CCL2	Compare /capture register 2, low byte	0C4H
CCH2	Compare /capture register 2, high byte	0C5H
CCL3	Compare /capture register 3, low byte	0C6H
CCH3	Compare /capture register 3, high byte	0C7H
T2CON	Timer 2 control register	0C8H
CRCL	Compare relead capture register, low byte	0CAH
CRCH	Compare relead capture register, high byte	0CBH
TL2	Timer 2, low byte	0CCH
TH2	Timer 2, high byte	0CDH
PSW	Program status word register	0D0H
ADCON	A/ D converter control register	0D8H
ADDAT	A/ D converter data register	0D9H
DAPR	D/ A converter program register	0DAH
P6	Port 6	0DBH
ACC	Accumulator	0B0H
P4	Port 4	0B8H
B	B register	0F0H
P5	Port 5	0F8H

Figura 2.8: Lista de Registros de Funciones Especiales (SFR).

2.2.3 Expansiones en el microcontrolador 80C535

La expansión de memoria externa en el microcontrolador 80C535 es posible direccionarla hasta 64 KB, usando 16 bits de dirección.

El SAB 80C535 no tiene ROM interno. La memoria externa de programa puede ser expandida hasta 64 Kbytes. La tarjeta 80C535 Compuboard tiene un EPROM de 32 Kbytes de memoria externa de programa , se podría extender la memoria externa de programa mediante un EPROM o ROM de 32 Kbytes para completar los 64 Kbytes máximos permitidos.

2.3 Control de periféricos, puertos de entrada y salida

El SAP 80C535 permite para I/O digital en 48 líneas agrupadas en seis puertos bidireccionables de 8 bits, es decir, permiten la lectura y escritura en el periférico correspondiente. Cada uno de los bits del puerto consiste en un latch, un driver de salida y un buffer de entrada. El acceso a la lectura y escritura en la I/O del puerto P0 al P5 son ejecutados vía sus correspondientes registros de funciones especiales. El control de los puertos van a permitir comunicarse con una buena parte de los periféricos comerciales y con los que usuario diseñe.

Operación de escritura en los puertos del microcontrolador 80C535, puede ser realizada por cualquier puerto, no obstante, el puerto P0 es el que presenta una mayor cargabilidad, permitiendo comandar 8 cargas TTL-LS, mientras que los otros puertos permiten cuatro cargas TTL-LS. En la ejecución de una instrucción que cambia el valor del latch del puerto, el nuevo valor llega al latch durante el estado 6, fase 2 del final de ciclo de

instrucción. Para comandar cargas de mayor consumo energético, como relés, se recomienda utilizar, entre el puerto y la carga, drivers no inversores, como el ULN 2003, o ınversores, como el ULN 2803, que tienen una cargabilidad de 500mA y soportan hasta 50V. La figura 2.9 muestra el esquema circuital de una de las interfaces de salida construidas para este proyecto. El 80C535 presenta dos señales de control, una para ejecutar la lectura (RD) y otra para la escritura (WR). Por esto la activación de estas señales depende del formato de la instrucción.

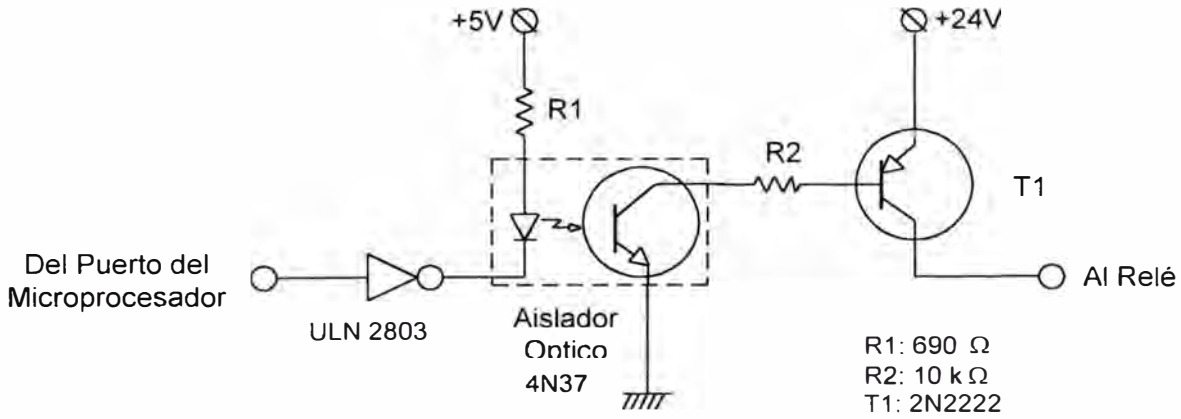


Figura 2.9 : Interfase de salida

Para la operación de escritura en el puerto, la instrucción más habitual es la siguiente:

MOV PX, <DATO> ; PX ⟵ <DATO>

X toma valores 0, 1, 2, 3, ..., 5 según el puerto.

Admitiendo <DATO> todos los tipos de direccionamientos mencionados.

La operación de lectura o de adquisición de datos no representa ningún tipo de problema; solamente se deberá cambiar el orden de los operandos en la instrucción respecto a la operación de escritura.

Para la operación de lectura, el formato de la instrucción más habitual es el siguiente:

MOV <DATO>, PX ; <destino> ◄— PX

La figura 2.10. muestra el esquema circuital de unas ocho interfaces de entrada construidas para este proyecto.

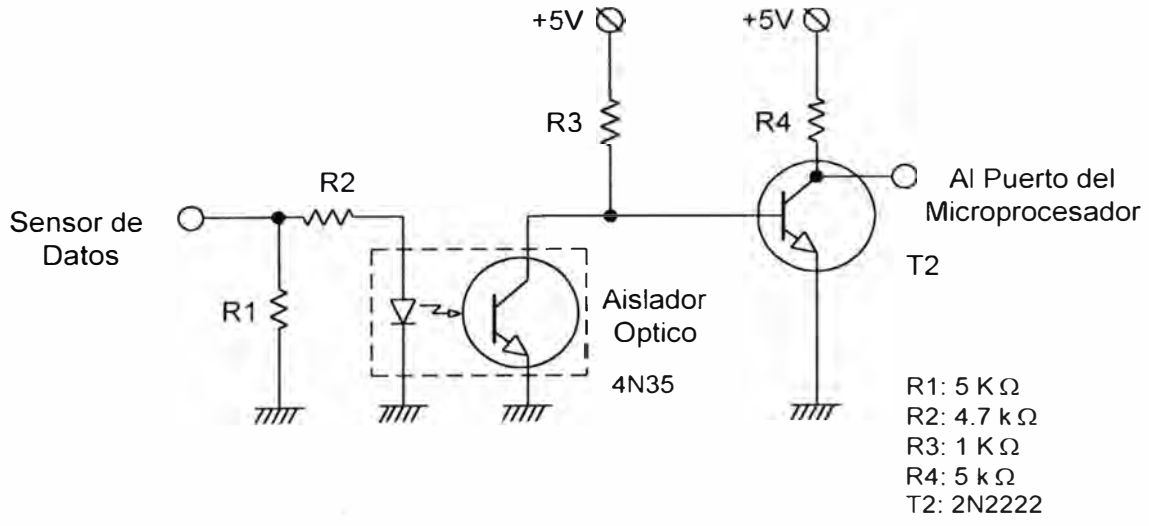


Figura 2.10: Interfase de entrada

2.3.1 Ampliación de las líneas de entrada y salida

Debido al uso de un sistema flexible como es el microcontrolador, nosotros podemos ampliar el proyecto para implementar procesos adicionales. En caso que las líneas de entrada/ salida (I/O)de este microcontrolador sean insuficientes se puede ampliar con facilidad las líneas

I/O con ayuda de un integrado, como por ejemplo el PPI 8055 el que nos permite con 12 salidas ampliarlo a 24.

2.4 Programación del microcontrolador 80C535

2.4.1 Introducción

El conjunto de instrucciones del SAB 80C535 incluye 111 instrucciones, 49 de las cuales son de simple byte, 45 de dos-bytes y 17 de tres-bytes. El formato de la instrucción del código de operación consiste de una función nemónica seguido por un "destinatario, fuente" campo operando. Estos campos especifican el tipo de dato y métodos de direccionamiento a ser usados.

Como todos los otros miembros de la familia 8051, el 80C535 puede ser programado con las mismo conjunto de instrucciones común al miembro básico, el 8051. Esto es, el 80C535 es 100% software compatible a el 8051 y puede ser programado con asembler 8051 o lenguaje de alto nivel.

2.4.2 Modos de direccionamiento

El SAB80C535 usa cinco modos de direccionamiento:

- registro
- directo
- inmediato
- registro indirecto
- registro base más registro índice indirecto

Direccionamiento por Registro

Accede a uno de los 8 registros del banco seleccionado. El bit menos significativo de la instrucción código de operación indica el registro a ser

usado. ACC, B, DPTR y CY, el acumulador del procesador booleano, pueden también ser direccionados como registros.

Direccionamiento Directo

El operando se especifica en la instrucción por un campo de dirección de 8 bits. Es el único método de acceder los registros de función especial. Los 128 bytes inferiores del RAM interno de datos son también direccionados directamente.

Direccionamiento Inmediato

Al código de operación le sigue una constante en la memoria de programas.

Direccionamiento Registro Indirecto

La instrucción especifica un registro que contiene la dirección del operando. Los registros para direccionar sobre el mapa de 8 bits pueden ser el R0 y R1del banco de registros seleccionado, como un puntero de localización en el bloque de 256-byte: los 256 bytes del RAM interno o los 256 bytes inferiores de la memoria de datos externo. Notar que los registros de función especial no son accesibles por este método. La mitad superior de la RAM interna puede ser accedida únicamente por direccionamiento indirecto. Acceso a la totalidad de los 64 Kbytes del espacio de direcciones de la memoria de datos externo es logrado por usar el puntero de dato de 16 bits. Ejecución de las instrucciones PUSH y POP también usan direccionamiento de registro indirecto. La pila puede residir en cualquier parte en el RAM interno.

Direccionamiento registro base más registro índice

Este direccionamiento sólo es posible en la memoria de programas y sólo permite la lectura. Es utilizado para la lectura de tablas.

Permite acceder a un byte de la memoria de programas vía un movimiento indirecto de la localización cuya dirección es la suma de un registro base de 16 bits (DPTR o PC) apunta a la base de la tabla y el contenido del acumulador, ACC, es el desplazamiento que permite acceder a la lectura de esa posición de la tabla. Este modo facilita ver la tabla de accesos.

2.4.3 Tipos de instrucciones

El conjunto de instrucciones esta dividido en cuatro grupos funcionales:

- transferencia de dato
- aritmética
- lógica
- transferencia de control

Transferencia de Dato

Operaciones de datos están divididas en tres clases:

- de uso general
- acumulador -especifico
- dirección-objeto

Ninguno de estas operaciones afectan el flag PSW establecido excepto un POP o MOV directamente a el PSW.

Instrucciones Aritméticas

El SAB 80C535 tiene las cuatro operaciones básicas de matemáticas (adición, sustracción, multiplicación y división). Unicamente operaciones de 8 bits usando aritmética sin signo son sostenidos directamente. El flag overflow, permite las operaciones de adición y sustracción sirve para ambos casos sin signo y con signo binarios enteros. Aritmética también puede ejecutar directamente sobre paquetes de representación BCD.

Instrucciones Lógicas

El SAB 80C535 ejecuta operaciones de lógica básica tanto como operador bit y byte.

Instrucciones de Transferencia de Control.

Hay tres clases de operaciones de transferencia de control: llamadas incondicionales, retornos, saltos, saltos condicionales e interrupciones. Todas las operaciones de transferencia de control, algunas con una especifica condición, causa la ejecución del programa para continuar una no secuencial localización en el programa de memoria.

CAPITULO III
SOFTWARE Y HARDWARE PARA EL PROCESO DE ELEVACION Y SEPARACION DE PAQUETES

3.1 Elementos del proceso

El proceso está constituido por tres transportadores de polines: uno que transporta los paquetes hacia la plataforma del elevador (cilindro A), un segundo transportador para transportar los paquetes pequeños (ó paquetes de color oscuro y paquetes no metálicos) hacia una faja transportadora Nº 1, y uno tercero que transporta los paquetes grandes (paquetes de color claro y paquetes metálicos) hacia una faja transportadora Nº 2. Adicionalmente, el proceso está constituido por un alimentador vertical y tres cilindros de doble efecto. Conjuntamente con estos elementos están montados en una estructura metálica con planchas de acero inoxidable y perfiles de fierro, los dispositivos de entrada y salida del microcontrolador, que se detallan en las siguientes secciones.

Es preciso mencionar que antes de iniciarse el proceso de elevación y separación de paquetes se debe garantizar el funcionamiento de las fajas transportadoras Nº 1 y Nº 2 para evitar acumulación de paquetes en las fajas. Los circuitos de fuerza y control de estas fajas se muestran en la figura 3.12.

3.2 Dispositivos de entrada

Los dispositivos de entrada al microcontrolador son en total ocho y son

los siguientes (Especificaciones técnicas en Anexo C1):

Interruptores de fin de carrera (S1, S2, S3 y S4). Los dos primeros trabajan con el vástago del cilindro A y los dos restantes con el vástago del cilindro B; además de estos interruptores se tiene dos sensores de presencia magnéticos (B5 y B6) que actúan como finales de carrera con el vástago del cilindro C.

Para detectar paquetes provenientes del transportador 1 el módulo posee un sensor de proximidad óptico (B7) montado cerca de la plataforma del cilindro A. Para la separación de paquetes por tamaño y por color se tiene un sensor de presencia tipo óptico (B8) montado junto al cilindro C. Para la separación de paquetes por tipo de material se tiene un sensor de presencia inductivo (B8) montado junto al cilindro B.

3.3 Dispositivos de salida

Los dispositivos de salida del microcontrolador son en total tres y son los siguientes (Especificaciones técnicas en Anexo C2):

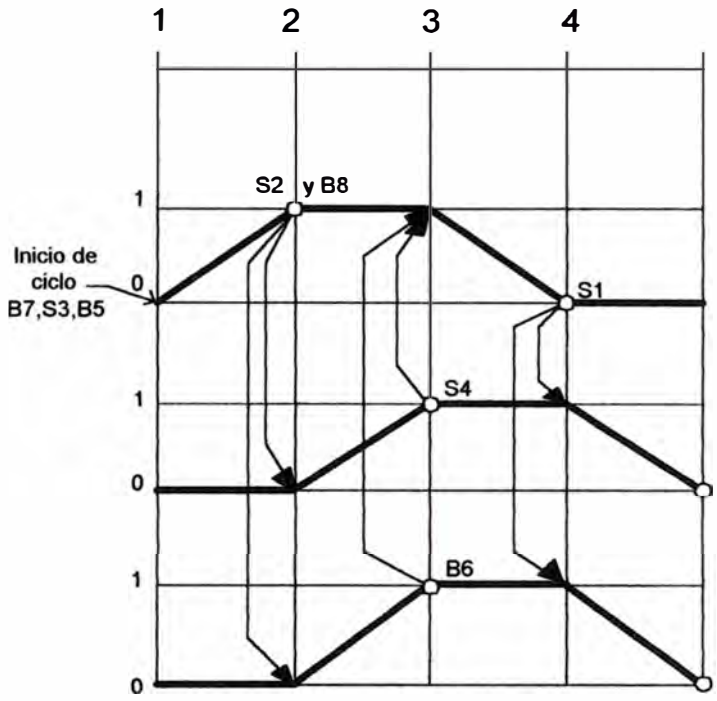
Tres electroválvulas (Y1, Y2 y Y3) tipo 5/2, servopilotadas con "over rider" (control de prueba) manual y con resorte de recuperación, que controlan los cilindros neumáticos A, B y C; es decir, son los elementos de control final.

Estas electroválvulas operan con señal de corriente continua a 24 V DC.

3.4 Síntesis de la lógica de control en diagrama escalera

La síntesis de la lógica de control tomará como base el enunciado del problema mediante la representación de los diagramas de espacio-fase respectivos.

Elemento de trabajo	Elemento de señal
A	S1
	S2
B Para paquetes grandes	S3
	S4
C Para paquetes pequeños	B5
	B6



B8: Sensor de presencia tipo óptico

Figura 3.1: Diagrama de espacio-fase de selección de paquete pequeño y grande

De acuerdo al diagrama espacio-fase de la figura 3.1, para iniciar el ciclo se requiere:

1) Presencia de paquete, proveniente de transportador de polines N. 1, (B7 activado) y que los vástagos de los cilindros B y C estén retraídos (S3 cerrado bajo forzamiento y B5 activado, con estás condiciones se activa el relé K1 (ver figura 3.2) y este a su vez se autosostiene con K1 (13,14) y conmuta mediante K1 (23,24) la electroválvula Y1, con ello el vástago del cilindro A se expande elevando el paquete.

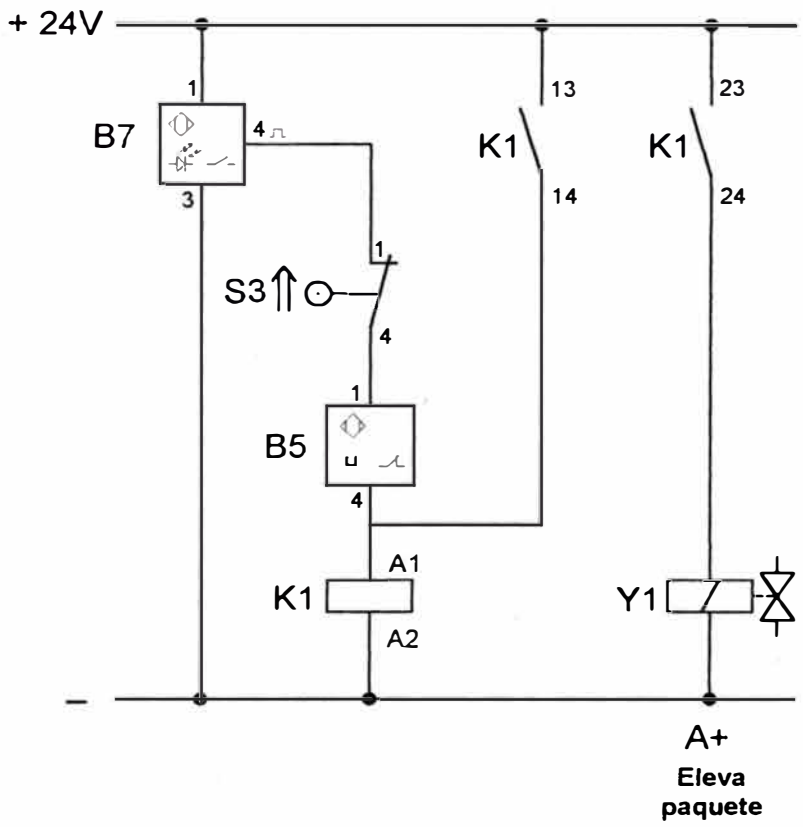


Figura 3.2

2) Cuando el vástago del cilindro A se ha expandido totalmente, cierra el final de carrera S2, y si el paquete elevado es grande, el sensor de presencia óptico B8 cambia de estado sus salidas (cierra el contacto NA y abre el contacto NC). Si empleamos el contacto NA se energizará el relé K2, ausosteniéndose mediante K2 (13,14), y conmuta mediante K2 (23,24) a la electroválvula Y2, lo que propicia la expansión del vástago del cilindro B, que empujará el paquete hacia el transportador de polines Nº 3 (ver figura 3.3).

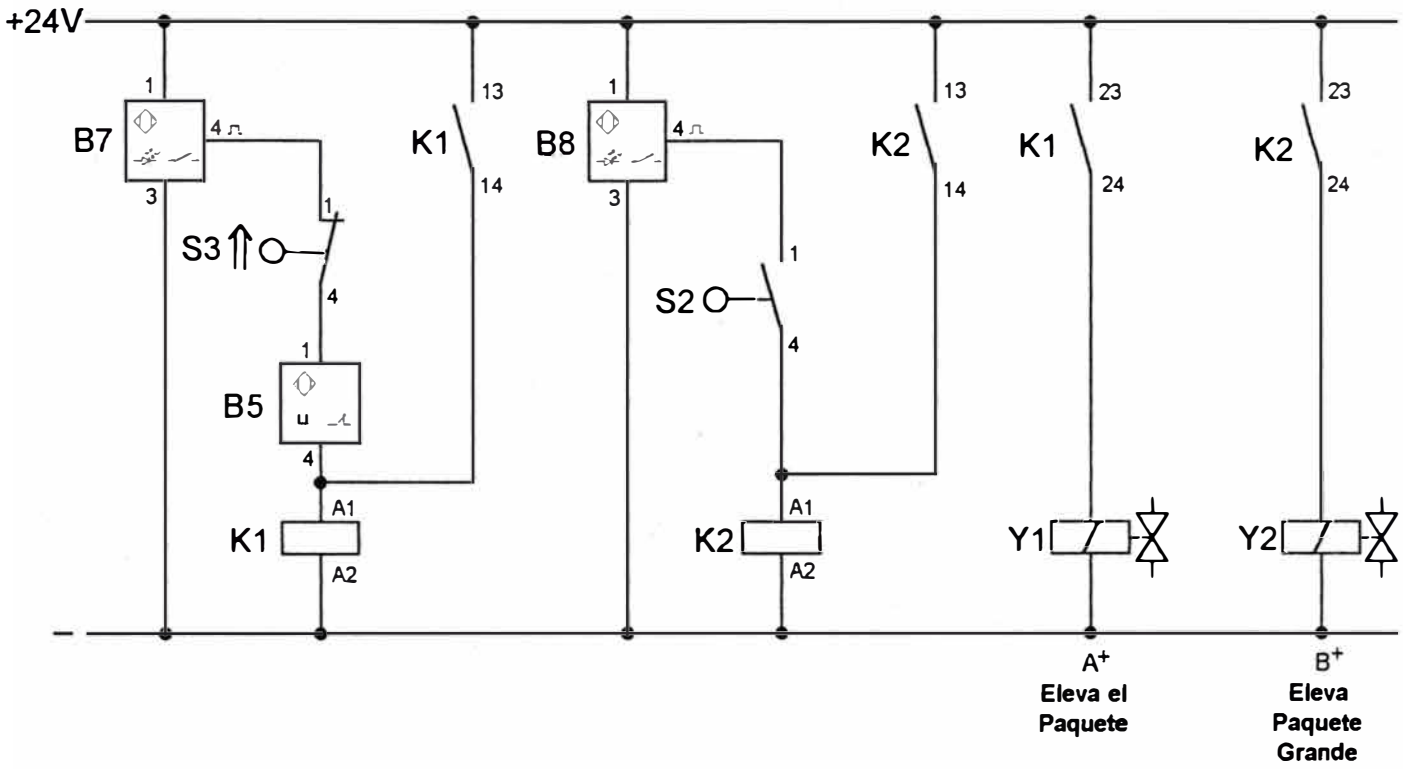


Figura 3.3

3) cuando el vástago del cilindro B se ha expandido totalmente desactivará al interruptor final de carrera S_4, propiciando que se desenergice el relé K1 propiciando que la electroválvula Y1 conmute a su estado anterior, por lo que el vástago del cilindro A se retrae mediante el resorte de recuperación (ver figura 3.4)

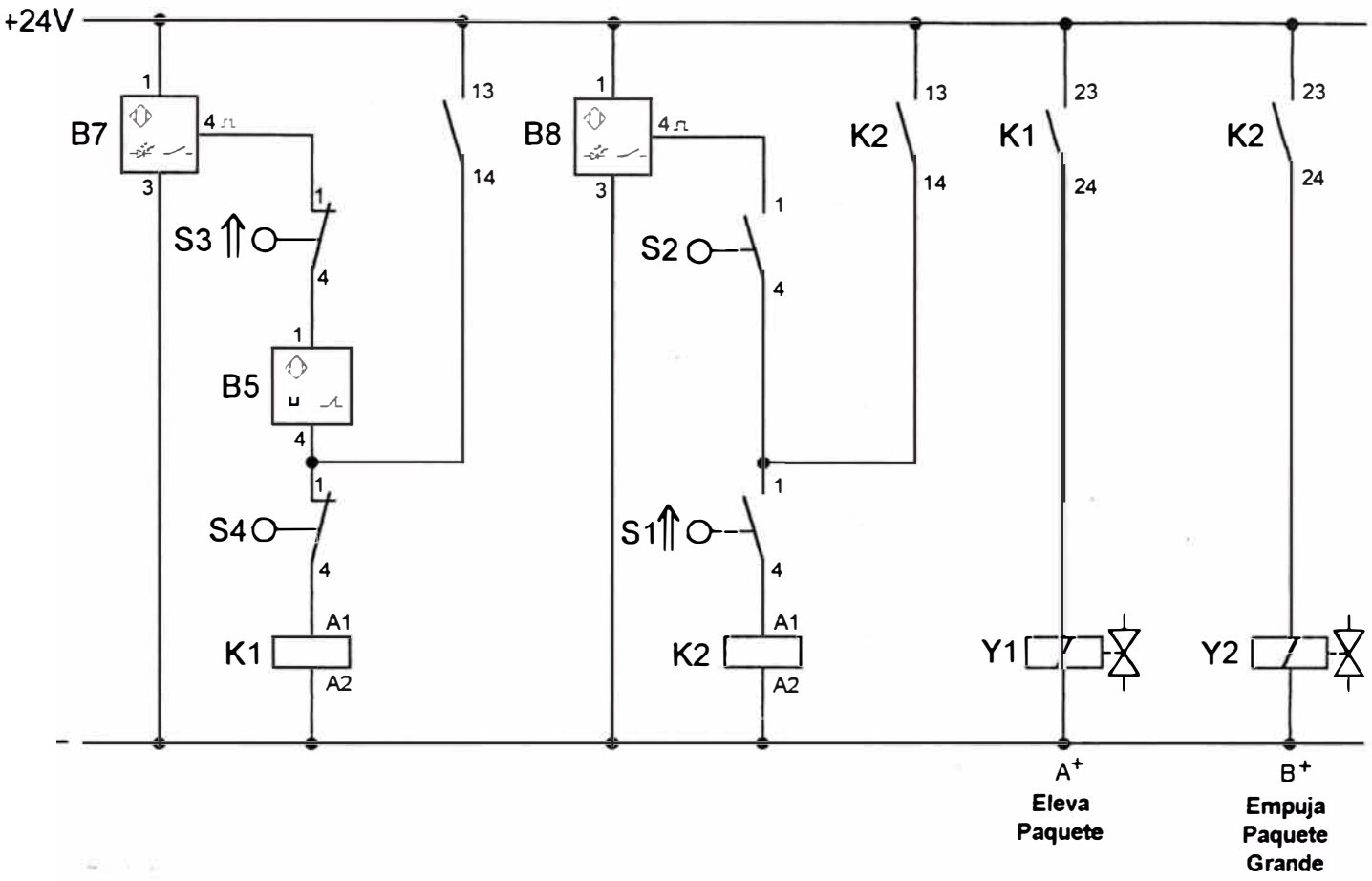


Figura 3.4

4) Cuando el vástago del cilindro A se ha retraído totalmente, vuelve a abrir el contacto NC del final de carrera S_1 (estado de forzamiento), desenergizando el relé K_2 y esta a su vez desactivará la electroválvula Y2, conmutando a su posición anterior mediante el resorte de

recuperación, con lo que el vástago del cilindro B se retrae, concluyéndose así el ciclo de trabajo (ver figura 3.4).

5) Habiendo presencia de paquete proveniente del transportador de polines Nº1, se repetirá el paso 1).

6) Cuando el vástago del cilindro A se ha expandido totalmente, cierra el final de carrera S_2 y si el paquete elevado es pequeño, el sensor de presencia óptico B_8 no lo detecta y no cambia de estado sus salidas. Si en este empleamos su contacto NC se energizará el relé K3. que se autosostiene mediante K3(13, 14), y conmutará la electroválvula Y3, lo que propicia la expansión del vástago del cilindro C, que empujará el paquete pequeño hacia el transportador de polines Nº 2 (ver figura 3.5).

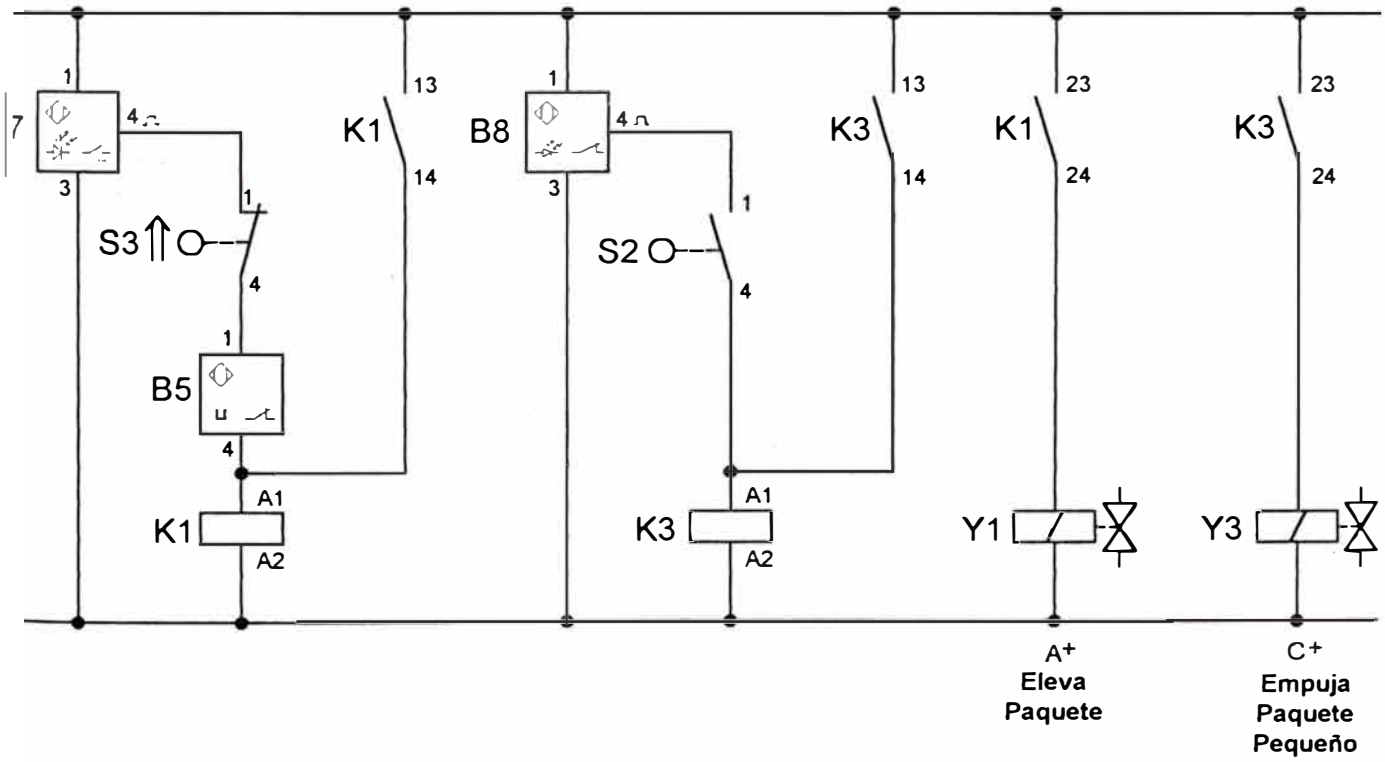


Figura 3.5

7) Cuando el vástago del cilindro C se ha expandido totalmente, desactivará al sensor de presencia magnético B_6 haciendo cambiar de estado a su contacto NC, lo que propiciará la retracción del vástago del

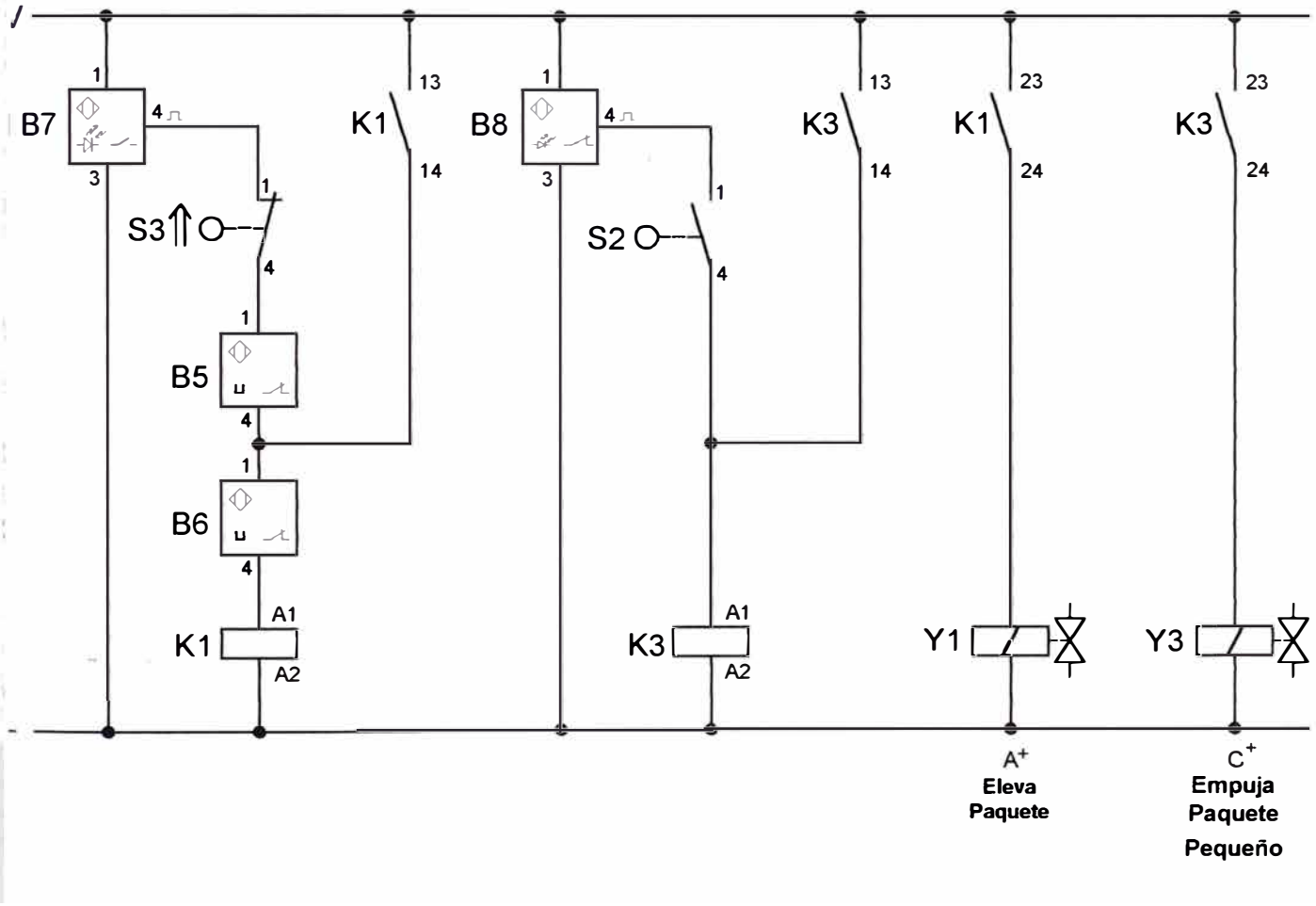


Figura 3.6

cilindro A (ver figura 3.6).

8) Cuando el vástago del cilindro A se ha retraído totalmente, vuelve a abrir el contacto NC del final de carrera S_1 (estado de forzamiento), propiciando que se desenergice el relé K_3 y esta a su vez desactivará la

electroválvula Y_3, conmutando a su posición anterior, mediante el resorte de recuperación, con ello el vástago del cilindro C se retrae, culminando así el ciclo de trabajo (ver figura 3.7).

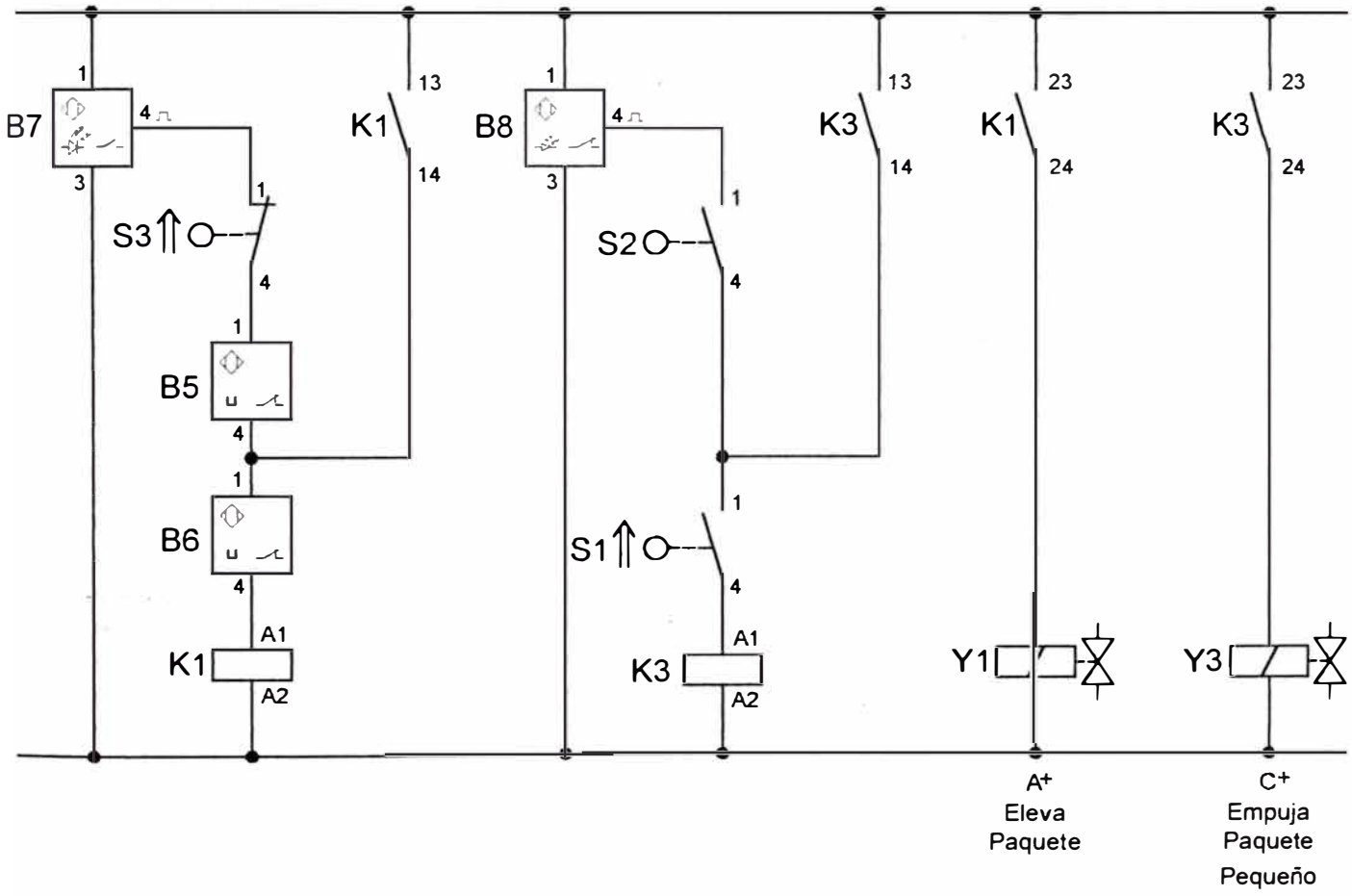


Figura 3.7

La figura 3.8, muestra la lógica de control completa con los ciclos de trabajo de selección de paquetes por tamaño y color.

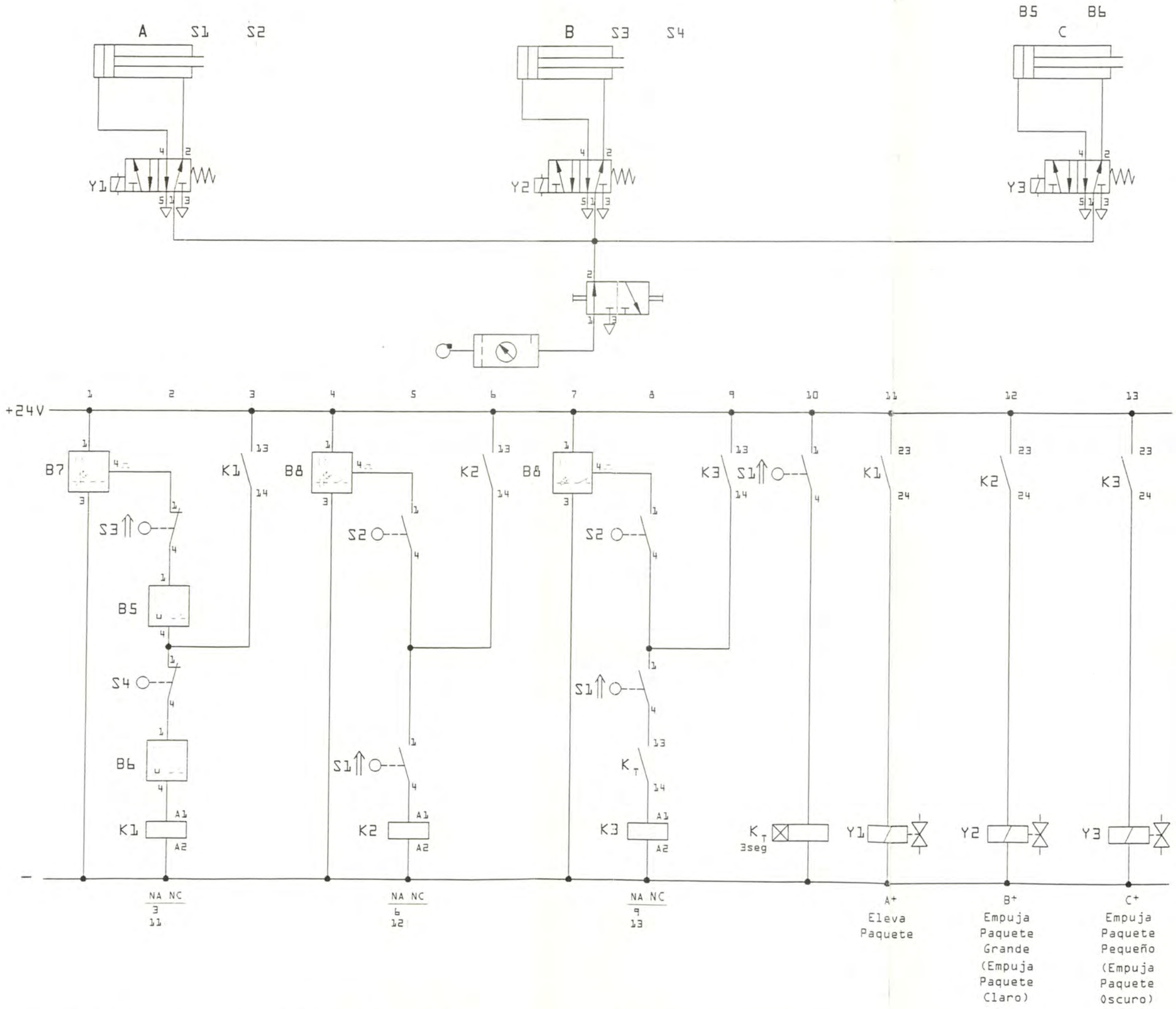
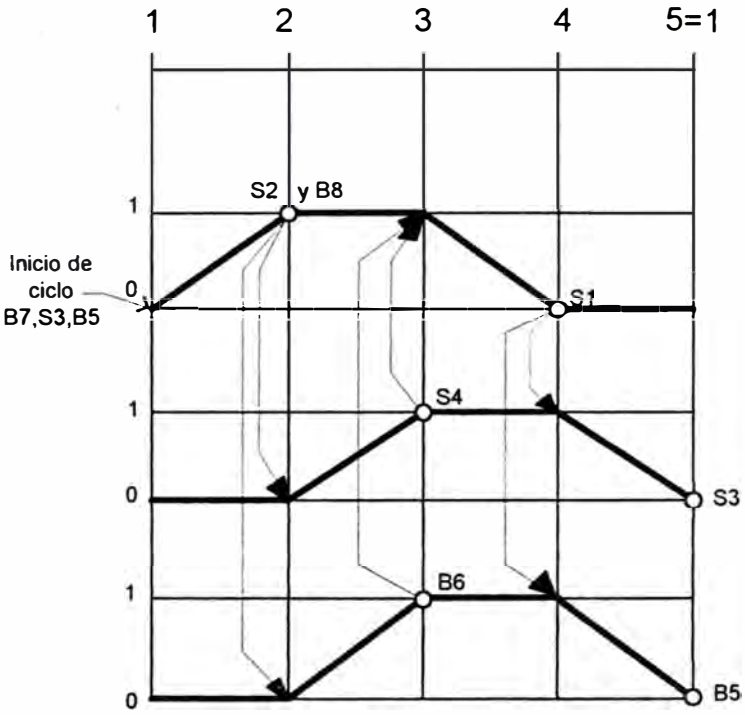


Figura 3.8:ESQUEMAS ELECTRONEUMATICOS PARA EL ELEVADOR Y SEPARADOR DE PAQUETES POR TAMAÑO Y COLOR

Selección de paquetes por color

La lógica de control para este caso se basa en los diagramas de espacio-fase de la figura 3.9, y es la misma que la separación de paquetes por tamaño, si consideramos que se separarán paquetes de colores, uno claro y otro oscuro.

Elemento de trabajo	Elemento de señal
A	S1
	S2
B Para paquetes claros	S3
	S4
C Para paquetes oscuros	B5
	B6



B8: Sensor de presencia tipo óptico

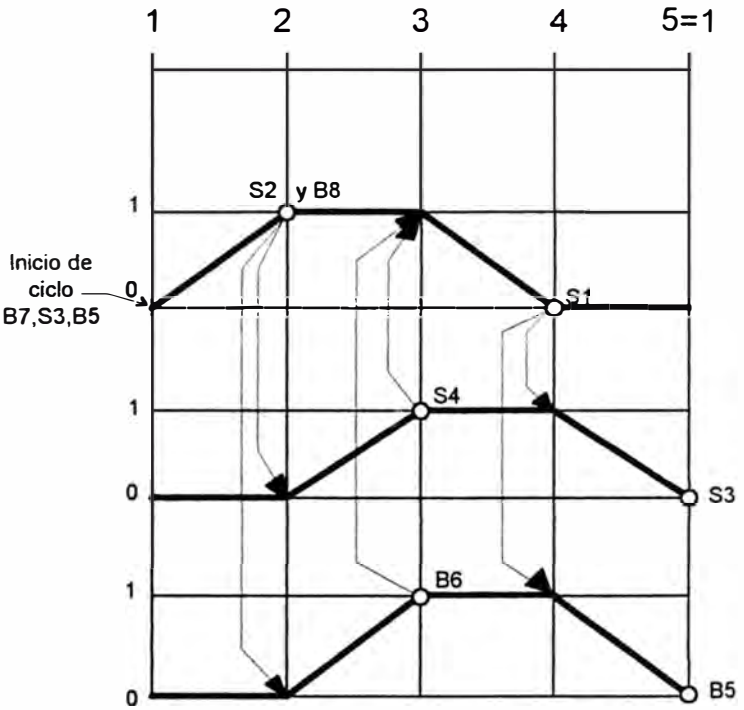
Figura 3.9

Entonces, el sensor de presencia óptico B_8, será el que permita la discriminación de color, para lo cual se empleará el cilindro B para separar paquete claro y el cilindro C para separar paquetes oscuros, ver figura 3.8.

Selección de paquete, por tipo de material

La lógica de control para separar paquetes por tipo de material (en nuestro caso, metal y no metal) se basa en los diagramas espacio-fase de la figura 3.10, y es la misma que para los dos casos anteriores, con la diferencia, que el sensor de presencia B_8 ahora será de tipo inductivo y será quien discrimine paquetes de metal y no metal, para lo cual se empleará el cilindro B para separar paquete metálico y el cilindro C para separar paquete no metálico, ver figura 3.11.

Elemento de trabajo	Elemento de señal
A	S1
	S2
B Para paquetes metálicos	S3
	S4
C Para paquetes no metálicos	B5
	B6



B8: Sensor de presencia tipo inductivo

Figura 3.10

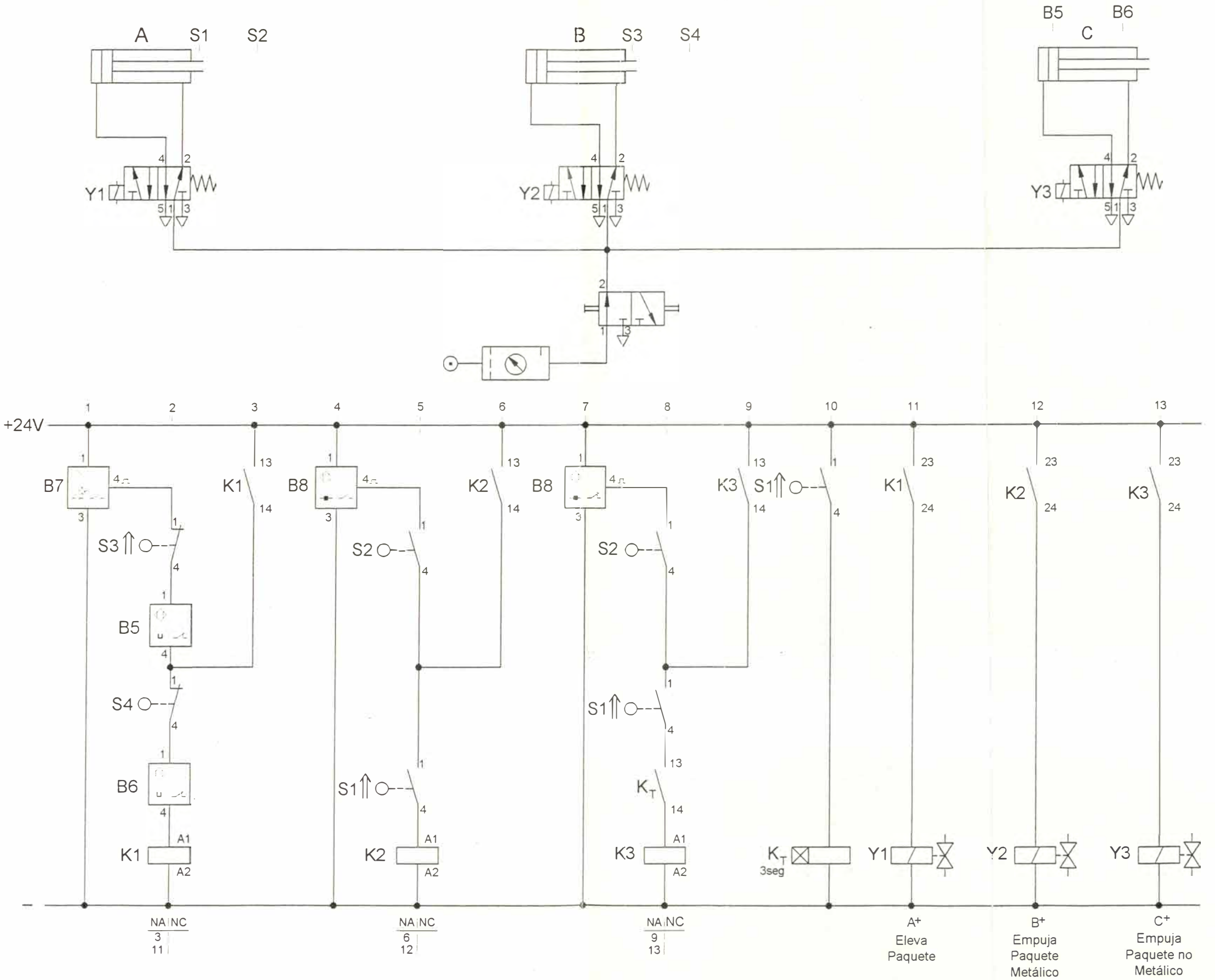
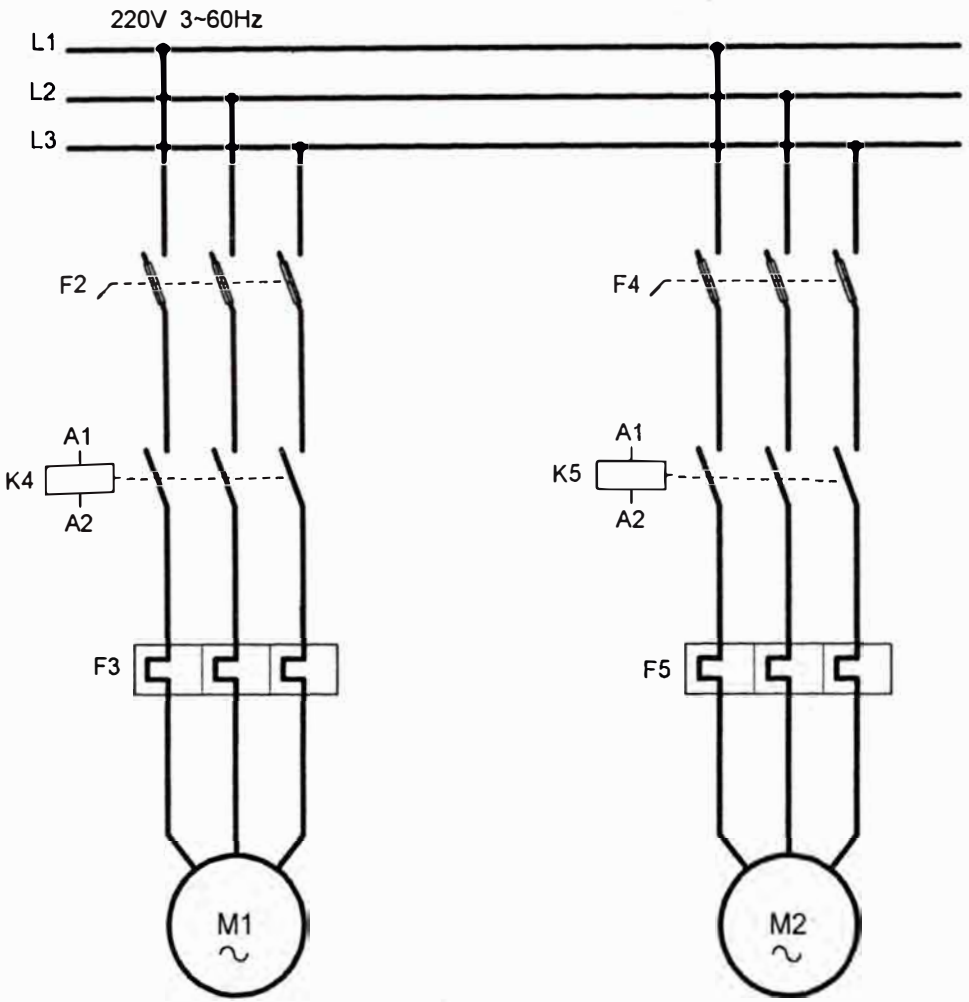
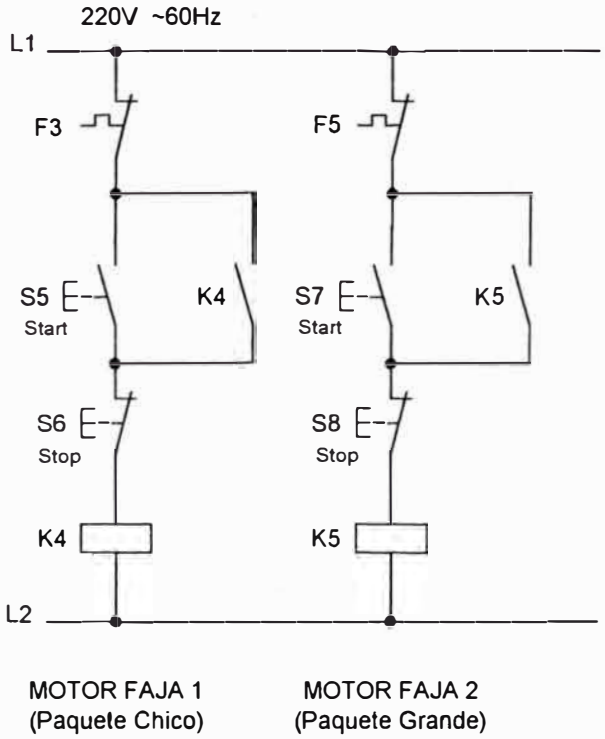


Figura 3.11: **ESQUEMAS ELECTRONEUMATICOS PARA EL ELEVADOR Y SEPARADOR DE PAQUETES POR TIPO DE MATERIAL**



a) Circuito de Fuerza



b) Circuito de Control

Figura 3.12 CIRCUITOS DE FUERZA Y CONTROL DE LAS FAJAS TRANSPORTADORAS DE PAQUETES

3.5 Programación de la lógica de control en lenguaje Assembler (de máquina)

Para la elaboración del programa de la lógica de control en lenguaje assembler se empleará el diagrama de flujo de la Figura 3.13; que obedece a la síntesis descrita en la sección 3.4.

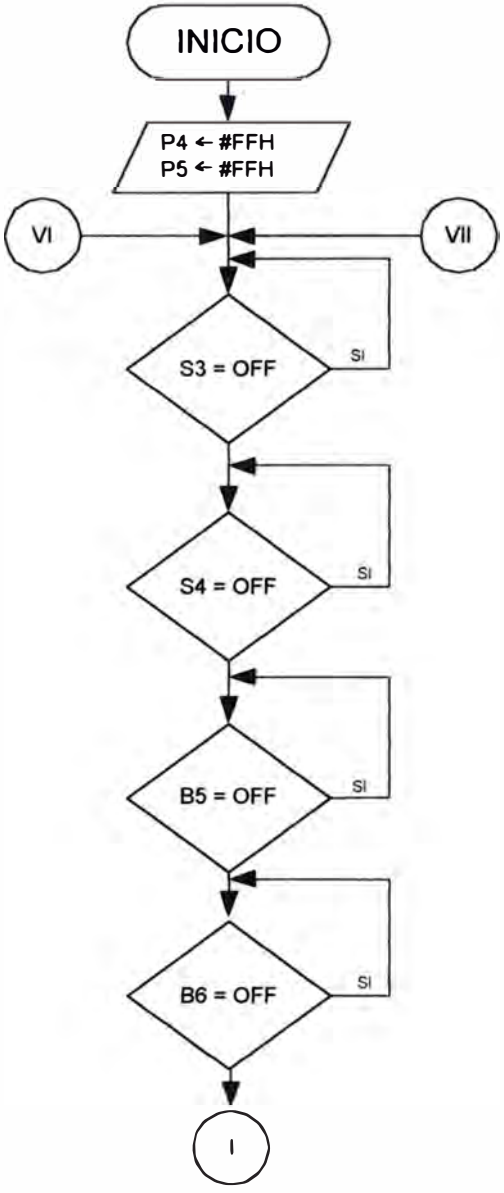


Figura 3.13 Diagrama de Flujo para la programación de la lógica de control

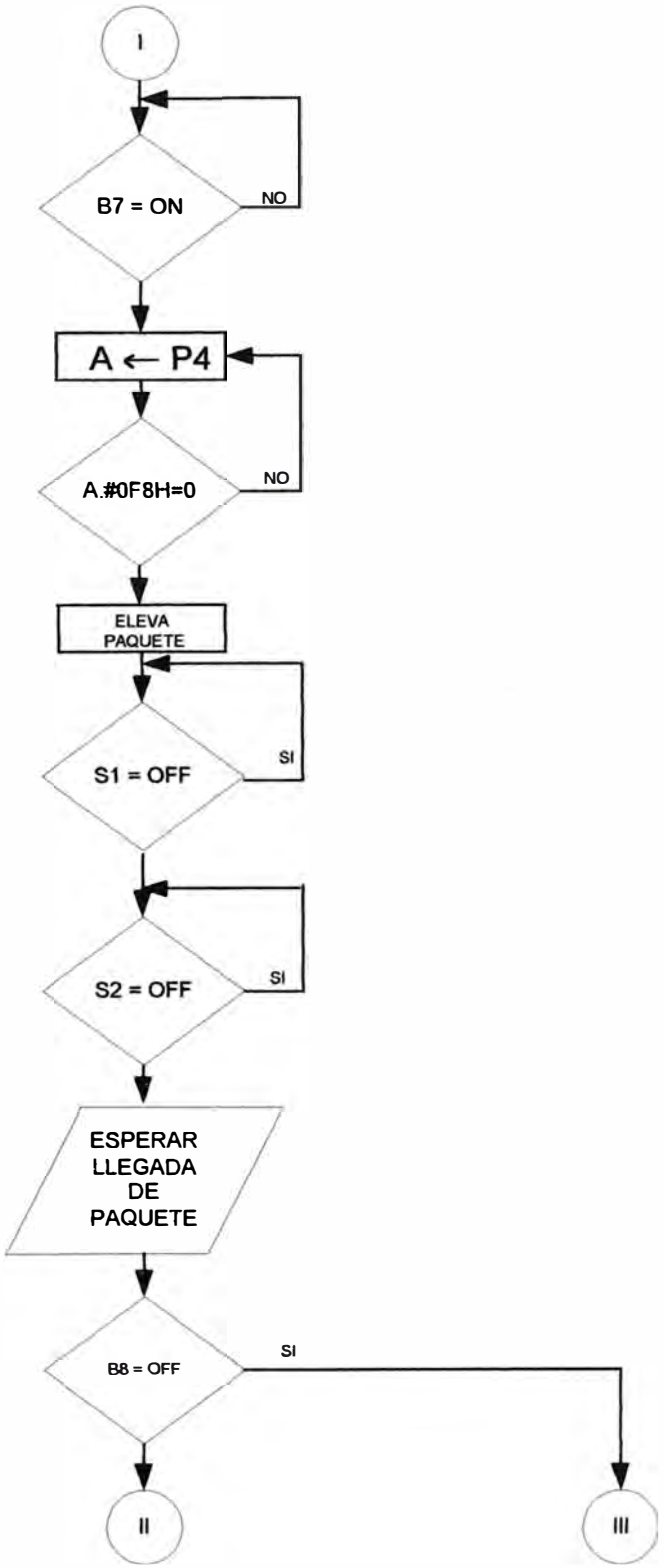


Figura 3.13 (Continuación)

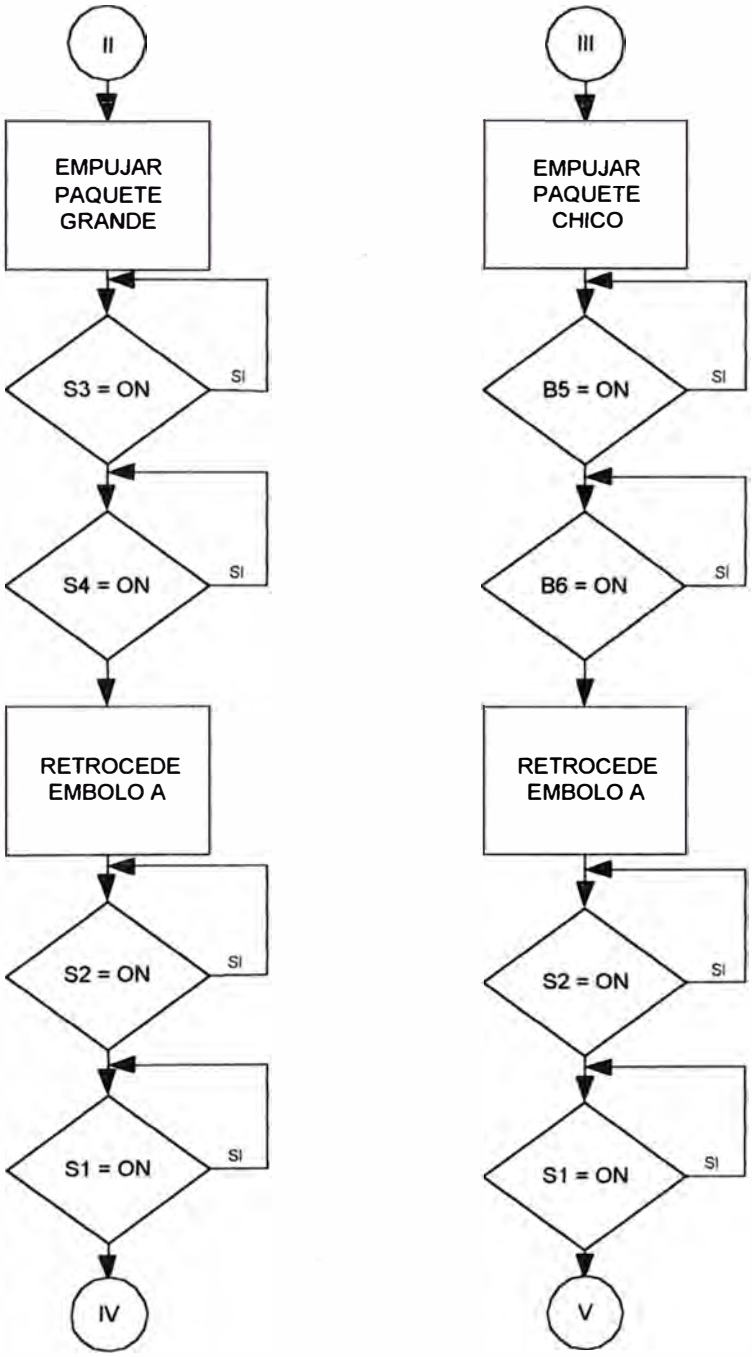


Figura 3.13 (Continuación)

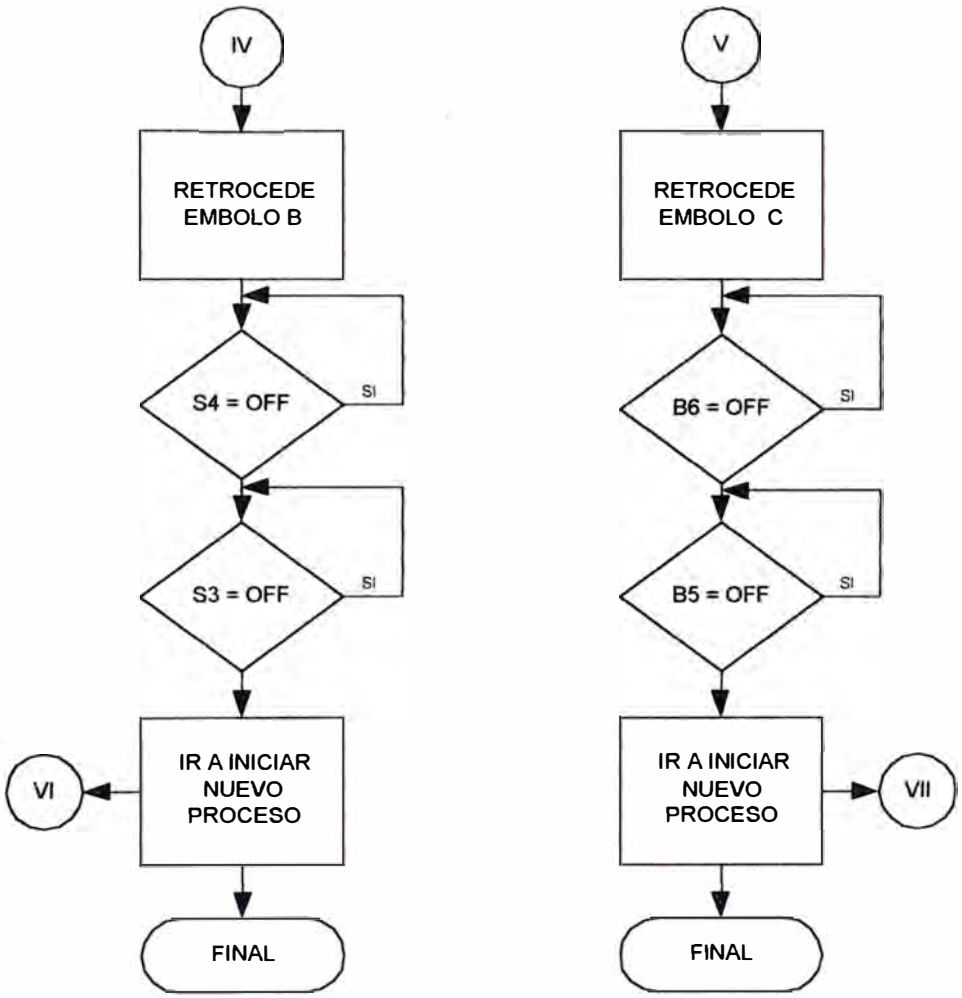


Figura 3.13 (Continuación)

```
;  PROGRAMACION DE LA LOGICA DE CONTROL EN
;
;       LENGUAJE ASSEMBLER (de máquina)

P4 EQU 0E8H
P5 EQU 0F8H
MON EQU 200H
ccLTIME EQU 021H
COMMAND EQU 030H
ACC EQU 0E0H
;
;
;
      ORG 4200H
      MOV P4,#11111111B;PUERTO DE ENTRADA
      MOV P5,#11111111B;PUERTO DE SALIDA
LABEL1  MOV A,P4
      ANL A,#0F8H
      JNZ LABEL1
      CLR P5.0;              ACTIVA CILINDRO A
LABEL2  JB P4.2,LABEL2
LABEL3  JB P4.1,LABEL3
      MOV DPTR,#3000
      MOV COMMAND,#ccLTIME
      LCALL MON
      JB P4.0,LAZO1
      CLR P5.3;              ACTIVA CILINDRO B(PAQ. GRANDE)
LABEL4  JNB P4.7,LABEL4
LABEL5  JNB P4.6,LABEL5
      SETB P5.0;             RETROCEDE CILINDRO A
LABEL6  JNB P4.1,LABEL6
LABEL7  JNB P4.2,LABEL7
      SETB P5.3;             RETROCEDE CILINDRO B
LABEL8  JB P4.6,LABEL8
LABEL9  JB P4.7,LABEL9
      SJMP LABEL1
LAZO1   CLR P5.6;              ACTIVA CILINDRO C(PAQ. CHICO)
LAZO2   JNB P4.5,LAZO2
LAZO3   JNB P4.4,LAZO3
      SETB P5.0;             RETROCEDE CILINDRO A
LAZO4   JNB P4.1,LAZO4
LAZO5   JNB P4.2,LAZO5
      SETB P5.6;             RETROCEDE CILINDRO C
LAZO6   JB P4.4,LAZO6
LAZO7   JB P4.5,LAZO7
      SJMP LABEL1
      END
```

```
; Rutina de Espera de Llegada
; de Paquete

MON     MOV R0, #2BH
SAL0    MOV R1, #42H
SAL1    MOV R2, #7FH
SAL2     DJNZ R2, SAL2
         DJNZ R1, SAL1
         DJNZ R0, SAL0
         NOP
         RET

         END
```

CAPITULO IV
COSTOS

Los costos de los componentes electrónicos que se detallan en la tabla 4.1 son aproximados. No se están considerando los costos del proceso y otros dispositivos captadores (finales de carrera, sensores de proximidad) y elementos de control final (electroválvulas). Tampoco se están considerando los costos de diseño y puesta a punto del sistema.

Tabla 4.1: costos de componentes y dispositivos para el sistema de control.

Ítem	Descripción	Cantidad	Precio Unitario (Dól. USA)	Precio Total (Dol.USA)
	Controlador			
01	Tarjeta Universal 80C535-Cpmpuboard con lass siguientes características * Microcontrolador 80C535 (Siemens) * 12 MHz * 32 KB CMSOS RAM * 32 KB EPROM (con monitor) * Interfaz RS-232 * Botón RESET * Alimentación +5V DC * Dimensiones 115x68 mm * 4 puentes (Jumper)	01 u.	220,000	220,000
	Componentes de la tarjeta de entrada			
02	Optoaislador/fototransistor CI 4 N35. I_F = 60 mA en el LED; NPN I_o = 100mA V_R = 6V	08 u	0,714	5,712
03	Transistor 2N2222, NPN-Si, AF/RF Amp. SW, I_o=0,8A, h_{FE} = 200	11 u	0,086	0,946
04	Resistencia de 0,25 W de 4,7 K Ω; 5 KΩ y 10 KΩ	21 u	0,029	0,609
05	Borneras Plásticas para 14 puntos de conexión de cable # 22 AWG	14	0,143	2,002
06	Conector para cable flat de 10 pines	01 u.	0,457	0,457

Ítém	Descripción	Cantidad	Precio Unitario (Dól. USA)	Precio Total (Dol.USA)
07	Placa para circuito impreso de 9,5 x 9 cm.	02 u	0,857	1,714
08	Acido y plumón para circuito impreso	01 u.	1,857	1,857
	Componentes de la tarjeta de salida			
09	Conector para cable flat de 10 pines	01 u.	0,457	0,457
10	Ocho unidades de arreglo Darlington CMOS/TTL Input Driver I_o máx = 500 mA CI ULN 2003	01 u.	0,429	0,429
11	Optoaislador/fototransistor CI4N37 I_F = 60 mA en el LED; NPN Io = 100mA V_R = 6V	08 u	0,714	5,712
	Transistor 2N2222, NPN-Si, af, Driver. Ic = 1,5 A; h_{FE}=100	08 u.	0,132	1,056
13	Resistencias de 0,25 w de 690 Ω y 10 Ω K	16	0,029	0,464
14	Relay 5A 30 VDC/240 V AC; IC 24 VDC	03 u.	0,714	2,142
15	Borneras de plástico para 5 puntos de conexión de cable # 22 AWG	05 u.	0,143	0,715
16	Placa para circuito impreso de 9,5 x 9 cm	02 u.	0,857	1,714
17	Acido y plumón para circuito impreso	01 u.	1,857	1,857

Ítem	Descripción	Cantidad	Precio Unitario (Dól. USA)	Precio Total (Dol.USA)
	Otros			
18	Cables # 22 AWG para conexión de tarjetas de control con el proceso	2 m.	0,429	0,858
19	Placa de acero inoxidable y pernos para fijar sensor de presencia B8 óptico	01 u.	2,286	2,286
20	Placa de fierro para fijar al sensor de presencia óptico B7	01 u.	4,286	4,286
21	Sensor de presencia óptico # K12PB8	01 u.	175,00	175,00
22	Sensor de presencia inductivo II20 VB	01 u.	64,00	64,000
			TOTAL : $	494,273

CONCLUSIONES

Se ha empleado un microcontrolador MC 80C535 para el control de un proceso de manufactura de elevación y selección de paquetes, como forma alternativa a la lógica cableada basada en relés y a la lógica programada basada en PLCs y PCs.

La lógica de control se ha programado en lenguaje Assembler o lenguaje de máquina en lugar de un lenguaje de alto nivel, como por ejemplo el lenguaje ladder, con la finalidad de minimizar el tiempo de ejecución del programa de control y espacio de memoria.

Para procesar las señales de entrada provenientes de los sensores se construyó una tarjeta de interfaz con un número específico de entradas, para el proceso en particular, a muy bajo costo. De igual manera se ha construido la tarjeta de interfaz para las señales de salida, mediante las cuales se puede energizar o desenergizar electroválvulas. Estas salidas han sido a relé por cuanto la frecuencia de conmutación, en nuestro caso, es baja.

La programación ha sido mediante una PC directamente en conexión en protocolo RS232. No se ha requerido ninguna interfaz de comunicación, como usualmente requieren la mayoría de PLCs para ser programadas por una PC, por cuanto éstas usan protocolo de comunicación basados en el RS485.

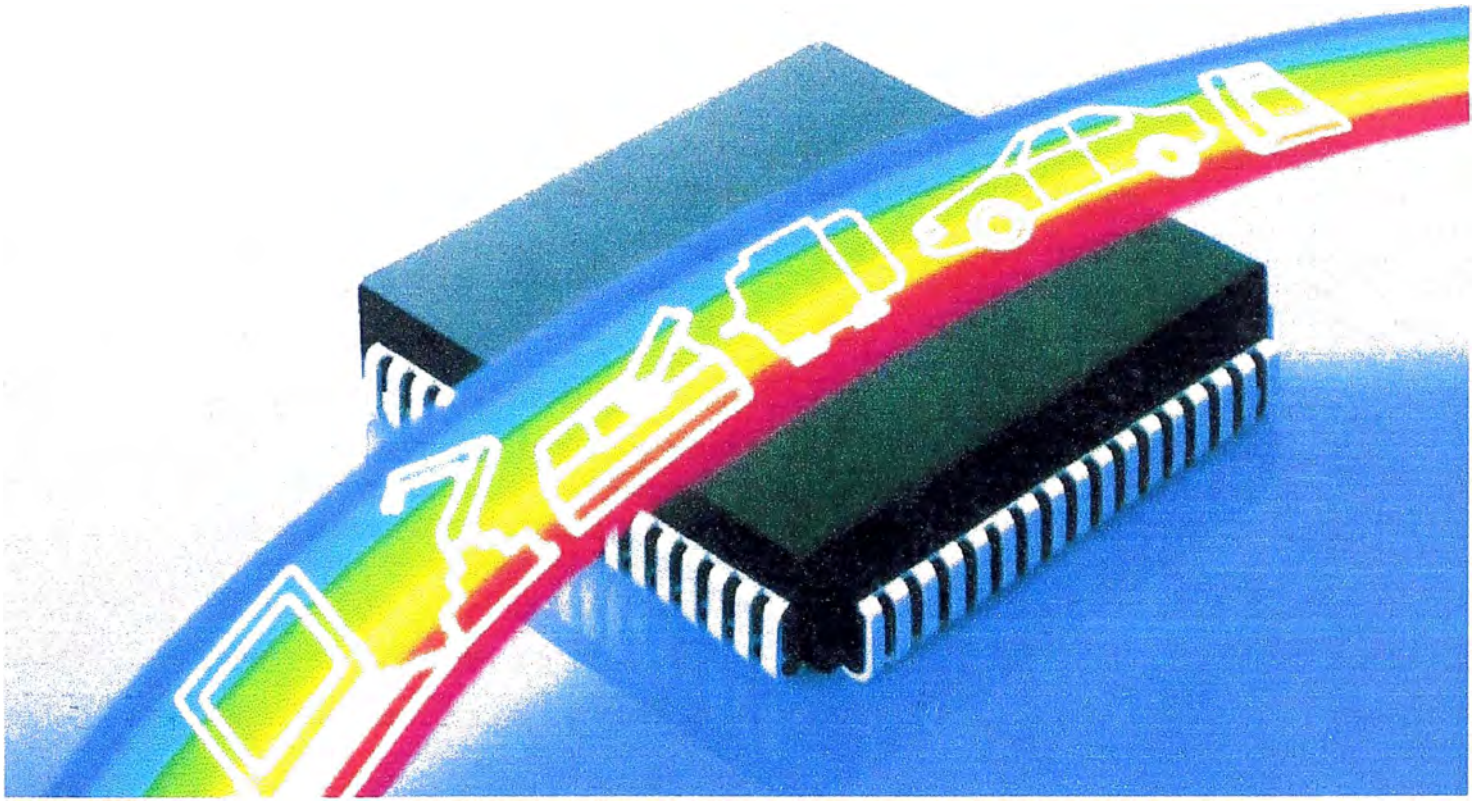
Por lo tanto, adicionalmente a las ventajas técnicas, se ha logrado ventajas económicas en el software, en las tarjetas de interfaz de entrada y de salida, y en el costo mínimo del microcontrolador. Como es sabido el software para la mayoría de marcas de PLCs en lenguaje Ladder o lista de instrucciones cuesta no menos de 400 dólares USA. Además, para un número de ocho entradas y un número de tres salidas, el PLC más pequeño (nano PLC) y su interfaz para programación cuesta no menos de 450 dólares USA. Siendo el costo de la tarjeta universal compuboard 80C535, que contiene al microcontrolador, de 220 dólares USA y siendo el software de programación y el manual de instrucciones gratuito, pues se encuentra a disposición en internet, y siendo además el costo de las tarjetas de interfaz de entradas y salidas de aproximadamente 30 dólares américanos, podemos notar entonces cláramente la ventaja económica, 250 dólares contra 850 dólares.

Por lo expuesto, se han cumplido con los objetivos de este trabajo de tesis que fueron: costo mínimo de equipamiento, consumo mínimo de energía y pérdida mínima por disipación. La comparación se ha realizado con un Nano PLC no se ha tenido en cuenta a otros dispositivos que han salido al mercado durante la realización de esta tesis, cuyos costos resultan ser menores al microcontrolador 80C535.

ANEXO A

CONJUNTO DE INSTRUCCIONES DEL MICROCONTROLADOR 80C535

SIEMENS



Microcomputer Components

SAB 80515/SAB 80C515
8-Bit Single-Chip Microcontroller Family

User's Manual 08.95

ANL <dest-byte>, <src-byte>

Function: Logical AND for byte variables

Description: ANL performs the bitwise logical AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is a accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the accumulator or immediate data.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: If the accumulator holds $0C3_H$ (11000011B) and register 0 holds $0AA_H$ (10101010B) then the instruction

ANL A,R0

will leave 81_H (10000001B) in the accumulator.

When the destination is a directly addressed byte, this instruction will clear combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the accumulator at run-time.
The instruction

ANL P1, #01110011B

will clear bits 7, 3, and 2 of output port 1.

ANL A,Rn

Operation: ANL
(A) ← (A) ∧ (Rn)

Encoding:

0 1 0 1 1 r r r

Bytes: 1

Cycles: 1

ANL A,direct

Operation: ANL
(A) ← (A) ∧ (direct)

Encoding:

0 1 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ANL A, @Ri

Operation: ANL
(A) ← (A) ∧ ((Ri))

Encoding:

0 1 0 1	0 1 1 i

Bytes: 1

Cycles: 1

ANL A, #data

Operation: ANL
(A) ← (A) ∧ #data

Encoding:

0 1 0 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

ANL direct,A

Operation: ANL
(direct) ← (direct) ∧ (A)

Encoding:

0 1 0 1	0 1 0 1	direct address

Bytes: 2

Cycles: 1

ANL direct, #data

Operation: ANL
(direct) ← (direct) ∧ #data

Encoding:

0 1 0 1	0 0 1 1	direct address	immediate data

Bytes: 3

Cycles: 2

ANL C, <src-bit>

Function: Logical AND for bit variables

Description: If the Boolean value of the source bit is a logic 0 then clear the carry flag; otherwise leave the carry flag in its current state. A slash ("/" preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, *but the source bit itself is not affected*. No other flags are affected.

Only direct bit addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P 1.0 = 1, ACC.7 = 1, and OV = 0:

```
MOV     C,P 1.0          ; Load carry with input pin state
ANL     C,ACC.7          ; AND carry with accumulator bit 7
ANL     C,/OV            ; AND with inverse of overflow flag
```

ANL C,bit

Operation: ANL
(C) ← (C) ∧ (bit)

Encoding:

1 0 0 0	0 0 1 0	bit address

Bytes: 2

Cycles: 2

ANL C,/bit

Operation: ANL
(C) ← (C) ∧ ¬ (bit)

Encoding:

1 0 1 1	0 0 0 0	bit address

Bytes: 2

Cycles: 2

CLR A

Function: Clear accumulator

Description: The accumulator is cleared (all bits set to zero). No flags are affected.

Example: The accumulator contains $5C_H$ (01011100B). The instruction

```
CLR     A
```

will leave the accumulator set to 00_H (00000000B).

Operation:
CLR
(A) ← 0

Encoding:

1 1 1 0	0 1 0 0

Bytes: 1

Cycles: 1

CLR bit

Function: Clear bit

Description: The indicated bit is cleared (reset to zero). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example: Port 1 has previously been written with $5D_H$ (01011101B). The instruction

CLR P1.2

will leave the port set to 59_H (01011001B).

CLR C

Operation: CLR
(C) ← 0

Encoding:

1 1 0 0	0 0 1 1

Bytes: 1

Cycles: 1

CLR bit

Operation: CLR
(bit) ← 0

Encoding:

1 1 0 0	0 0 1 0	bit address

Bytes: 2

Cycles: 1

DJNZ <byte>, < rel-addr>

Function: Decrement and jump if not zero

Description: DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00_H will underflow to $0FF_H$. No flags are affected. The branch destination would be computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note:

When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, not the input pins.

Example: Internal RAM locations 40_H, 50_H, and 60_H contain the values, 01_H, 70_H, and 15_H, respectively. The instruction sequence

DJNZ 40_H,LABEL_1
DJNZ 50_H,LABEL_2
DJNZ 60_H,LABEL_3

will cause a jump to the instruction at label LABEL_2 with the values 00_H, $6F_H$, and 15_H in the three RAM locations. The first jump was *not* taken because the result was zero.

This instruction provides a simple way of executing a program loop a given number of times, or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The instruction sequence

```
        MOV    R2, #8
TOGGLE: CPL    P1.7
        DJNZ   R2,TOGGLE
```

will toggle P1.7 eight times, causing four output pulses to appear at bit 7 of output port 1. Each pulse will last three machine cycles; two for DJNZ and one to alter the pin.

DJNZ Rn,rel

Operation: DJNZ
(PC) ← (PC) + 2
(Rn) ← (Rn) – 1
if (Rn) > 0 or (Rn) < 0
then (PC) ← (PC) + rel

Encoding:

1 1 0 1	1 r r r	rel. address

Bytes: 2

Cycles: 2

DJNZ direct,rel

Operation: DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) – 1
if (direct) > 0 or (direct) < 0
then (PC) ← (PC) + rel

Encoding:

1 1 0 1	0 1 0 1	direct address	rel. address

Bytes: 3

Cycles: 2

JB bit,rel

Function: Jump if bit is set

Description: If the indicated bit is a one, jump to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. The bit tested is not modified. No flags are affected.

Example: The data present at input port 1 is 11001010B. The accumulator holds 56 (01010110B). The instruction sequence

```
JB      P1.2,LABEL1
JB      ACC.2,LABEL2
```

will cause program execution to branch to the instruction at label LABEL2.

Operation:
JB
(PC)← (PC)+ 3
if (bit) = 1
then (PC)← (PC)+ rel

Encoding:

Bytes: 3

Cycles: 2

JBC bit,rel

Function: Jump if bit is set and clear bit

Description: If the indicated bit is one, branch to the address indicated; otherwise proceed with the next instruction. *In either case, clear the designated bit.* The branch destination is computed by adding the signed relative displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note:

When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, not the input pin.

Example: The accumulator holds 56_H (01010110B). The instruction sequence

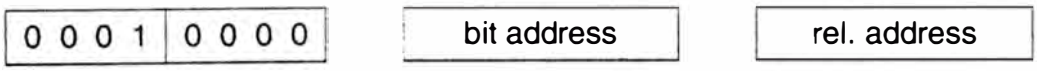
```
JBC     ACC.3,LABEL1
JBC     ACC.2,LABEL2
```

will cause program execution to continue at the instruction identified by the label LABEL2, with the accumulator modified to 52_H (01010010B).

Operation:

```
JBC
(PC) ← (PC) + 3
if (bit) = 1
then (bit) ← 0
     (PC) ← (PC) + rel
```

Encoding:



Bytes: 3

Cycles: 2

JC rel

Function: Jump if carry is set

Description: If the carry flag is set, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example: The carry flag is cleared. The instruction sequence

```
JC      LABEL1
CPL     C
JC      LABEL2
```

will set the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Operation:
JC
(PC) ← (PC) + 2
if (C) = 1
then (PC) ← (PC) + rel

Encoding:

Bytes: 2

Cycles: 2

JMP @A + DPTR

Function: Jump indirect

Description: Add the eight-bit unsigned contents of the accumulator with the sixteen-bit data pointer, and load the resulting sum to the program counter. This will be the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the accumulator nor the data pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the accumulator. The following sequence of instructions will branch to one of four AJMP instructions in a jump table starting at JMP_TBL:

```
            MOV     DPTR, #JMP_TBL
            JMP     @A + DPTR
JMP_TBL:    AJMP    LABEL0
            AJMP    LABEL1
            AJMP    LABEL2
            AJMP    LABEL3
```

If the accumulator equals 04_H when starting this sequence, execution will jump to label LABEL2. Remember that AJMP is a two-byte instruction, so the jump instructions start at every other address.

Operation: JMP
(PC) ← (A) + (DPTR)

Encoding:

0 1 1 1	0 0 1 1

Bytes: 1

Cycles: 2

JNB bit,rel

Function: Jump if bit is not set

Description: If the indicated bit is a zero, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The accumulator holds 56_H (01010110B). The instruction sequence

```
JNB     P1.3,LABEL1
JNB     ACC.3,LABEL2
```

will cause program execution to continue at the instruction at label LABEL2.

Operation:
JNB
(PC) ← (PC) + 3
if (bit) = 0
then (PC) ← (PC) + rel.

Encoding:

0 0 1 1	0 0 0 0	bit address	rel. address

Bytes: 3

Cycles: 2

JNC rel

Function: Jump if carry is not set

Description: If the carry flag is a zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example: The carry flag is set. The instruction sequence

```
JNC     LABEL1
CPL     C
JNC     LABEL2
```

will clear the carry and cause program execution to continue at the instruction identified by the label LABEL2.

Operation:
JNC
(PC) ← (PC) + 2
if (C) = 0
then (PC) ← (PC) + rel

Encoding:

0 1 0 1	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

JNZ rel

Function: Jump if accumulator is not zero

Description: If any bit of the accumulator is a one, branch to the indicated address; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The accumulator is not modified. No flags are affected.

Example: The accumulator originally holds 00_H. The instruction sequence

```
JNZ     LABEL1
INC     A
JNZ     LABEL2
```

will set the accumulator to 01_H and continue at label LABEL2.

Operation:
JNZ
(PC) ← (PC) + 2
if (A) ≠ 0
then (PC) ← (PC) + rel.

Encoding:

0 1 1 1	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

JZ rel

Function: Jump if accumulator is zero

Description: If all bits of the accumulator are zero, branch to the address indicated; otherwise proceed with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The accumulator is not modified. No flags are affected.

Example: The accumulator originally contains 01_H. The instruction sequence

```
JZ      LABEL1
DEC     A
JZ      LABEL2
```

will change the accumulator to 00_H and cause program execution to continue at the instruction identified by the label LABEL2.

Operation:
JZ
(PC) ← (PC) + 2
if (A) = 0
then (PC) ← (PC) + rel

Encoding:

0 1 1 0	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

LCALL addr16

Function: Long call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the stack pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64 Kbyte program memory address space. No flags are affected.

Example: Initially the stack pointer equals 07_H. The label "SUBRTN" is assigned to program memory location 1234_H. After executing the instruction

LCALL SUBRTN

at location 0123_H, the stack pointer will contain 09_H, internal RAM locations 08_H and 09_H will contain 26_H and 01_H, and the PC will contain 1234_H.

Operation: LCALL
(PC) ← (PC) + 3
(SP) ← (SP) + 1
((SP)) ← (PC7-0)
(SP) ← (SP) + 1
((SP)) ← (PC15-8)
(PC) ← addr15-0

Encoding:

0 0 0 1 | 0 0 1 0	addr15 . . addr8	addr7 . . addr0

Bytes: 3

Cycles: 2

MOV <dest-byte>, <src-byte>

Function: Move byte variable

Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

Example: Internal RAM location 30_H holds 40_H. The value of RAM location 40_H is 10_H. The data present at input port 1 is 11001010B ($0CA_H$).

```
MOV   R0, #30H        ; R0 < = 30H
MOV   A, @R0          ; A < = 40H
MOV   R1,A            ; R1 < = 40H
MOV   B, @R1          ; B < = 10H
MOV   @R1,P1          ; RAM (40H) < = 0CAH
MOV   P2,P1           ; P2 < = 0CAH
```

leaves the value 30_H in register 0, 40_H in both the accumulator and register 1, 10_H in register B, and $0CA_H$ (11001010B) both in RAM location 40_H and output on port 2.

MOV A,Rn

Operation: MOV
(A) ← (Rn)

Encoding:

1 1 1 0	1 r r r

Bytes: 1

Cycles: 1

MOV A,direct *)

Operation: MOV
(A) ← (direct)

Encoding:

1 1 1 0	0 1 0 1	direct address

Bytes: 2

Cycles: 1

*) MOV A,ACC is not a valid instruction.

MOV A,@Ri

Operation: MOV
(A) ← ((Ri))

Encoding:

1 1 1 0	0 1 1 i

Bytes: 1

Cycles: 1

MOV A, #data

Operation: MOV
(A) ← #data

Encoding:

0 1 1 1	0 1 0 0	immediate data

Bytes: 2

Cycles: 1

MOV Rn,A

Operation: MOV
(Rn) ← (A)

Encoding:

1 1 1 1	1 r r r

Bytes: 1

Cycles: 1

MOV Rn,direct

Operation: MOV
(Rn) ← (direct)

Encoding:

1 0 1 0	1 r r r	direct address

Bytes: 2

Cycles: 2

MOV Rn, #data

Operation: MOV
(Rn) ← #data

Encoding:

Bytes: 2

Cycles: 1

MOV direct,A

Operation: MOV
(direct) ← (A)

Encoding:

Bytes: 2

Cycles: 1

MOV direct,Rn

Operation: MOV
(direct) ← (Rn)

Encoding:

Bytes: 2

Cycles: 2

MOV direct,direct

Operation: MOV
(direct) ← (direct)

Encoding:

Bytes: 3

Cycles: 2

MOV direct, @ Ri

Operation: MOV
(direct) ← ((Ri))

Encoding:

1 0 0 0	0 1 1 i	direct address

Bytes: 2

Cycles: 2

MOV direct, #data

Operation: MOV
(direct) ← #data

Encoding:

0 1 1 1	0 1 0 1	direct address	immediate data

Bytes: 3

Cycles: 2

MOV @ Ri,A

Operation: MOV
((Ri)) ← (A)

Encoding:

1 1 1 1	0 1 1 i

Bytes: 1

Cycles: 1

MOV @ Ri,direct

Ooeration: MOV
((Ri)) ← (direct)

Encoding:

1 0 1 0	0 1 1 i	direct address

Bytes: 2

Cycles: 2

MOV @ Ri,#data

Operation: MOV
((Ri)) ← #data

Encoding:

0 1 1 1	0 1 1 i	immediate data

Bytes: 2

Cycles: 1

MOV <dest-bit>, <src-bit>

Function: Move bit data

Description: The Boolean variable indicated by the second operand is copied into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input port 3 is 11000101B. The data previously written to output port 1 is 35_H (00110101B).

```
MOV     P1.3,C
MOV     C,P3.3
MOV     P1.2,C
```

will leave the carry cleared and change port 1 to 39_H (00111001 B).

MOV C,bit

Operation: MOV
(C) ← (bit)

Encoding:

1 0 1 0	0 0 1 0	bit address

Bytes: 2

Cycles: 1

MOV bit,C

Operation: MOV
(bit) ← (C)

Encoding:

1 0 0 1	0 0 1 0	bit address

Bytes: 2

Cycles: 2

MOV DPTR, #data16

Function: Load data pointer with a 16 -bitconstant

Description: The data pointer is loaded with the 16 -bitconstant indicated. The 16 bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the low-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

Example: The instruction

MOV DPTR, #1234_H

will load the value 1234_H into the data pointer: DPH will hold 12_H and DPL will hold 34_H.

Operation: MOV
(DPTR) ← #data15-0
DPH ☐ DPL ← #data15-8 ☐ #data7-0

Encoding:

1 0 0 1	0 0 0 0	immed. data 15 . . . 8	immed. data 7 . . . 0

Bytes: 3

Cycles: 2

MOVC A, @A + <base-reg>

Function: Move code byte

Description: The MOVC instructions load the accumulator with a code byte, or constant from program memory. The address of the byte fetched is the sum of the original unsigned eight-bit accumulator contents and the contents of a sixteen-bit base register, which may be either the data pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added to the accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the accumulator. The following instructions will translate the value in the accumulator to one of four values defined by the DB (define byte) directive.

```
REL_PC: INC     A
        MOVC    A, @A + PC
        RET
        DB      66H
        DB      77H
        DB      88H
        DB      99H
```

If the subroutine is called with the accumulator equal to 01_H, it will return with 77_H in the accumulator. The INC A before the MOVC instruction is needed to "get around" the RET instruction above the table. If several bytes of code separated the MOVC from the table, the corresponding number would be added to the accumulator instead.

MOVC A, @A + DPTR

Operation: MOVC
(A) ← ((A) + (DPTR))

Encoding:

1 0 0 1	0 0 1 1

Bytes: 1

Cycles: 2

MOVC A, @A + PC

Operation: MOVC
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

Encoding:

1 0 0 0	0 0 1 1

Bytes: 1

Cycles: 2

MOVX <dest-byte>, <src-byte>

Function: Move external

Description: The MOVX instructions transfer data between the accumulator and a byte of external data memory, hence the "X" appended to MOV. There are two types of instructions, differing in whether they provide an eight bit or sixteen-bit indirect address to the external data RAM.

In the first type, the contents of R0 or R1 in the current register bank provide an eight-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or a relatively small RAM array. For somewhat larger arrays, any output port pins can be used to output higher-order address bits. These pins would be controlled by an output instruction preceding the MOVX.

In the second type of MOVX instructions, the data pointer generates a sixteen-bit address. P2 outputs the high-order eight address bits (the contents of DPH) while P0 multiplexes the low-order eight bits (DPL) with data. The P2 special function register retains its previous contents while the P2 output buffers are emitting the contents of DPH. This form is faster and more efficient when accessing very large data arrays (up to 64 Kbyte), since no additional instructions are needed to set up the output ports.

It is possible in some situations to mix the two MOVX types. A large RAM array with its high-order address lines driven by P2 can be addressed via the data pointer, or with code to output high-order address bits to P2 followed by a MOVX instruction using R0 or R1.

Example: An external 256 byte RAM using multiplexed address/data lines (e.g. an SAB 8155 RAM/I/O/timer) is connected to the SAB 80(c)5XX port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12_H and 34_H. Location 34_H of the external RAM holds the value 56_H. The instruction sequence

```
MOVX   A, @R1
MOVX   @R0,A
```

copies the value 56_H into both the accumulator and external RAM location 12_H.

MOVX A,@Ri

Operation: MOVX
(A) ← ((Ri))

Encoding:

1 1 1 0	0 0 1 i

Bytes: 1

Cycles: 2

MOVX A,@DPTR

Operation: MOVX
(A) ← ((DPTR))

Encoding:

1 1 1 0	0 0 0 0

Bytes: 1

Cycles: 2

MOVX @Ri,A

Operation: MOVX
((Ri)) ← (A)

Encoding:

1 1 1 1	0 0 1 i

Bytes: 1

Cycles: 2

MOVX @DPTR,A

Operation: MOVX
((DPTR)) ← (A)

Encoding:

1 1 1 1	0 0 0 0

Bytes: 1

Cycles: 2

NOP

Function: No operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: It is desired to produce a low-going output pulse on bit 7 of port 2 lasting exactly 5 cycles. A simple SETB/CLR sequence would generate a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming no interrupts are enabled) with the instruction sequence

```
CLR  P2.7
NOP
NOP
NOP
NOP
SETB P2.7
```

Operation: NOP

Encoding:

0 0 0 0	0 0 0 0

Bytes: 1

Cycles: 1

SETB <bit>

Function: Set bit

Description: SETB sets the indicated bit to one. SETB can operate on the carry flag or any directiy addressable bit. No other flags are affected.

Example: The carry flag is cleared. Output port 1 has been written with the value 34_H (00110100B). The instructions

```
SETB    C
SETB    P1.0
```

will leave the carry flag set to 1 and change the data output on port 1 to 35_H (00110101B).

SETB C

Operation: SETB
(C) ← 1

Encoding:

1 1 0 1	0 0 1 1

Bytes: 1

Cycles: 1

SETB bit

Operation: SETB
(bit) ← 1

Encoding:

1 1 0 1	0 0 1 0	bit address

Bytes: 2

Cycles: 1

SJMP rel

Function: Short jump

Description: Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction to 127 bytes following it.

Example: The label "RELADR" is assigned to an instruction at program memory location 0123_H. The instruction

SJMP RELADR

will assemble into location 0100_H. After the instruction is executed, the PC will contain the value 0123_H.

Note:

Under the above conditions the instruction following SJMP will be at 102_H. Therefore, the displacement byte of the instruction will be the relative offset (0123_H-0102_H) = 21_H. In other words, an SJMP with a displacement of $0FE_H$ would be a one-instruction infinite loop.

Operation: SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel

Encoding:

1 0 0 0	0 0 0 0	rel. address

Bytes: 2

Cycles: 2

Instruction Set Summary

Mnemonic		Description	Byte	Cycle
Arithmetic Operations				
ADD	A,Rn	Add register to accumulator	1	1
ADD	A,direct	Add direct byte to accumulator	2	1
ADD	A, @Ri	Add indirect RAM to accumulator	1	1
ADD	A,#data	Add immediate data to accumulator	2	1
ADDC	A,Rn	Add register to accumulator with carry flag	1	1
ADDC	A,direct	Add direct byte to A with carry flag	2	1
ADDC	A, @Ri	Add indirect RAM to A with carry flag	1	1
ADDC	A, #data	Add immediate data to A with carry flag	2	1
SUBB	A,Rn	Subtract register from A with borrow	1	1
SUBB	A,direct	Subtract direct byte from A with borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with borrow	1	1
SUBB	A,#data	Subtract immediate data from A with borrow	2	1
INC	A	Increment accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Increment indirect RAM	1	1
DEC	A	Decrement accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
INC	DPTR	Increment data pointer	1	2
MUL	AB	Multiply A and B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal adjust accumulator	1	1

Instruction Set Summary (cont'd)

Mnemonic		Description	Byte	Cycle
Logic Operations				
ANL	A,Rn	AND register to accumulator	1	1
ANL	A,direct	AND direct byte to accumulator	2	1
ANL	A,@Ri	AND indirect RAM to accumulator	1	1
ANL	A,#data	AND immediate data to accumulator	2	1
ANL	direct,A	AND accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to accumulator	1	1
ORL	A,direct	OR direct byte to accumulator	2	1
ORL	A,@Ri	OR indirect RAM to accumulator	1	1
ORL	A,#data	OR immediate data to accumulator	2	1
ORL	direct,A	OR accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive OR register to accumulator	1	1
XRL	A direct	Exclusive OR direct byte to accumulator	2	1
XRL	A,@Ri	Exclusive OR indirect RAM to accumulator	1	1
XRL	A,#data	Exclusive OR immediate data to accumulator	2	1
XRL	direct,A	Exclusive OR accumulator to direct byte	2	1
XRL	direct,#data	Exclusive OR immediate data to direct byte	3	2
CLR	A	Clear accumulator	1	1
CPL	A	Complement accumulator	1	1
RL	A	Rotate accumulator left	1	1
RLC	A	Rotate accumulator left through carry	1	1
RR	A	Rotate accumulator right	1	1
RRC	A	Rotate accumulator right through carry	1	1
SWAP	A	Swap nibbles within the accumulator	1	1

Instruction Set Summary (cont'd)

Mnemonic		Description	Byte	Cycle
Data Transfer				
MOV	A,Rn	Move register to accumulator	1	1
MOV	A,direct *)	Move direct byte to accumulator	2	1
MOV	A,@Ri	Move indirect RAM to accumulator	1	1
MOV	A,#data	Move immediate data to accumulator	2	1
MOV	Rn,A	Move accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct byte	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri, #data	Move immediate data to indirect RAM	2	1
MOV	DPTR, #data16	Load data pointer with a 16-bit constant	3	2
MOVC	A,@A + DPTR	Move code byte relative to DPTR to accumulator	1	2
MOVC	A,@A + PC	Move code byte relative to PC to accumulator	1	2
MOVX	A,@Ri	Move external RAM (8-bit addr.) to A	1	2
MOVX	A,@DPTR	Move external RAM (16-bit addr.) to A	1	2
MOVX	@Ri,A	Move A to external RAM (8-bit addr.)	1	2
MOVX	@DPTR,A	Move A to external RAM (16-bit addr.)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte from stack	2	2
XCH	A,Rn	Exchange register with accumulator	1	1
XCH	A,direct	Exchange direct byte with accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with accumulator	1	1
XCHD	A,@Ri	Exchange low-order nibble indir. RAM with A	1	1

*) MOV A,ACC is not a valid instruction

Instruction Set Summary (cont'd)

Mnemonic		Description	Byte	Cycle
Boolean Variable Manipulation				
CLR	C	Clear carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set carry flag	1	1
SETB	bit	Set direct bit	2	1
CPL	C	Complement carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to carry flag	2	2
ANL	C,/bit	AND complement of direct bit to carry	2	2
ORL	C,bit	OR direct bit to carry flag	2	2
ORL	C,/bit	OR complement of direct bit to carry	2	2
MOV	C,bit	Move direct bit to carry flag	2	1
MOV	bit,C	Move carry flag to direct bit	2	2
Program and Machine Control				
ACALL	addr11	Absolute subroutine call	2	2
LCALL	addr16	Long subroutine call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr11	Absolute jump	2	2
LJMP	addr16	Long iump	3	2
SJMP	rel	Short jump (relative addr.)	2	2
JMP	@A + DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if accumulator is zero	2	2
JNZ	rel	Jump if accumulator is not zero	2	2
JC	rel	Jump if carry flag is set	2	2
JNC	rel	Jump if carry flag is not set	2	2
JB	bit,rel	Jump if direct bit is set	3	2
JNB	bit,rel	Jump if direct bit is not set	3	2
JBC	bit,rel	Jump if direct bit is set and clear bit	3	2
CJNE	A,direct,rel	Compare direct byte to A and jump if not equal	3	2

Instruction Set Summary (cont'd)

Mnemonic	Description	Byte	Cycle
Program and Machine Control (cont'd)			
CJNE A,#data,rel	Compare immediate to A and jump if not equal	3	2
CJNE Rn,#data rel	Compare immed. to reg. and jump if not equal	3	2
CJNE @Ri,#data,rel	Compare immed. to ind. and jump if not equal	3	2
DJNZ Rn,rel	Decrement register and jump if not zero	2	2
DJNZ direct,rel	Decrement direct byte and jump if not zero	3	2
NOP	No operation	1	1

ANEXO B

INFORMACIÓN TÉCNICA DEL MICROCONTROLADOR 80C535

Pin Definitions and Functions

Symbol	Pin P-LCC-68	Pin P-MQFP-80	Input (I) Output (O)	Function
P4.0-P4.7	1-3, 5-9	72-74, 76-80	I/O	**Port 4** is an 8-bit bidirectional I/O port with internal pullup resistors. Port 4 pins that have 1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 4 pins being externally pulled low will source current (I_{IL} in the DC characteristics) because of the internal pullup resistors.
$\overline{PE}$	4	75	I	**$\overline{\text{Power saving mode enable}}$** A low level on this pin enables the use of the power saving modes (idle mode and power-down mode). When $\overline{PE}$ is held on high level it is impossible to enter the power saving modes.
$\overline{RESET}$	10	1	I	**$\overline{\text{Reset pin}}$** A low level on this pin for the duration of two machine cycles while the oscillator is running resets the SAB 80C515. A small internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
V_{AREF}	11	3		**Reference voltage** for the A/D converter
V_{AGND}	12	4		**Reference ground** for the A/D converter
P6.7-P6.0	13-20	5-12		**Port 6** is an 8-bit undirectional input port. Port pins can be used for digital input if voltage levels simultaneously meet the specifications for high/low input voltages and for the eight multiplexed analog inputs of the A/D converter.

Pin Definitions and Functions (cont'd)

Symbol	Pin P-LCC-68	Pin P-MQFP-80	Input (I) Output (O)	Function
P3.0-P3.7	21-28	15-22	I/O	**Port 3** is an 8-bit bidirectional I/O port with internal pullup resistors. Port 3 pins that have1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 3 pins being externally pulled low will source current (I_{IL}, in the DC characteristics) because of the internal pullup resistors. Port 3 also contains the interrupt, timer, serial port and external memory strobe pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate. The secondary functions are assigned to the pins of port 3, as follows: – RxD (P3.0): serial port's receiver data input (asynchronous) or data input/ output (synchronous) – TxD (P3.1): serial port's transmitter data output (asynchronous) or clock output (synchronous) – $\overline{INT0}$ (P3.2): interrupt 0 input/timer 0 gate control input – $\overline{INT1}$ (P3.3): interrupt 1 input/timer 1 gate control input – T0 (P3.4): counter 0 input – T1 (P3.5): counter 1 input – $\overline{WR}$ (P3.6): the write control signal latches the data byte from port 0 into the external data memory – $\overline{RD}$ (P3.7): the read control signal enables the external data memory to port 0

Pin Definitions and Functions (cont'd)

Symbol	Pin P-LCC-68	Pin P-MQFP-80	Input (I) Output (O)	Function
P1.7-P1.0	29-36	24-31	I/O	**Port 1** is an 8-bit bidirectional I/O port with internal pullup resistors. Port 1 pins that have 1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 1 pins being externally pulled low will source current (I_{IL} in the DC characteristics) because of the internal pullup resistors. The port is used for the low-order address byte during program verification. Port 1 also contains the interrupt, timer, clock, capture and compare pins that are used by various options. The output latch corresponding to a secondary function must be programmed to a one (1) for that function to operate (except when used for the compare functions). The secondary functions are assigned to the port 1 pins as follows: – $\overline{\text{INT3}}$/CC0 (P1.0): interrupt 3 input/ compare 0 output/capture 0 input – INT4/CC1 (P1.1): interrupt 4 input/ compare 1 output/capture 1 input – INT5/CC2 (P1.2): interrupt 5 input/ compare 2 output/capture 2 input – INT6/CC3 (P1.3): interrupt 6 input/ compare 3 output/capture 3 input – $\overline{\text{INT2}}$ (P1.4): interrupt 2 input – T2EX (P1.5): timer 2 external reload trigger input – CLKOUT (P1.6): system clock output – T2 (P1.7): counter 2 input

Pin Definitions and Functions (cont'd)

Symbol	Pin P-LCC-68	Pin P-MQFP-80	Input (I) Output (O)	Function
XTAL2 XTAL1	39 40	36 37		**XTAL2** Input to the inverting oscillator amplifier and input to the internal clock generator circuits. **XTAL1** Output of the inverting oscillator amplifier. To drive the device from an external clock source, XTAL2 should be driven, while XTAL1 is left unconnected. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is divided down by a divide-by-two flip-flop. Minimum and maximum high and low times and rise/fall times specified in the AC characteristics must be observed.
P2.0-P2.7	41-48	38-45	I/O	**Port 2** is an 8-bit bidirectional I/O port with internal pullup resistors. Port 2 pins that have 1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 2 pins being externally pulled low will source current (I_{IL} in the DC characteristics) because of the internal pullup resistors. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX@DPTR). In this application it uses strong internal pullup resistors when issuing 1's. During accesses to external data memory that use 8-bit addresses (MOVX@Ri), port 2 issues the contents of the P2 special function register.

Pin Definitions and Functions (cont'd)

Symbol	Pin P-LCC-68	Pin P-MQFP-80	Input (I) Output (O)	Function
$\overline{\text{PSEN}}$	49	47	O	The **Program store enable** output is a control signal that enables the external program memory to the bus during external fetch operations. It is activated every six oscillator periods, except during external data memory accesses. The signal remains high during internal program execution.
ALE	50	48	O	The **Address latch enable** output is used for latching the address into external memory during normal operation. It is activated every six oscillator periods, except during an external data memory access.
$\overline{\text{EA}}$	51	49	I	**External access enable** When held high, the SAB 80C515 executes instructions from the internal ROM as long as the PC is less than 8192. When held low, the SAB 80C515 fetches all instructions from external program memory. For the SAB 80C535 this pin must be tied low.
P0.0-P0.7	52-59	52-59	I/O	**Port 0** is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application it uses strong internal pullup resistors when issuing 1's. Port 0 also outputs the code bytes during program verification in the SAB 80C515. External pullup resistors are required during program verification.

Pin Definitions and Functions (cont'd)

Symbol	Pin P-LCC-68	Pin P-MQFP-80	Input (I) Output (O)	Function
P5.7-P5.0	60-67	60-67	I/O	**Port 5** is an 8-bit bidirectional I/O port with internal pullup resistors. Port 5 pins that have 1's written to them are pulled high by the internal pullup resistors, and in that state can be used as inputs. As inputs, port 5 pins being externally pulled low will source current (I_{IL} in the DC characteristics) because of the internal pullup resistors.
V_{CC}	37	33	–	**Supply voltage** during normal, idle, and power-down operation. Internally connected to pin 68.
V_{SS}	38	34	–	**Ground** (0 V)
V_{CC}	68	69	–	**Supply voltage** during normal, idle, and power-down operation. Internally connected to pin 37.
N. C.	–	2, 13, 14, 23, 32, 35, 46, 50, 51, 68, 70, 71	–	**Not connected** These pins of the P-MQFP-80 package must not be connected

Table 2: Special Function Registers - Functional Blocks

Block	Symbol	Name	Address	Contents after Reset
CPU	ACC	Accumulator	**$0E0_H$** [1]	00_H
	B	B-Register	**$0F0_H$** [1]	00_H
	DPH	Data Pointer, High Byte	83_H	00_H
	DPL	Data Pointer, Low Byte	82_H	00_H
	PSW	Program Status Word Register	**$0D0_H$** [1]	00_H
	SP	Stack Pointer	81_H	07_H
A/D-Converter	ADCON	A/D Converter Control Register	**$0D8_H$** [1]	$00X0\ 0000_B$ [2]
	ADDAT	A/D Converter Data Register	$0D9_H$	00_H
	DAPR	D/A Converter Program Register	$0DA_H$	00_H
Interrupt System	EN0	Interrupt Enable Register 0	**$0A8_H$** [1]	00_H
	IEN1	Interrupt Enable Register 1	**$0B8_H$** [1]	00_H
	IP0	Interrupt Priority Register 0	$0A9_H$	00_H
	IP1	Interrupt Priority Register 1	$0B9_H$	$X000\ 0000_B$ [2]
	IRCON	Interrupt Request Control Register	**$0C0_H$** [1]	$XX00\ 0000_B$ [3]
	TCON [2]	Timer Control Register	**88_H** [1]	00_H
	T2CON [2]	Timer 2 Control Register	**$0C8_H$** [1]	00_H
				00_H
Compare/ Capture-Unit (CCU)	CCEN	Comp./Capture Enable Reg.	$0C1_H$	00_H
	CCH1	Comp./Capture Reg. 1, High Byte	$0C3_H$	00_H
	CCH2	Comp./Capture Reg. 2, High Byte	$0C5_H$	00_H
	CCH3	Comp./Capture Reg. 3, High Byte	$0C7_H$	00_H
	CCL1	Comp./Capture Reg. 1, Low Byte	$0C2_H$	00_H
	CCL2	Comp./Capture Reg. 2, Low Byte	$0C4_H$	00_H
	CCL3	Comp./Capture Reg. 3, Low Byte	$0C6_H$	00_H
	CRCH	Com./Rel./Capt. Reg. High Byte	$0CB_H$	00_H
	CRCL	Com./Rel./Capt. Reg. Low Byte	$0CA_H$	00_H
	TH2	Timer 2, High Byte	$0CD_H$	00_H
	TL2	Timer 2, Low Byte	$0CC_H$	00_H
	T2CON	Timer 2 Control Register	**$0C8_H$** [1]	00_H

[1] Bit-addressable special function registers

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks.

[3] X means that the value is indeterminate and the location is reserved

Table 2: Special Function Registers- Functional Blocks (cont'd)

Block	Symbol	Name	Address	Contents after Reset
Ports	P0	Port 0	**80_H** [1]	$0FF_H$
	P1	Port 1	**90_H** [1]	$0FF_H$
	P2	Port 2	**$0A0_H$** [1]	$0FF_H$
	P3	Port 3	**$0B0_H$** [1]	$0FF_H$
	P4	Port 4	**$0E8_H$** [1]	$0FF_H$
	P5	Port 5	**$0F8_H$** [1]	$0FF_H$
	P6	Port 6, Analog/Digital Input	$0DB_H$	
Pow.Sav.Modes	PCON	Power Control Register	87_H	$000X\ 0000_B$ [2]
Serial Channels	ADCON [2]	A/D Converter Control Reg.	**$0D8_H$** [1]	$00X0\ 0000_B$ [2]
	PCON [2]	Power Control Register	87_H	$000X\ 0000_B$ [2]
	SBUF	Serial Channel Buffer Reg.	99_H	$0XX_H$ [3]
	SCON	Serial Channel Control Reg.	**98_H** [1]	00_H
Timer 0/ Timer 1	TCON	Timer Control Register	**88_H** [1]	00_H
	TH0	Timer 0, High Byte	$8C_H$	00_H
	TH1	Timer 1, High Byte	$8D_H$	00_H
	TL0	Timer 0, Low Byte	$8A_H$	00_H
	TL1	Timer 1, Low Byte	$8B_H$	00_H
	TMOD	Timer Mode Register	89_H	00_H
Watchdog	IEN0 [2]	Interrupt Enable Register 0	**$0A8_H$** [1]	00_H
	IEN1 [2]	Interrupt Enable Register 1	**$0B8_H$** [1]	00_H
	IP0 [2]	Interrupt Priority Register 0	$0A9_H$	$X000\ 0000_B$ [2]
	IP1 [2]	Interrupt Priority Register 1	$0B9_H$	$XX00\ 0000_B$ [3]

[1] Bit-addressable special function registers

[2] This special function register is listed repeatedly since some bits of it also belong to other functional blocks.

[3] X means that the value is indeterminate and the location is reserved

I/O Ports

The SAB 80C515 has six 8-bit I/O ports and one 8-bit input port. Port 0 is an open-drain bidirectional I/O port, while ports 1 to 5 are quasi-bidirectional I/O ports with internal pullup resistors. That means, when configured as inputs, ports 1 to 5 will be pulled high and will source current when externally pulled low. Port 0 will float when configured as input.

Port 0 and port 2 can be used to expand the program and data memory externally. During an access to external memory, port 0 emits the low-order address byte and reads/writes the data byte, while port 2 emits the high-order address byte. In this function, port 0 is not an open-drain port, but uses a strong internal pullup FET. Ports 1 and 3 are provided for several alternate functions, as listed below:

Port	Symbol	Function
P1.0	$\overline{\text{INT3}}$/CC0	External interrupt 3 input, compare 0 output, capture 0 input
P1.1	INT4/CC1	External interrupt 4 input, compare 1 output, capture 1 input
P1.2	INT5/CC2	External interrupt 5 input, compare 2 output, capture 2 input
P1.3	INT6/CC3	External interrupt 6 input, compare 3 output, capture 3 input
P1.4	$\overline{\text{INT2}}$	External interrupt 2 input
P1.5	T2EX	Timer 2 external reload trigger input
P1.6	CLKOUT	System clock output
P1.7	T2	Timer 2 external count or gate input
P3.0	RxD	Serial port's receiver data input (asynchronous) or data input/output (synchronous)
P3.1	TxD	Serial port's transmitter data output (asynchronous) or clock output (synchronous)
P3.2	$\overline{\text{INT0}}$	External interrupt 0 input, timer 0 gate control
P3.3	$\overline{\text{INT1}}$	External interrupt 1 input, timer 1 gate control
P3.4	T0	Timer 0 external counter input
P3.5	T1	Timer 1 external counter input
P3.6	$\overline{\text{WR}}$	External data memory write strobe
P3.7	$\overline{\text{RD}}$	External data memory read strobe

The SAB 80C515 has dual-purpose input port. As the ANx lines in the SAB 80515 (NMOS version), the eight port lines at port 6 can be used as analog inputs. But if the input voltages at port 6 meet the specified digital input levels (V_{IL} an d V_{IH}), the port can also be used as digital input port. Reading the special function register P6 allows the user to input the digital values currently applied to the port pins. It is not necessary to select these modes by software; the voltages applied at port 6 pins can be converted to digital values using the A/D converter and at the same time the pins can be read via SFR P6.
It must be noted, however, that the results in port P6 bits will be indeterminate if the levels at the corresponding pins are not within their respective V_{IL}/ V_{IH} specifications. Furthermore, it is not possible to use port P6 as output lines. Special function register P6 is located at address $0DB_H$.

Serial Port

The serial port of the SAB 80C515 enables full duplex communication between microcontrollers or between microcontroller and peripheral devices.
The serial port can operate in 4 modes:

Mode 0: Shift register mode. Serial data enters and exits through RxD. TxD outputs the shift clock. 8-bits are transmitted/received: 8 data bits (LSB first). The baud rate is fixed at 1/12 of the oscillator frequency.

Mode 1: 10-bits are transmitted (through RxD) or received (through TxD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). The baud rate is variable.

Mode 2: 11-bits are transmitted (through RxD) or received (through TxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency.

Mode 3: 11-bits are transmitted (through TxD) or received (through RxD): a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). Mode 3 is identical to mode 2 except for the baud rate. The baud rate in mode 3 is variable.

The variable baud rates in modes 1 and 3 can be generated by timer 1 or an internal baud rate generator.

A/D Converter

The 8-bit A/D converter of the SAB 80C515 has eight multiplexed analog inputs (Port 6) and uses the successive approximation method.
There are three characteristic time frames in a conversion cycle (see A/D converter characteristics): the conversion time t_C, which is the time required for one conversion; the sample time t_S which is included in the conversion time and is measured from the start of the conversion; the load time t_L, which in turn is part of the sample time and also is measured from the conversion start.
Within the load time t_L, the analog input capacitance C_I must be loaded to the analog inpult voltage level. For the rest of the sample time t_S, after the load time has passed, the selected analog input must be held constant. During the rest of the conversion time t_C the conversion itself is actually performed. Conversion can be programmed to be single or continuous; at the end of a conversion an interrupt can be generated.

A unique feature is the capability of internal reference voltage programming. The internal reference voltages $V_{IntAREF}$ and $V_{IntAGND}$ for the A/D converter both are programmable to one of 16 steps with respect to the external reference voltages. This feature permits a conversion with a smaller internal reference voltage range to gain a higher resolution.
In addition, the internal reference voltages can easily be adapted by software to the desired analog input voltage range.

Figure 4 shows a block diagram of the A/D converter.

Watchdog Timer

This feature is provided as a means of graceful recovery from a software upset. After an external reset, the watchdog timer is cleared and stopped. It can be started and cleared by software, but it cannot be stopped during active mode of the device. If the software fails to clear the watchdog timer at least every 65532 machine cycles (about 65 ms if a 12 MHz oscillator frequency is used), an internal reset will be initiated. The reset cause (external reset or reset caused by the watchdog) can be examined by software. To clear the watchdog, two bits in two different special function registers must be set by two consecutive instructions (bits IEN0.6 and IEN1.6). This is done to prevent the watchdog from being cleared by unexpected opcodes.

It must be noted, however, that the watchdog timer is halted during the idle mode and power-down mode of the processor (see section "Power Saving Modes" below).
Therefore, it is possible to use the idle mode in combination with the watchdog timer function. But even the watchdog timer cannot reset the device when one of the power saving modes has been is entered accidentally.
For these reasons several precautions are taken against unintentional entering of the power-down or idle mode (see below).

Power Saving Modes

The ACMOS technology of the SAB 80C515 allows two new power saving modes of the device: The idle mode and the power-down mode. These modes replace the power-down supply mode via pin V_{PD} of the SAB 80515 (NMOS). The SAB 80C515 is supplied via pins V_{CC} also during idle and power-down operation.
However, there are applications where unintentional entering of these power saving modes must be absolutely avoided. Such critical applications often use the watchdog timer to prevent the system from program upsets. Then accidental entering of the power saving modes would even stop the watchdog timer and would circumvent the watchdog timer's task of system protection.

Thus, the SAB 80C515 has an extra pin that allows it to disable both of the power saving modes. When pin $\overline{PE}$ is held high, idle mode and power-down mode are completely disabled and the instruction sequences that are used for entering these modes (see below) will NOT affect the normal operations of the device. When $\overline{PE}$ is held low, the use of the idle mode and power-down mode is possible as described in the following sections.

Pin $\overline{PE}$ has a weak internal pullup resistor. Thus, when left open, the power saving modes are disabled.

The Special Function Register PCON

In the NMOS version SAB 80515 the SFR PCON (address 87_H) contains only bit SMOD; in the CMOS version SAB 80C515 there are more bits used (see table 4).
The bits PDE, PDS and IDLE, IDLS select the power-down mode or the idle mode, respectively, when the use of the power saving modes is enabled by pin $\overline{PE}$ (see next page).

If the power-down mode and the idle mode are set at the same time, power-down takes precedence.
Furthermore, register PCON contains two general purpose flags. For example, the flag bits GF0 and GF1 can be used to give an indication if an interrupt occurred during normal operation or during an idle. Then an instruction that activates Idle can also set one or both flag bits. When idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.
The reset value of PCON is $000X0000_B$.

Table 4
SFR PCON (87_H)

SMOD	PDS	IDLS	–	GF1	GF0	PDE	IDLE	87_H
7	6	5	4	3	2	1	0	

Symbol	Position	Function
SMOD	PCON.7	When set, the baud rate of the serial channel in mode 1, 2, 3 is doubled.
PDS	PCON.6	Power-down start bit. The instruction that sets the PDS flag bit is the last instruction before entering the power-down mode.
IDLS	PCON.5	Idle start bit. The instruction that sets the IDLS flag bit is the last instruction before entering the idle mode.
–	PCON.4	Reserved
GF1	PCON.3	General purpose flag
GF0	PCON.2	General purpose flag
PDE	PCON.1	Power-down enable bit. When set, starting of the power-down mode is enabled.
IDLE	PCON.0	Idle mode enable bit. When set, starting of the idle mode is enabled.

Idle Mode

In the idle mode the oscillator of the SAB 80C515 continues to run, but the CPU is gated off from the clock signal. However, the interrupt system, the serial port, the A/D converter, and all timers with the exception of the watchdog timer are further provided with the clock. The CPU status is preserved in its entirety: the stack pointer, program counter, program status word, accumulator, and all other registers maintain their data during idle mode.

The reduction of power consumption, which can be achieved by this feature depends on the number of peripherals running.

If all timers are stopped and the A/D converter and the serial interface are not running, the maximum power reduction can be achieved. This state is also the test condition for the idle mode I_{CC} (see DC characteristics, note 5).

So the user has to take care which peripheral should continue to run and which has to be stopped during idle mode. Also the state of all port pins – either the pins controlled by their latches or controlled by their secondary functions – depends on the status of the controller when entering idle mode.

Normally the port pins hold the logical state they had at the time idle mode was activated. If some pins are programmed to serve their alternate functions they still continue to output during idle mode if the assigned function is on. This applies to the compare outputs as well as to the clock output signal or to the serial interface in case it cannot finish reception or transmission during normal operation. The control signals ALE and $\overline{\text{PSEN}}$ hold at logic high levels (see table 5).

Table 5
Status of External Pins During Idle and Power-Down Mode

	Last instruction executed from internal code memory		**Last instruction executed from external code memory**	
Outputs	**Idle**	**Power-down**	**Idle**	**Power-down**
ALE	High	Low	High	Low
PSEN	High	Low	High	Low
PORT 0	Data	Data	Float	Float
PORT 1	Data/alternate outputs	Data/last output	Data/alternate outputs	Data/last output
PORT 2	Data	Data	Address	Data
PORT 3	Data/alternate outputs	Data/last output	Data/alternate outputs	Data/last output
PORT 4	Data	Data	Data	Data
PORT 5	Data	Data	Data	Data

As in normal operation mode, the ports can be used as inputs during idle mode. Thus a capture or reload operation can be triggered, the timers can be used to count external events, and external interrupts will be detected.

The idle mode is a useful feature which makes it possible to "freeze" the processor's status – either for a predefined time, or until an external event reverts the controller to normal operation, as discussed below. The watchdog timer is the only peripheral which is automatically stopped during idle mode. If it were not disabled on entering idle mode, the watchdog timer would reset the controller, thus abandoning the idle mode.

When idle mode is used, pin $\overline{PE}$ must be held on low level. The idle mode is then entered by two consecutive instructions. The first instruction sets the flag bit IDLE (PCON.0) and must not set bit IDLS (PCON.5), the following instruction sets the start bit IDLS (PCON.5) and must not set bit IDLE (PCON.0). The hardware ensures that a concurrent setting of both bits, IDLE and IDLS, does not initiate the idle mode. Bits IDLE and IDLS will automatically be cleared after being set. If one of these register bits is read the value that appears is 0 (see table 4). This double instruction is implemented to minimize the chance of an unintentional entering of the idle mode which would leave the watchdog timer's task of system protection without effect.

Note that PCON is not a bit-addressable register, so the above mentioned sequence for entering the idle mode is obtained by byte-handling instructions, as shown in the following example:

```
ORL     PCON,#00000001B     ;Set bit IDLE, bit IDLS must not be set
ORL     PCON,#00100000B     ;Set bit IDLS, bit IDLE must not be set
```

The instruction that sets bit IDLS is the last instruction executed before going into idle mode.

There are two ways to terminate the idle mode:

- The idle mode can be terminated by activating any enable interrupt. This interrupt will be serviced and normally the instruction to be executed following the RETI instruction will be the one following the instruction that sets the bit IDLS.

- The other way to terminate the idle mode, is a hardware reset. Since the oscillator is still running, the hardware reset must be held active only for two machine cycles for a complete reset.

Power-Down Mode

In the power-down mode, the on-chip oscillator is stopped. Therefore all functions are stopped; only the contents of the on-chip RAM and the SFR's are maintained.The port pins controlled by their port latches output the values that are held by their SFR's.
The port pins which serve the alternate output functions show the values they had at the end of the last cycle of the instruction which initiated the power-down mode; when the clockout signal (CLKOUT, P1.6) is enabled, it will stop at low level. ALE and $\overline{PSEN}$ hold at logic low level (see table 5).

To enter the power-down mode the pin $\overline{PE}$ must be on low level. The power-down mode then is entered by two consecutive instructions. The first instruction has to set the flag bit PDE (PCON.1) and must not set bit PDS (PCON.6), the following instruction has to set the start bit PDS (PCON.6) and must not set bit PDE (PCON.1). The hardware ensures that a concurrent setting of both bits, PDE and PDS, does not initiate the power-down mode. Bits PDE and PDS will automatically be cleared after having been set and the value shown by reading one of these bits is always 0 (see table 4). This double instruction is implemented to minimize the chance of unintentionally entering the power-down mode which could possibly "freeze" the chip's activity in an undesired status.

Absolute Maximum Ratings

Ambient temperature under bias	
SAB 80C515	0 to 70 °C
SAB 80C515-T3	– 40 to 85 °C
Storage temperature	– 65 to 150 °C
Voltage on V_{CC} pins with respect to ground (V_{SS})	– 0.5 to 6.5 V
Voltage on any pin with respect to ground (V_{SS})	– 0.5 to V_{CC} + 0.5 V
Input current on any pin during overload condition	– 10 mA to + 10 mA
Absolute sum of all input currents during overload condition	|100 mA|
Power disipation	2 W

Note *Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage of the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for longer periods may affect device reliability. During overload conditions ($V_{IN} > V_{CC}$ or $V_{IN} < V_{SS}$) the Voltage on V_{CC} pins with respect to ground (V_{SS}) must not exeed the values defined by the absolute maximum ratings.*

DC Characteristics

$V_{CC} = 5\ V \pm 10\ \%$; $V_{SS} = 0\ V$

$T_A = 0$ to 70 °C for the SAB 80C515/80C535
$T_A = -40$ to 85 °C for the SAB 80C515/80C535-T3

Parameter	Symbol	Limit values		Unit	Test condition
		min.	max.		
Input low voltage (except $\overline{EA}$)	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input low voltage ($\overline{EA}$)	V_{IL1}	– 0.5	0.2 V_{CC} – 0.3	V	–
Input high voltage (except $\overline{RESET}$ and XTAL[2)]	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage to XTAL2	V_{IH1}	0.7 V_{CC}	V_{CC} + 0.5	V	–
Input high voltage to $\overline{RESET}$	V_{IH2}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage, ports 1, 2, 3, 4, 5	V_{OL}	–	– 0.45	V	$I_{OL} = 1.6$ mA [1)]

Notes see page 251.

DC Characteristics (cont'd)

Parameter	Symbol	Limit values		Unit	Test condition
		min.	max.		
Output low voltage, port 0, ALE, $\overline{PSEN}$	V_{OL1}	–	0.45	V	I_{OL} = 3.2 mA 1)
Output high voltage, ports 1, 2, 3, 4, 5	V_{OH}	2.4 0.9 V_{CC}	– –	V V	I_{OH} = – 80 µA I_{OH} = – 10 µA
Output high voltage (port 0 in external bus mode, ALE, PSEN)	V_{OH1}	2.4 0.9 V_{CC}	– –	V V	I_{OH} = – 400 µA I_{OH} = – 40 µA 2)
Logic 0 input current, ports 1, 2, 3, 4, 5	I_{IL}	– 10	– 70	µA	V_{IN} = 0.45 V
Input low current to $\overline{RESET}$ for reset	I_{IL2}	– 10	– 100	µA	V_{IN} = 0.45 V
Input low current ($\overline{XTAL2}$)	I_{IL3}	–	– 15	µA	V_{IN} = 0.45 V
Input low current ($\overline{PE}$)	I_{IL4}	–	– 20	µA	V_{IN} = 0.45 V
Logical 1-to-0 transition current, ports 1, 2, 3, 4, 5	I_{TL}	– 65	– 650	µA	V_{IN} = 2 V
Input leakage current (port 0, port 6, AN0-7, $\overline{EA}$)	I_{LI}	–	± 1	µA	0.45 < V_{IN} < V_{CC}
Pin capacitance	C_{IO}	–	10	pF	f_C = 1 MHz, T_A = 25 °C
Power-supply current: Active mode, 12 MHz 6) Idle mode, 12 MHz 6) Active mode, 16 MHz 6) Idle mode, 16 MHz 6) Power-down mode	 – I_{CC} – I_{CC} – I_{CC} – I_{CC} – I_{PD}	 – – – – –	 35 13 46 17 50	 mA mA mA mA µA	 V_{CC} = 5 V 4) V_{CC} = 5 V 5) V_{CC} = 5 V 4) V_{CC} = 5 V 5) V_{CC} = 2 V to 5.5 V 3)

Notes see page 251.

AC Characteristics

V_{CC} = 5 V ± 10%; V_{SS} = 0 V (C_L for Port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all outputs = 80 pF); T_A = 0 to 70 °C for SAB 80C515/80C535
T_A = – 40 to 85 °C for SAB 80C515/80C535-T40/85

Parameter	Symbol	Limit values				Unit
		12 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	

Program Memory Characteristics

ALE pulse width	t_{LHLL}	127	–	$2\,t_{CLCL} - 40$	–	ns
Address setup to ALE	t_{AVLL}	53	–	$t_{CLCL} - 30$	–	ns
Address hold after ALE	t_{LLAX}	48	–	$t_{CLCL} - 35$	–	ns
ALE to valid instruction in	t_{LLIV}	–	233	–	$4\,t_{CLCL} - 100$	ns
ALE to $\overline{PSEN}$	t_{LLPL}	58	–	$t_{CLCL} - 25$	–	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	215		$3\,t_{CLCL} - 35$		ns
$\overline{PSEN}$ to valid instruction in	t_{PLIV}	–	150	–	$3\,t_{CLCL} - 100$	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ} [1]	–	63	–	$t_{CLCL} - 20$	ns
Address valid after $\overline{PSEN}$	t_{PXAV} [1]	75		$t_{CLCL} - 8$		ns
Address to valid instruction in	t_{AVIV}	–	302	–	$5\,t_{CLCL} - 115$	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	–	0	–	ns

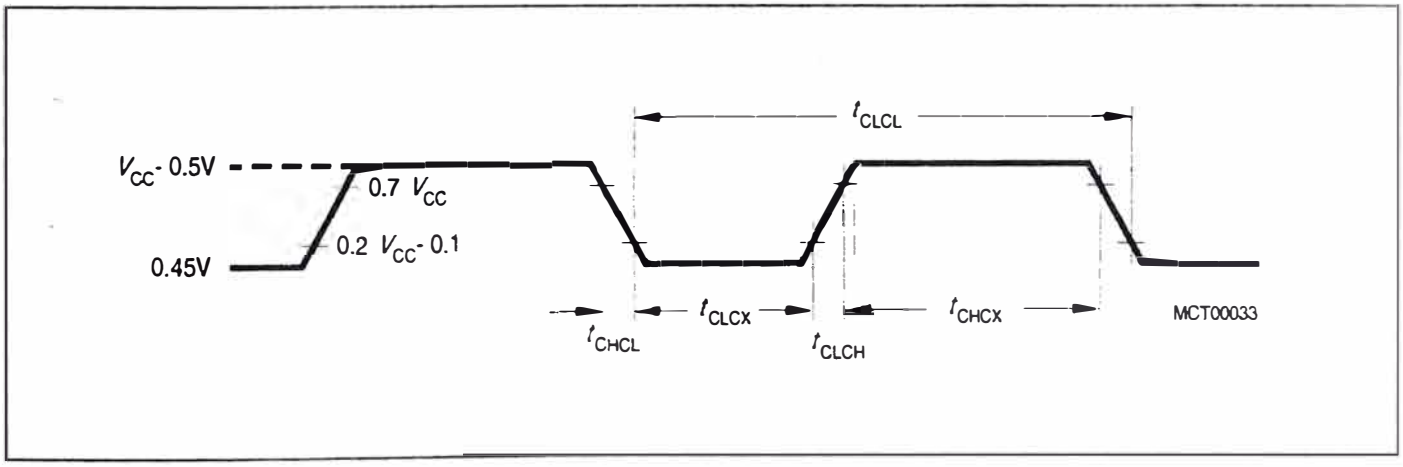
[1] Interfacing the SAB 80C515 to devices with float times up to 75 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

AC Characteristics (cont'd)

Parameter	Symbol	Limit values				Unit
		12 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
External Data Memory Characteristics						
$\overline{RD}$ pulse width	t_{RLRH}	400	–	6 t_{CLCL} – 100	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	400	–	6 t_{CLCL} – 100	–	ns
Address hold after ALE	t_{LLAX2}	132	–	2 t_{CLCL} – 35	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	252	–	5 t_{CLCL} – 165	ns
DATA hold after $\overline{RD}$	t_{RHDX}	0	–	0		ns
Data float after $\overline{RD}$	t_{RHDZ}	–	97	–	2 t_{CLCL} – 70	ns
ALE to valid data in	t_{LLDV}	–	517	–	8 t_{CLCL} – 150	ns
Address to valid data in	t_{AVDV}	–	585	–	9 t_{CLCL} – 165	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	200	300	3 t_{CLCL} – 50	3 t_{CLCL} + 50	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	43	123	t_{CLCL} – 40	t_{CLCL} + 40	ns
Address valid to $\overline{WR}$	t_{AVWL}	203	–	4 t_{CLCL} – 130	–	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	33	–	t_{CLCL} – 50	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	288	–	7 t_{CLCL} – 150	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	13	–	t_{CLCL} – 50	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

AC Characteristics (cont'd)

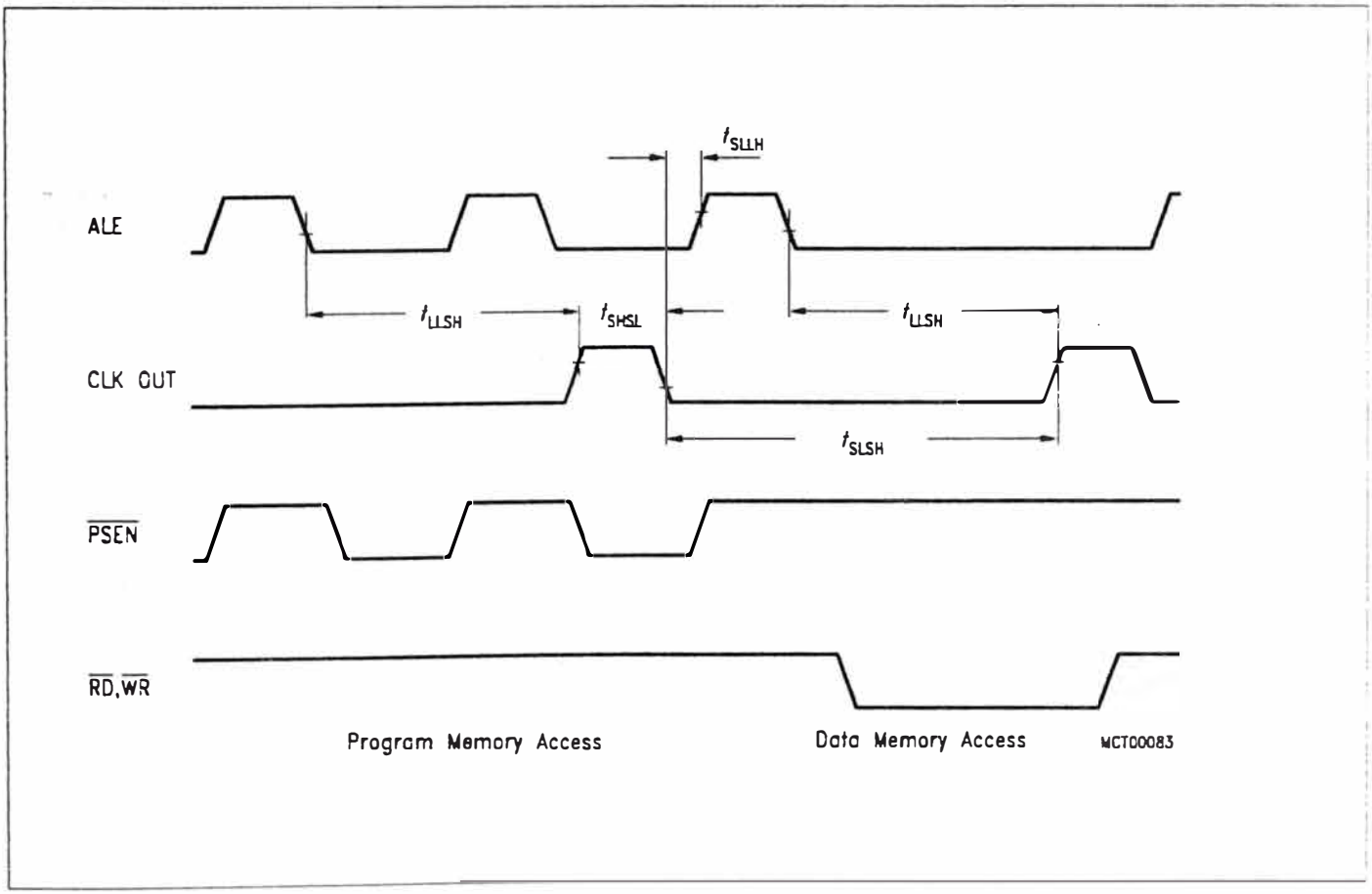
Parameter	Symbol	Limit values		Unit
		Variable clock Frequ. = 3.5 MHz to 12 MHz		
		min.	max.	
External Clock Drive				
Oscillator period	t_{CLCL}	83.3	285	ns
Oscillator frequency	$1/t_{CLCL}$	0.5	12	MHz
High time	t_{CHCX}	20	–	ns
Low time	t_{CLCX}	20	–	ns
Rise time	t_{CLCH}	–	20	ns
Fall time	t_{CHCL}	–	20	ns



External Clock Cycle

AC Characteristics (cont'd)

Parameter	Symbol	Limit values				Unit
		12 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 12 MHz		
		min.	max.	min.	max.	
System Clock Timing						
ALE to CLKOUT	t_{LLSH}	543	–	7 t_{CLCL} – 40	–	ns
CLKOUT high time	t_{SHSL}	127	–	2 t_{CLCL} – 40	–	ns
CLKOUT low time	t_{SLSH}	793	–	10 t_{CLCL} – 40	–	ns
CLKOUT low to ALE high	t_{SLLH}	43	123	t_{CLCL} – 40	t_{CLCL} + 40	ns



System Clock Timing

AC Characteristics for SAB 80C515-16/80C535-16

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V (C_L for Port 0, ALE and $\overline{PSEN}$ outputs = 100 pF;
C_L for all outputs = 80 pF) T_A = 0 to 70 °C for SAB 80C515-16/80C535-16
T_A = – 40 to 85 °C for SAB 80C515-16/80C535-16-T40/85

Parameter	Symbol	Limit values				Unit
		16 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 16 MHz		
		min.	max.	min.	max.	
Program Memory Characteristics						
ALE pulse width	t_{LHLL}	85	–	$2\,t_{CLCL} - 40$	–	ns
Address setup to ALE	t_{AVLL}	33	–	$t_{CLCL} - 30$	–	ns
Address hold after ALE	t_{LLAX}	28	–	$t_{CLCL} - 35$	–	ns
ALE to valid instruction in	t_{LLIV}	–	150	–	$4\,t_{CLCL} - 100$	ns
ALE to $\overline{PSEN}$	t_{LLPL}	38	–	$t_{CLCL} - 25$	–	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	153		$3\,t_{CLCL} - 35$		ns
$\overline{PSEN}$ to valid instruction in	t_{PLIV}	–	88	–	$3\,t_{CLCL} - 100$	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ} [1]	–	43	–	$t_{CLCL} - 20$	ns
Address valid after $\overline{PSEN}$	t_{PXAV} [1]	55		$t_{CLCL} - 8$		ns
Address to valid instruction in	t_{AVIV}	–	198	–	$5\,t_{CLCL} - 115$	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	–	0	–	ns

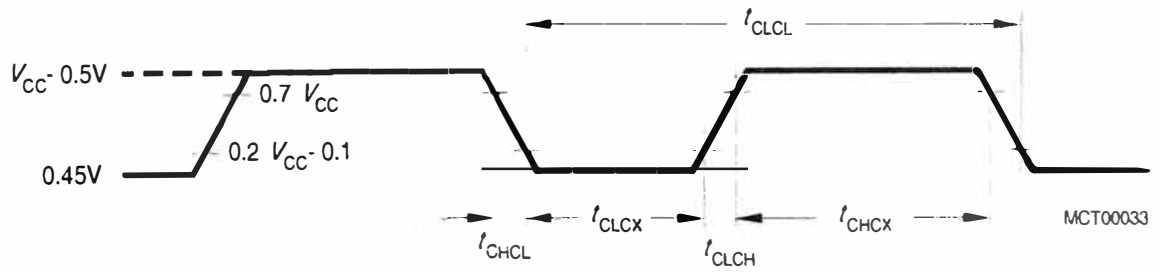
[1] Interfacing the SAB 80C515-16 to devices with float times up to 55 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

AC Characteristics (cont'd)

Parameter	Symbol	Limit values				Unit
		16 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 16 MHz		
		min.	max.	min.	max.	
External Data Memory Characteristics						
$\overline{RD}$ pulse width	t_{RLRH}	275	–	6 t_{CLCL} – 100	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	275	–	6 t_{CLCL} – 100	–	ns
Address hold after ALE	t_{LLAX2}	90	–	2 t_{CLCL} – 35	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	148	–	5 t_{CLCL} – 165	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	55	–	2 t_{CLCL} – 70	ns
ALE to valid data in	t_{LLDV}	–	350	–	8t_{CLCL} – 150	ns
Address to valid data in	t_{AVDV}	–	398	–	9 t_{CLCL} – 165	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	138	238	3 t_{CLCL} – 50	3 t_{CLCL} + 50	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	23	103	t_{CLCL} – 40	t_{CLCL} + 40	ns
Address valid to $\overline{WR}$	t_{AVWL}	120	–	4 t_{CLCL} – 130	–	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	13	–	t_{CLCL} – 50	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	288	–	7 t_{CLCL} – 150	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	13	–	t_{CLCL} – 50	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

AC Characteristics (cont'd)

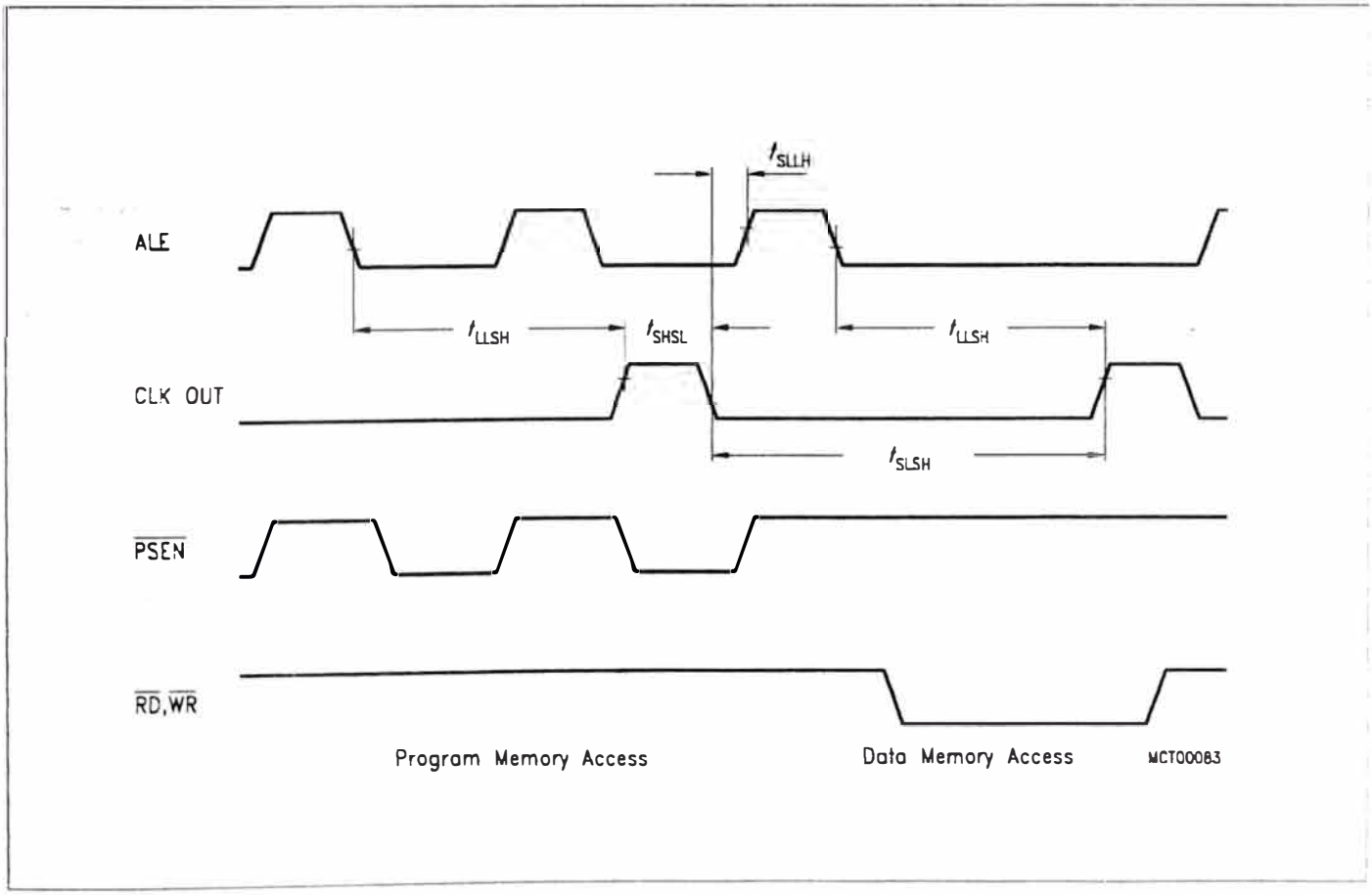
Parameter	Symbol	Limit values		Unit
		Variable clock Frequ. = 3.5 MHz to 16 MHz		
		min.	max.	
External Clock Drive				
Oscillator period	t_{CLCL}	62.5	285	ns
Oscillator frequency	$1/t_{CLCL}$	0.5	16	MHz
High time	t_{CHCX}	15	–	ns
Low time	t_{CLCX}	15	–	ns
Rise time	t_{CLCH}	–	15	ns
Fall time	t_{CHCL}	–	15	ns



External Clock Cycle

AC Characteristics (cont'd)

Parameter	Symbol	Limit values				Unit
		16 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 16 MHz		
		min.	max.	min.	max.	
System Clock Timing						
ALE to CLK OUT	t_{LLSH}	398	–	7 t_{CLCL} – 40	–	ns
CLK OUT high time	t_{SHSL}	85	–	2 t_{CLCL} – 40	–	ns
CLK OUT low time	t_{SLSH}	585	–	10 t_{CLCL} – 40	–	ns
CLK OUT low to ALE high	t_{SLLH}	23	103	t_{CLCL} – 40	t_{CLCL} + 40	ns



System Clock Timing

AC Characteristics for SAB 80C515-20 / 80C535-20

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V T_A = 0 °C to + 70 °C
(C_L for port 0, ALE and $\overline{PSEN}$ outputs = 100 pF; C_L for all other outputs = 80 pF)

Parameter	Symbol	Limit values				Unit
		20 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 20 MHz		
		min.	max.	min.	max.	
Program Memory Characteristics						
ALE pulse width	t_{LHLL}	60	–	2 t_{CLCL} – 40	–	ns
Address setup to ALE	t_{AVLL}	20	–	t_{CLCL} – 30	–	ns
Address hold after ALE	t_{LLAX}	20	–	t_{CLCL} – 30	–	ns
ALE low to valid instr in	t_{LLIV}	–	100	–	4 t_{CLCL} – 100	ns
ALE to $\overline{PSEN}$	t_{LLPL}	25	–	t_{CLCL} – 25	–	ns
$\overline{PSEN}$ pulse width	t_{PLPH}	115	–	3 t_{CLCL} – 35	–	ns
$\overline{PSEN}$ to valid instr in	t_{PLIV}	–	75	–	3 t_{CLCL} – 75	ns
Input instruction hold after $\overline{PSEN}$	t_{PXIX}	0	–	0	–	ns
Input instruction float after $\overline{PSEN}$	t_{PXIZ}*)	–	40	–	t_{CLCL} – 10	ns
Address valid after $\overline{PSEN}$	t_{PXAV}*)	47	–	t_{CLCL} – 3	–	ns
Address to valid instr in	t_{AVIV}	–	190	–	5 t_{CLCL} – 60	ns
Address float to $\overline{PSEN}$	t_{AZPL}	0	–	0	–	ns

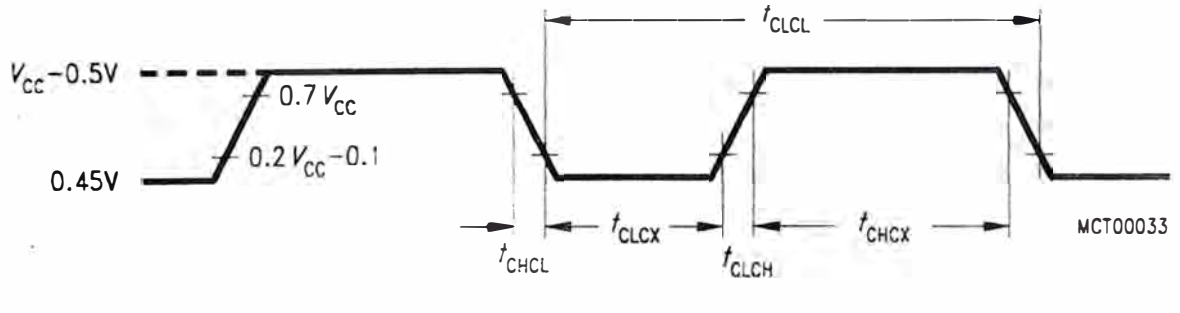
*) Interfacing the SAB 80C515 / 80C535 microcontrollers to devices with float times up to 45 ns is permissible. This limited bus contention will not cause any damage to port 0 drivers.

AC Characteristics (cont'd)

Parameter	Symbol	Limit values				Unit
		20 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 20 MHz		
		min.	max.	min.	max.	
External Data Memory Characteristics						
$\overline{RD}$ pulse width	t_{RLRH}	200	–	6 t_{CLCL} – 100	–	ns
$\overline{WR}$ pulse width	t_{WLWH}	200	–	6 t_{CLCL} – 100	–	ns
Address hold after ALE	t_{LLAX2}	65	–	2 t_{CLCL} – 35	–	ns
$\overline{RD}$ to valid data in	t_{RLDV}	–	155	–	5 t_{CLCL} – 95	ns
Data hold after $\overline{RD}$	t_{RHDX}	0	–	0	–	ns
Data float after $\overline{RD}$	t_{RHDZ}	–	40	–	2 t_{CLCL} – 60	ns
ALE to valid data in	t_{LLDV}	–	250	–	8 t_{CLCL} – 150	ns
Address to valid data in	t_{AVDV}	–	285	–	9 t_{CLCL} – 165	ns
ALE to $\overline{WR}$ or $\overline{RD}$	t_{LLWL}	100	200	3 t_{CLCL} – 50	3 t_{CLCL} + 50	ns
Address valid to $\overline{WR}$ or $\overline{RD}$	t_{AVWL}	70	–	4 t_{CLCL} – 130	–	ns
$\overline{WR}$ or $\overline{RD}$ high to ALE high	t_{WHLH}	20	80	t_{CLCL} – 30	t_{CLCL} + 30	ns
Data valid to $\overline{WR}$ transition	t_{QVWX}	5	–	t_{CLCL} – 45	–	ns
Data setup before $\overline{WR}$	t_{QVWH}	200	–	7 t_{CLCL} – 150	–	ns
Data hold after $\overline{WR}$	t_{WHQX}	10	–	t_{CLCL} – 40	–	ns
Address float after $\overline{RD}$	t_{RLAZ}	–	0	–	0	ns

AC Characteristics (cont'd)

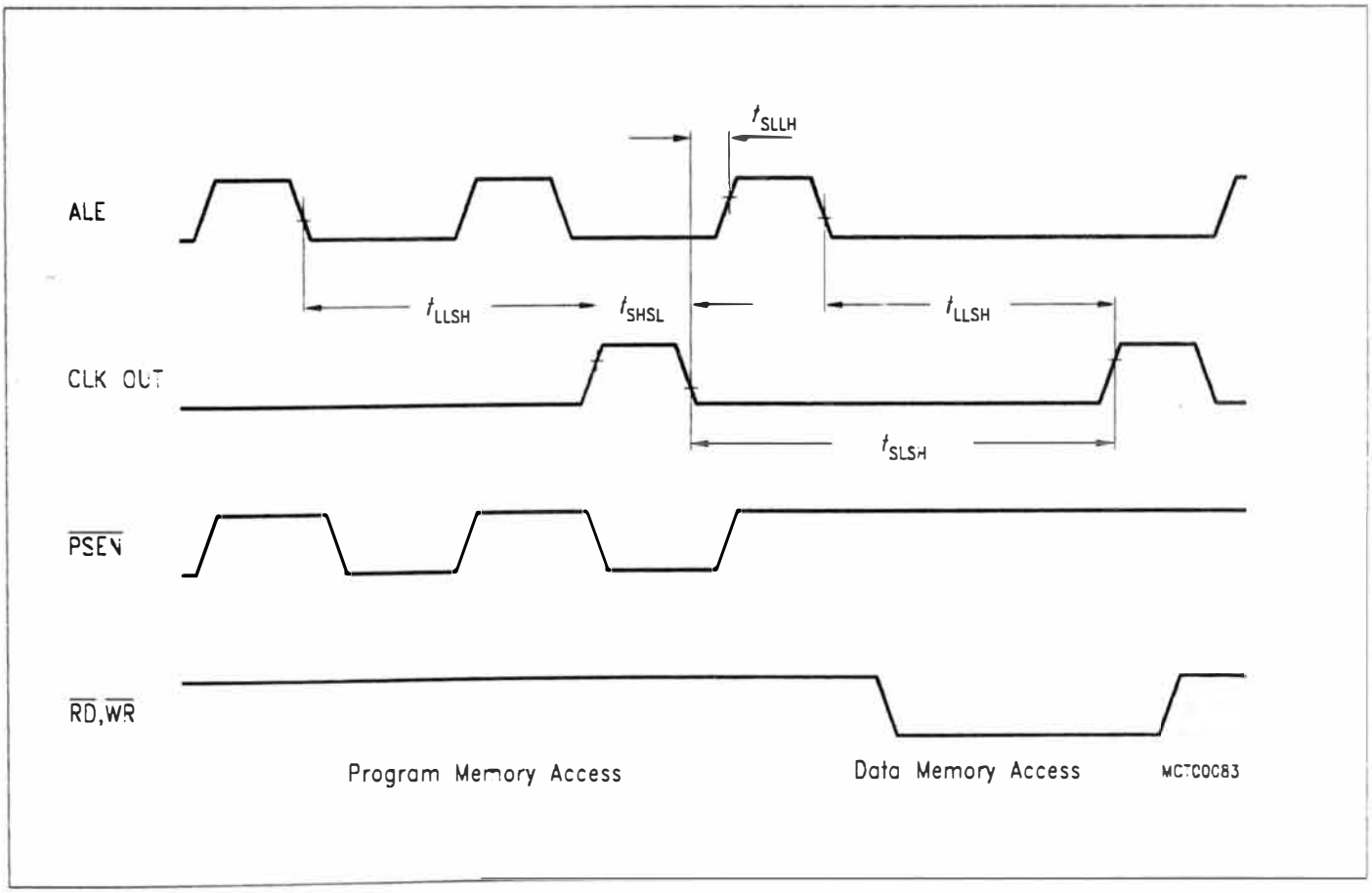
Parameter	Symbol	Limit Values		Unit
		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 20 MHz		
		min.	max.	
External Clock Drive				
Oscillator period	t_{CLCL}	50	285	ns
High time	t_{CHCX}	12	$t_{CLCL} - t_{CLCX}$	ns
Low time	t_{CLCX}	12	$t_{CLCL} - t_{CHCX}$	ns
Rise time	t_{CLCH}	–	12	ns
Fall time	t_{CHCL}	–	12	ns



External Clock Cycle

AC Characteristics (cont'd)

Parameter	Symbol	Limit values				Unit
		20 MHz clock		Variable clock $1/t_{CLCL}$ = 3.5 MHz to 20 MHz		
		min.	max.	min.	max.	
System Clock Timing						
ALE to CLKOUT	t_{LLSH}	310	–	$7\ t_{CLCL} - 40$	–	ns
CLKOUT high time	t_{SHSL}	60	–	$2\ t_{CLCL} - 40$	–	ns
CLKOUT low time	t_{SLSH}	460	–	$10\ t_{CLCL} - 40$	–	ns
CLKOUT low to ALE high	t_{SLLH}	10	90	$t_{CLCL} - 40$	$t_{CLCL} + 40$	ns

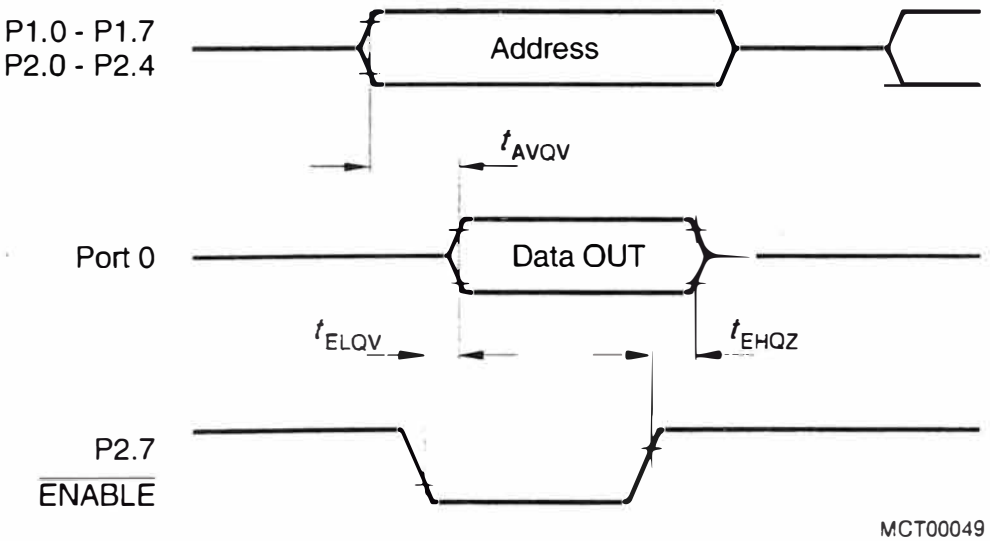


External Clock Cycle

ROM Verification Characteristics

T_A = 25 °C ± 5 °C; V_{CC} = 5 V ± 10 %; V_{SS} = 0 V

Parameter	Symbol	Limit values		Unit
		min.	max.	
ROM Verification				
Address to valid data	t_{AVQV}	–	48 t_{CLCL}	ns
ENABLE to valid data	t_{ELQV}	–	48 t_{CLCL}	ns
Data float after ENABLE	t_{EHQZ}	0	48 t_{CLCL}	ns
Oscillator frequency	$1/t_{CLCL1}$	4	6	MHz
Address to valid data	t_{AVQV}	–	48 t_{CLCL}	ns

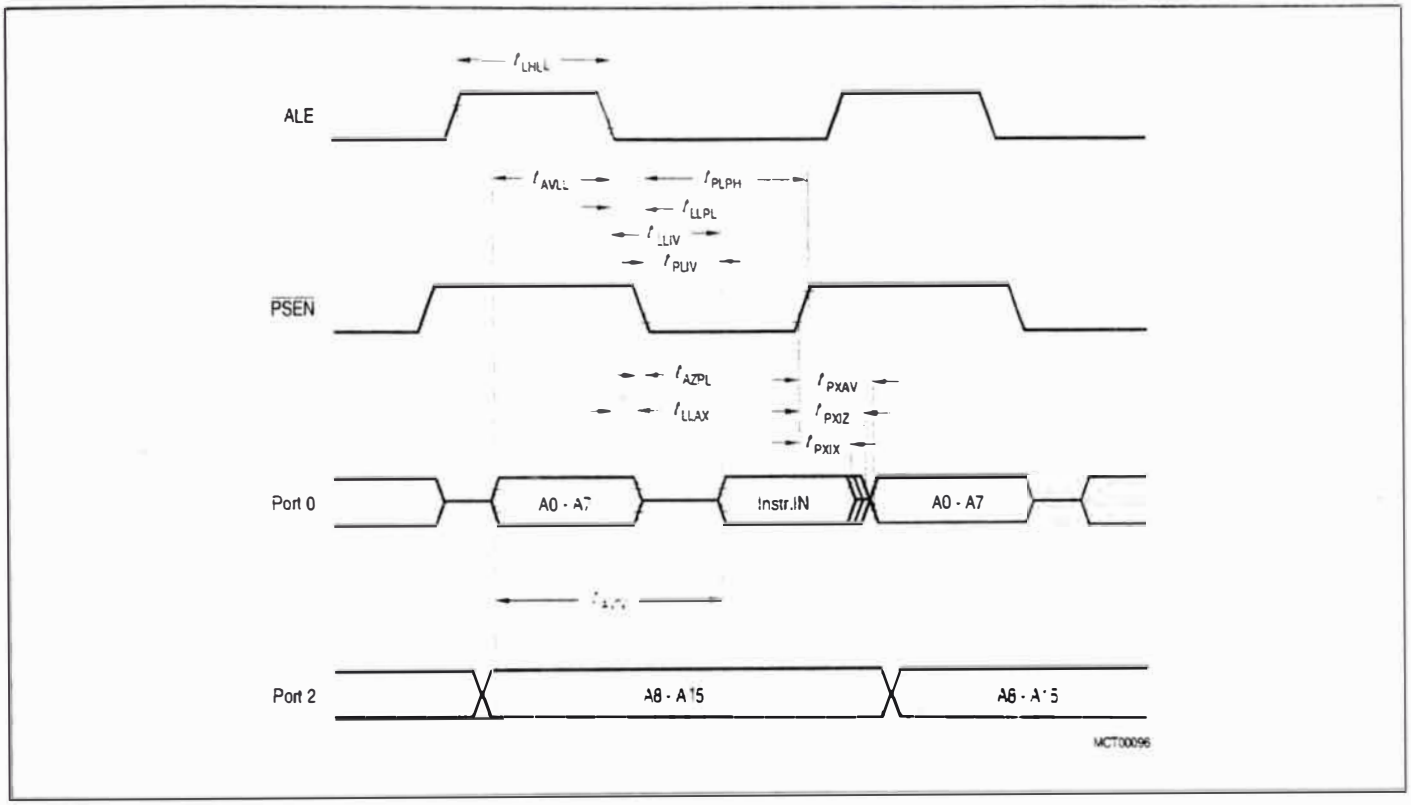


Address: P1.0 - P1.7 = A0 - A7
P2.0 - P2.4 = A8 - A12
Data: P0.0 - P0.7 = D0 - D7

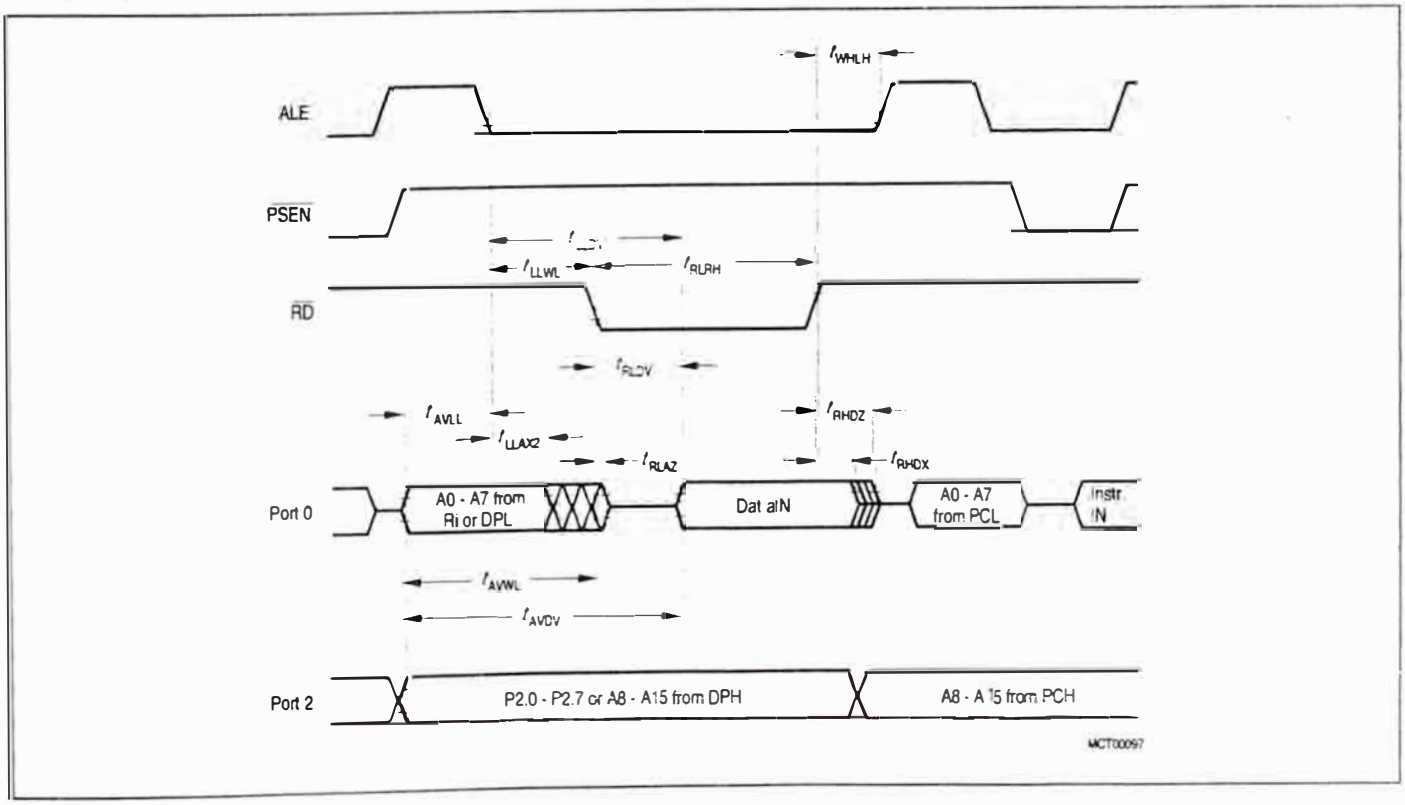
Inputs: P2.5 - P2.6, $\overline{PSEN}$ = V_{SS}
ALE, $\overline{EA}$ = V_{IH}
$\overline{RESET}$ = V_{SS}

ROM Verification

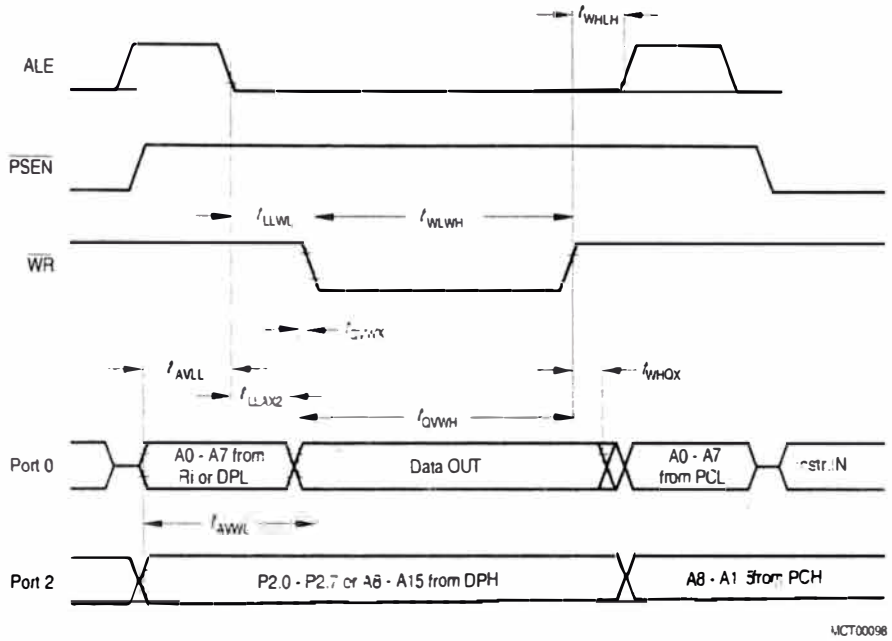
Waveforms



Program Memory Read Cycle

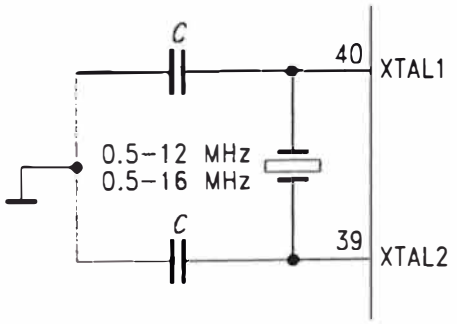


Data Memory Read Cycle



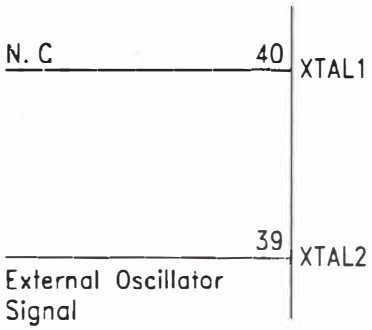
AC inputs during testing are driven at V_{CC} – 0.5 V for a logic '1' and 0.45 V for a logic '0'. Timing measurements are made at $V_{IH\ min}$ for a logic '1' and $V_{IL\ max}$ for a logic '0'.

Data Memory Write Cycle



C = 30pF ± 10pF
(incl. stray capacitance)

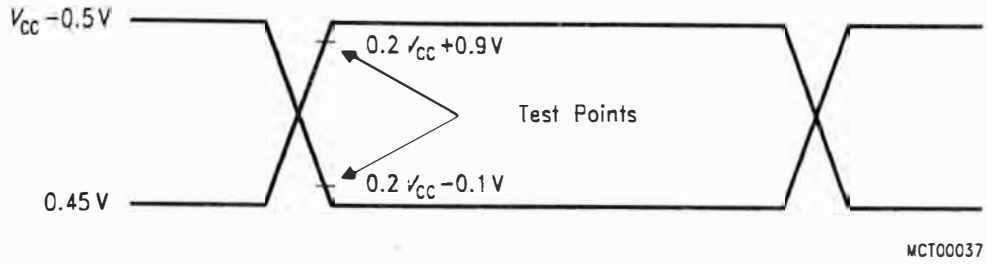
Crystal Oscillator Mode



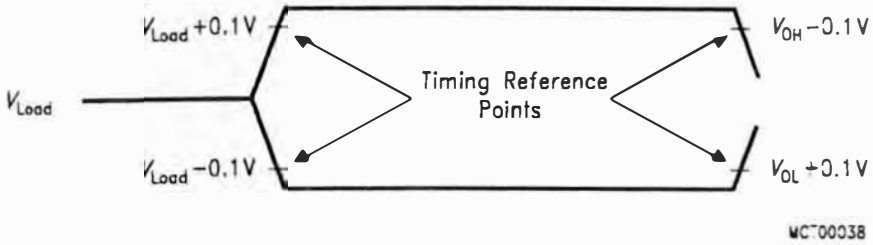
Driving from External Source

For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV deviation from the load voltage V_{OH}/V_{OL} occurs. $I_{OL}/I_{OH} \geq \pm 20$ mA.

Recommended Oscillator Circuits



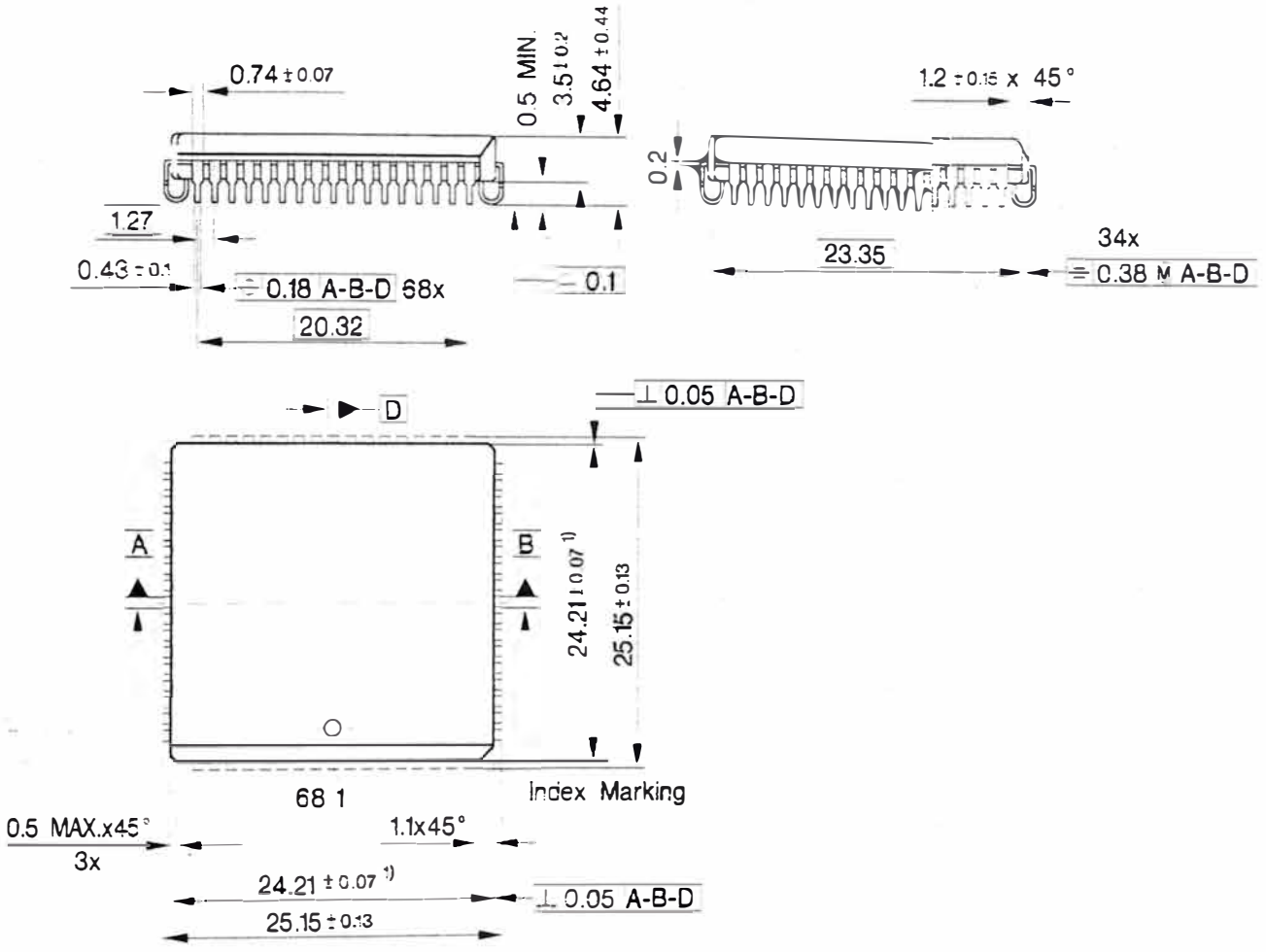
AC Testing: Input, Output Waveforms



AC Testing: Float Waveforms

Package Outlines

Plastic Package, P-LCC-68 – SMD
(Plastic Leaded Chip-Carrier)



1) Does not include plastic or metal protrusion of 0.15 max. per side

Sorts of Packing
Package outlines for tubes, trays etc. are contained in our Data Book "Package Information".

SMD = Surface Mounted Device

Dimensions in mm

ANEXO C
ESPECIFICACIONES TECNICAS DE SENSORES, ACTUADORES Y ACCESORIOS

C.1 SENSORES

Sensores

Los sensores empleados en el proceso fueron, de acuerdo a su naturaleza de dos tipos: mecánicos y de proximidad.

a) Sensores mecánicos.

Constituidos por finales de carreras con contactos conmutantes. En el proceso se emplearon cuatro de estos sensores: S1, S2, S3 y S4, que indican la posición de los vástagos de los cilindros neumáticos A y B, respectivamente.

Especificaciones Técnicas

M odelo:	V3L – 511092 – D8K
Corriente Nominal:	100 mA
Voltaje Nominal AC:	125 V AC
Voltaje Nominal DC:	30 V DC
Contactos Conmutantes:	1NO + 1NC

b) Sensores de Proximidad.

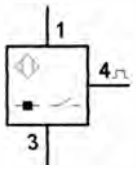
Se emplearon los siguientes tipos: un inductivo (B8), dos ópticos (B7 y B8) y dos magnéticos (B5 y B6).

Especificaciones Técnicas del Sensor Inductivo B_8

Tipo de Detección: Detección de presencia de objetos metálicos, mediante el uso de un oscilador cuyo bobinado constituye la cara

sensible. Frente a esta se crea un campo magnético alterno. Cuando se coloca un objeto metálico en ese campo, las corrientes inducidas generan una carga adicional que provoca el cese de las oscilaciones y en su etapa de salida hace cambios de estado a un contacto NO y/o NC.

Símbolo:



Marca:	Wenglor
Modelo:	1120UB
Configuración:	PNP
Alcance Nominal:	20 mm.
Voltaje de Polarización:	10 – 30 V DC
Voltaje Residual:	0,3 V DC
Corriente Nominal: de conmutación	400 mA.
Frecuencia de Conmutación:	< 2 KHz.
Histerésis:	15 % del alcance nominal
Carcasa:	Plástica
Versión:	Cable
Temperatura de operación:	-25 ^{0}C a +70 ^{0}C.

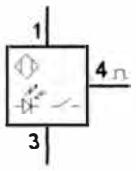
Tipo de Protección: IP 67

Características de Salida: 1 contacto NO.

Especificaciones Técnicas del Sensor Optico B_7

Tipo de Detección: Detecta presencia de objetos de todo tipo de material, enviando un haz de luz visible roja, de una fuente (LED) que incide en el objeto detectado y se refleja el rayo que es recepcionado por un fototransistor, ubicado al igual que la fuente en la cara de detección, causando que en su etapa de salida se abra o cierre un contacto.

Símbolo:



Marca: Wenglor

Modelo: # K12PB8

Configuración: PNP

Alcance Nominal: 18 - 120 mm .

Voltaje de Polarización: 10 – 30V DC

Voltaje residual: 0,3 V DC

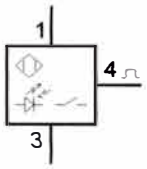
Corriente Nominal: 100 mA.
de conmutación

Frecuencia de Conmutación: 600 KHz.

Histerésis:	< 15 % del alcance nominal
Tipo de Luz :	Luz Roja 640 nm.
Carcasa:	Plástica
Versión:	Plug
Temperatura de Operación:	- 25 ^{0}C a +60 ^{0}C.
Tipo de Protección:	IP 67
Características de Salida:	1 contacto NO

Especificaciones Técnicas del Sensor Optico B_8

Símbolo:



Marca:	Telemecanique
Modelo:	XV5P18 PP340
Configuración:	PNP
Alcance Nominal:	0 – 200 mm.
Voltaje de Polarización:	12 a 24 V DC
Voltaje residual :	0,3 V DC
Corriente Nominal: de conmutación	100 mA.
Frecuencia de Conmutación:	< 2 KHz.
Histerésis:	15 % del alcance nominal

Tipo de Luz : infrarrojo 900 nm.

Carcasa: Plástica

Versión Cable

Temperatura de Operación: - 25 ^{0}C a +70 ^{0}C

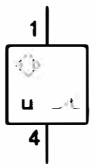
Tipo de Protección: IP 67

Característica de salida: 1 contacto NO

Especificaciones Técnicas de los Sensores Magnéticos B_5 y B_6.

Tipo de Detección: Detecta objetos ferromagnéticos que pasan en un sentido u otro, frente a un micro contacto encapsulado en un tubo de vidrio, con el cierre del contacto.

Símbolo:



Marca: Elobeu

Configuración: PNP

Voltaje de Alimentación AC: 10 – 220 V AC

Voltaje de Alimentación DC: 10 – 220 V DC

Máxima Corriente de : Conmutación 0,5 A

Conductancia: 100 Ohms

Frecuencia de Conmutación: 500 Hz

Tiempo de Conmutación: 2 ms

Temperatrura de Operación: - 20 ^{0}C a +60 ^{0}C

Tipo de Protección: IP 66

C.2 ACTUADORES

ACTUADORES

Los actuadores o elementos finales de control estan constituidos por dos electroválvulas, Y1, Y2 e Y3; que reciben las señales ON/OFF del microcontrolador a través de una interface, cuyas características se describieron en el Capítulo 2, sección 2.3. A su vez estas electroválvulas actuaban sobre los cilindros A, B y C, las que realizan la elevación y selección de paquetes,respectivamente, obedeciendo a la lógica de control programada.

Especificaciones Técnicas de las Electroválvulas Y_1, Y_2 e Y_3.

Símbolo:

Marca :	SMC
Modelo :	VF5120
Tipo de Actuación :	5 vias y 2 posiciones de conmutación
Tipo de Fluido :	Aire
Rango de Presión de Operación :	1,5 a 9 Kgf/cm^2
Temperatura de Operación :	Max. + 50^{0}C
Tiempo de Respuesta :	15 ms o menos (a 5 Kgf/cm^2)
Frecuencia de Conmutación :	15 Hz
Over ride Manual :	Tipo Non - Locking
Lubricación :	Aceite Turbina # 1 ISO VG 32
Posición de Montaje :	Libre

Resistencia a Impactos :	45 ~` 1000 Hz
Cubierta :	A prueba de polvo.
Voltaje del Solenoide	24 V DC
Aislamiento de la Bobina	Clase E (120 ^{0}C)
Consumo de Energía :	1,8 W
Conexiones :	1/8" Salidas a Cilindros
	¼" Entrada de aire.

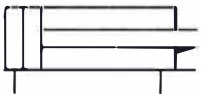
C.3 ACCESORIOS

ACCESORIOS

Dentro del rubro de accesorios se encuentran los tres cilindros neumáticos de dob e efecto, seis reguladores de caudal, una unidad de mantenimiento y dos motores eléctricos trifásicos, tipo jaula de ardilla con sistema de arranque directo.

Especificaciones Técnicas del Cilindro Neumático A

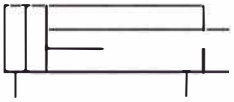
Símbolo :



Tipo :	Doble Efecto
Marca :	Micro
Fluido :	Aire
Presión de peración Máxima :	10 bar (1000 KPa)
Presión de Operación Mínima :	0,5 bar (50 Kpa)
Temperatura de Operación :	- 20 ^{0}C a + 80 ^{0}C
Velocidad del Vástago :	50 a 1000 mm/s
Diámetro :	35 mm
Longuitud :	380 mm
Sujeción :	Con patillas
Conexiones :	1/8"

Especificaciones Técnicas del Cilindro B.

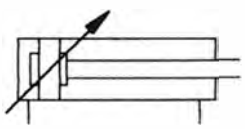
Símbolo :



Tipo :	Doble Efecto
Marca :	SMC
Modelo :	CG N25 – 100 YP
Fluido :	Aire
Presión de Operación Máxima :	10 bar (1000 Kpa)
Presión de Operación Mínima :	0,5 bar (50 Kpa)
Temperatura de Operación :	- 20 ^{0}C a + 80 ^{0}C
Velocidad del Vástago :	50 a 1000 mm/s
Diámetro :	30 mm
Longuitud :	165 mm
Sujeción :	Con patillas
Conexiones :	1/8"

Especificaciones Técnicas del Cilindro C

Símbolo :



Tipo :	Doble Efecto
Marca :	Herión
Tipo :	ISO 228
Fluido :	Aire
Presión de Operación Máxima :	10 bar (1000 Kpa)
Presión de Opertación Mínima :	0,5 bar (50 Kpa)
Temperatura de Operación :	- 20 ^{0}C a +80 ^{0}C
Velocidad del Vástago :	50 a 1000 mm/s
Diámetro :	25 mm
Longuitud :	120 mm
Sujeción :	Con patillas
Conexiones :	1/8"

Especificaciones Técnicas de los Reguladores de Caudal

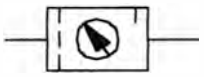
Símbolo :

Marca :	SMC
Modelo :	AS 2000
Fluido :	Aire
Presión :	10 bar (1000 Kpa)

Temperatura de Operación :	- 20 ^{0}C a + 80 ^{0}C
Conexión :	1/8"

Especificaciones Técnicas de la Unidad de Mantenimiento

Símbolo :



Marca :	SMC
Modelo :	AC 3000 – 025
Presión de Regulación :	0 a 1000 Kpa
Filtro :	Para partículas hasta 5 micras
Lubricador :	Aceite
Apertura y Cierre :	Con válvula manual de cierre marca SMC, Modelo VHS400 – 02

Especificaciones Técnicas de los Motores

Marca :	WEG
Tipo :	Jaula de Ardilla
Motor de Inducción :	Trifásico
Frecuencia :	60 Hz

Potencia :	0,55 KW (0,75 CV)
Velocidad :	1715 RPM
Voltaje Nominal :	220/380/440 V AC
Corriente Nominal :	3,00/1,74/1,50 A
Eficiencia	69 %
Cosø :	0,70

BIBLIOGRAFIA

1. Siemens: Microcomputer Componentes; SAP 80515 SAP 80C515, 8-Bit Single- Chip Microcontroller Family; User's Manual 08.95
2. Siemens: Tarjeta Compuboard 80C535; Guía de Laboratorio del curso de Microcontroladores de la FIEE-UNI
3. Gonzáles Vásquez,José A.: Introducción a los Microcontroladores"; Mc Graw Hill; 1ra. Edición en Español, 1992.
4. Martinez Eyzaguirre, José: Diseño y Construcción de un Módulo para la Enseñanza de Automatismos Neumáticos Basados en los PLCs"; Tesis para optar el Titulo de Ingeniero Electrónico de la Facultad de Ingeniería Eléctrica y Electrónica de la UNI, sustentada en Noviembre de 1997.
5. Thomas A. Hughes: Programable Controllers; Instrument Society Of América, ISA, 1989.
6. A. Porras/A.P. Montenegro: Automatas programables; Mc Graw-Hill/ Interamericana de España S.A.; 1990
7. W. Deppert / K. Stall, Dispositivos neumáticos; Marcombo

Boixareu Editores; 1974.

8. W. Deppert / K. Stall, Aplicaciones de la neumática; Marcombo Boixareu Editores; 1977

9. H. Meixner / R. Kobler: Neumática para la Formación Profesional; Publicado por el servicio de información de FESTO DIDACTIC GmbH; Setiembre de 1976.

10. H. Meixner / E. Sauer: Training System in Control Technology, Publicado por el servicio de información de FESTO DIDACTIC GmbH; Abril de 1984.

11. J.P.Hasebrink / R. Kobler: Introducción a los Mandos Neumáticos; Publicado por el servicio de información de FESTO DIDACTIC GmbH; 1982.

12. H. Meixner Neumática Avanzada; Publicado por el servicio de información de FESTO DIDACTIC GmbH, Esslingen; 1989.

13. Sick Optic Electronic: Sensores Opto-Electrónicos Sensick, Programa General, Publicado por Sick Opto Electronic, 1997.

14. SMC del Japón: 4.5 Port Solenoid Valve, Series Vz 1000 / 3000 / 5000; N° Catálogo E 121-A; publicación del año 1996.

15. SMC del Japón: Air Cylinder / Series CG1K, Catálogo General, publicado por Sick Opto Electronic, el año 1997.