



Guía didáctica para el estudiante

UNIDAD TEMÁTICA 4

ARQUITECTURA BÁSICA

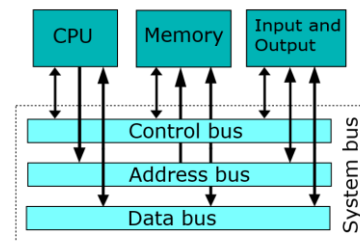
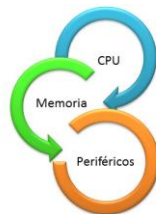
Autores: Ing. Daniel Argüello
Ing. Santiago Pérez
Revisor: Ing. Higinio Facchini

ISBN 978-950-42-0158-8



9 789504 201588

Versión 10.0 – Diciembre 2015



Objetivos:

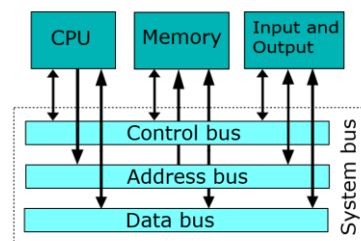
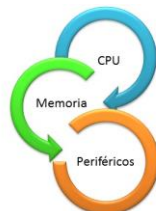
Objetivo General (Según Plan de Estudios vigente)

- Aplicar los aspectos centrales que hacen a la tecnología de la computación y conceptos sobre hardware, plataformas y arquitecturas, para abordar las cuestiones vinculadas al procesamiento y a las comunicaciones.

Objetivos específicos:

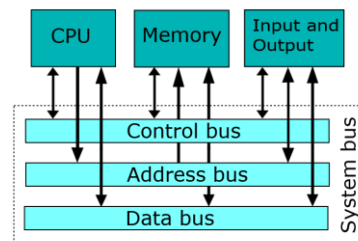
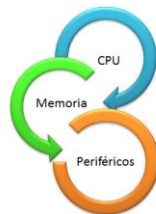
Los objetivos específicos en la Unidad Temática 4, de la Cátedra Arquitectura de Computadoras, son que el estudiante:

- Identifique las funciones de cada bloque en un computador elemental basado en la arquitectura de Von Neumann con transferencia de datos e instrucciones.
- Desarrolle la capacidad para crear programas de lógica simple en lenguaje de bajo nivel.



Índice:

❖ 1 Arquitectura de von Neumann.....	4
❖ 2 Máquina Elemental	8
❖ 2.1 Introducción	8
❖ 2.2 Arquitectura de la Computadora Elemental	12
❖ 2.2.1 Unidad de Procesamiento Central	12
❖ 2.2.2 Unidad de Memoria.....	12
❖ 2.2.3 Unidad de Entrada/Salida	13
❖ 2.3 El Conjunto de Instrucciones	14
❖ 2.4 El Ciclo de Máquina	18
❖ 2.5 Flujo de Información	19
❖ 2.6 Unidad de Control	21
❖ 2.6.1 Unidad de Control Cableada.....	22
❖ 2.6.2 Secuenciador	25
❖ 2.6.3 Lógica de Control.....	32
❖ 2.7 Unidad de Control Microprogramada	33
❖ 2.8 Bus en la Máquina Elemental	42
❖ 2.9 Unidad Aritmética y Lógica	44

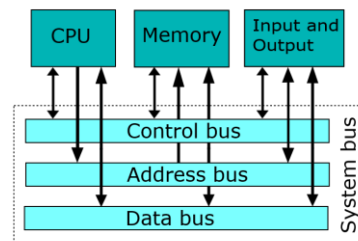
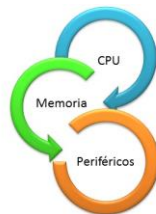


Guía didáctica para el estudiante

UNIDAD TEMÁTICA 4

Sección 1

ARQUITECTURA DE VON NEUMANN



1 ARQUITECTURA DE VON NEUMANN

La estructura básica de una computadora está compuesta por cinco elementos (Figura 4.1):

- 1) La Unidad de Proceso Central (CPU)
- 2) La Unidad de Memoria (UM)
- 3) La Unidad de Entrada/Salida (UE/S)
- 4) La Unidad de Buses (UB)
- 5) El Programa Almacenado en UM (PA)

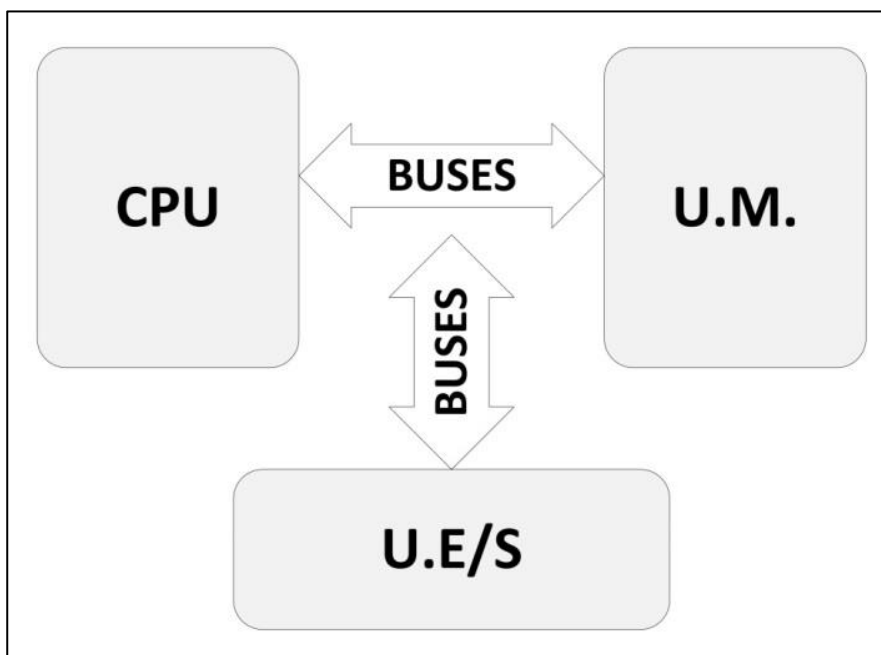
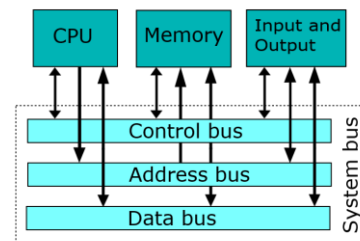
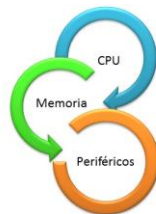


Fig. 4.1 Estructura básica de una computadora

Estos cinco elementos y su funcionamiento reciben el nombre de Arquitectura de Computadoras de John von Neumann, en recuerdo como uno de los principales propulsores (1903-1957), y la utilizaremos como definición de COMPUTADORA.

Es común referenciar a los primeros cuatro elementos como HARDWARE y al quinto como SOFTWARE.

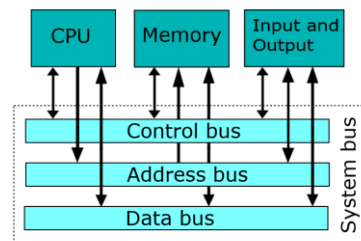
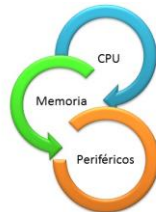


El funcionamiento de la Computadora implica una fuerte cooperación entre las unidades que la componen y supone que, en la memoria, reside un conjunto ordenado de INSTRUCCIONES llamado PROGRAMA.

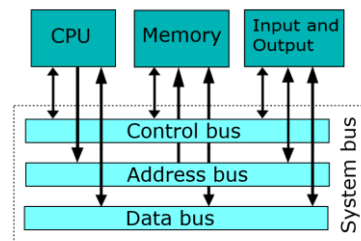
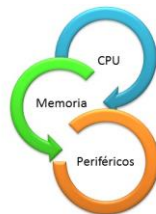
Funcionamiento:

Las INSTRUCCIONES del PROGRAMA son buscadas y ejecutadas secuencialmente (por la CPU) hasta que el programa finaliza. Durante la ejecución de una instrucción es posible que sea necesario obtener datos (o guardar resultados) desde o hacia la UM, o también obtener datos (o enviarlos) desde o hacia la UE/S, por lo tanto, será necesario coordinar el movimiento de datos a la UM y UE/S.

- 1) CPU: La **UNIDAD DE PROCESAMIENTO CENTRAL** cuenta con:
 - Varios REGISTROS. Los REGISTROS son usados principalmente para almacenar temporalmente la información (datos, direcciones, instrucciones).
 - Unidad Aritmética y Lógica (ALU). La ALU realiza las operaciones lógico-aritméticas necesarias, eventualmente, en la ejecución de una instrucción.
 - Unidad de Control (UC). La UC es la parte de la CPU que realiza concretamente el control del sistema, para lo cual genera un conjunto de señales de control (órdenes) que se diseminan en todo el Sistema.
- 2) UM: La **UNIDAD DE MEMORIA** almacena programas y datos. Es, principalmente, una memoria RAM de lectura/escritura ya que es necesario leer y escribir en la misma. No obstante, una porción de la UM podría ser ROM, para almacenar programas o algún tipo de datos que no cambien y deberían permanecer aún que se quite la alimentación.
- 3) UE/S: La **UNIDAD DE ENTRADA/SALIDA** se encarga de interconectar la computadora con los dispositivos externos (periféricos) accesibles al usuario, como son el teclado, el monitor, la impresora, etc., por los cuales podemos ingresar o extraer información a o desde la CPU. Las UE/S cuentan con INTERFACES (que generalmente realizan la adaptación de niveles eléctricos de las señales) y CONTROLADORES que son sistemas digitales para controlar periféricos específicos.



- 4) UB: La **UNIDAD DE BUSES** cumple la función de transportar información entre las unidades del Sistema. Por el tipo de información, podrían clasificarse los buses en:
- BUS de DATOS: Transporta Operandos o Instrucciones
 - BUS de DIRECCIONES: Transporta Direcciones
 - BUS de CONTROL: Transporta señales de Control.
- 5) PA: El **PROGRAMA ALMACENADO** es un conjunto de INSTRUCCIONES almacenadas correlativamente en la Unidad de Memoria. Los programas tienen un comienzo y un fin. Las instrucciones son propias de cada máquina y se conocen como Set de Instrucciones.

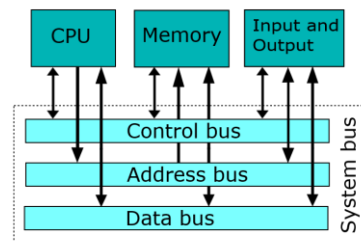
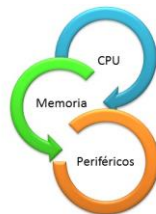


Guía didáctica para el estudiante

UNIDAD TEMÁTICA 4

Sección 2

MÁQUINA ELEMENTAL



2 MÁQUINA ELEMENTAL

2.1 INTRODUCCIÓN

A fin de estudiar la arquitectura de una computadora como la definida, presentaremos una Máquina Elemental utilizando el planteo de Caxton Foster, de su libro Arquitectura de Computadoras.

Previo al desarrollo es necesario distinguir entre las máquinas asíncronas y las síncronas. En las primeras, las órdenes emitidas por la UC se suceden secuencialmente, una después de la anterior. En cambio, en las síncronas las órdenes son emitidas por la UC en forma síncrona con una señal de sincronismo externa llamada reloj.

Las máquinas asíncronas son más rápidas que las síncronas, ya que las acciones ordenadas por la UC tardan lo que tarda el hardware involucrado, mientras que en las síncronas todas las acciones tardan un periodo de reloj. A pesar de la ventaja mencionada, las máquinas se construyen síncronas por la simplicidad en el hardware. Se puede inferir que la UC de una máquina asíncrona deberá contar con hardware adicional que detecte la finalización de las acciones ordenadas. Esto implica una complicación que no justifica la ganancia en tiempo que resultaría.

Las características principales de esta computadora son:

- Arquitectura de von Neumman,
- Usa sistema binario y aritmética en complemento a 2,
- Su memoria es de 4096 x 16 (4096 posiciones de memoria de 16 bits cada una), y
- Usa punto fijo y sus datos son de 16 bits (15 bits de mantisa y un bit de signo) como se observa en la Figura 4.2

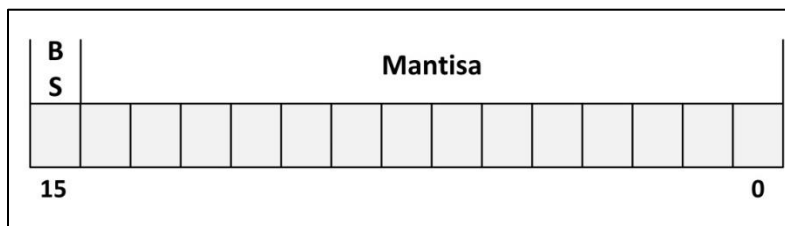
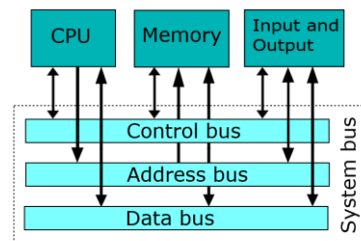
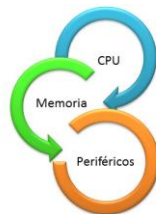


Fig. 4.2 Formato de los datos numéricos en punto fijo



- Usa instrucciones de formato fijo de 16 bits (4 bits para el código de operación y 12 bits para el campo de dirección) tal como se presentan en la Figura 4.3.

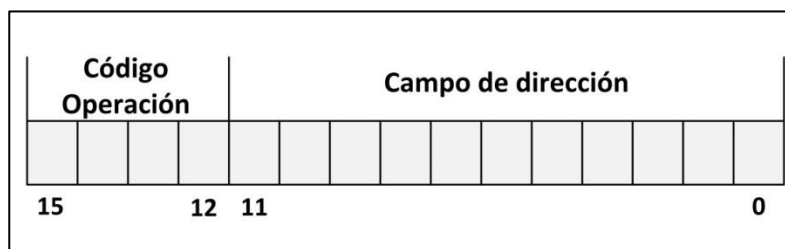


Fig. 4.3 Formato de las instrucciones

- Usa un BUS común de 16 bits multiplexado (datos y direcciones).
- La ALU realiza, sobre 1 o 2 operandos, las siguientes operaciones:
 - ADD
 - OR
 - XOR
 - AND
 - RAL
 - NOR
- Capacidad para manejar hasta 64 periféricos

Externamente tiene una consola con llaves, pulsadores y luces, que permite al operador comunicarse con la Máquina. El aspecto se muestra en la Figura 4.4.

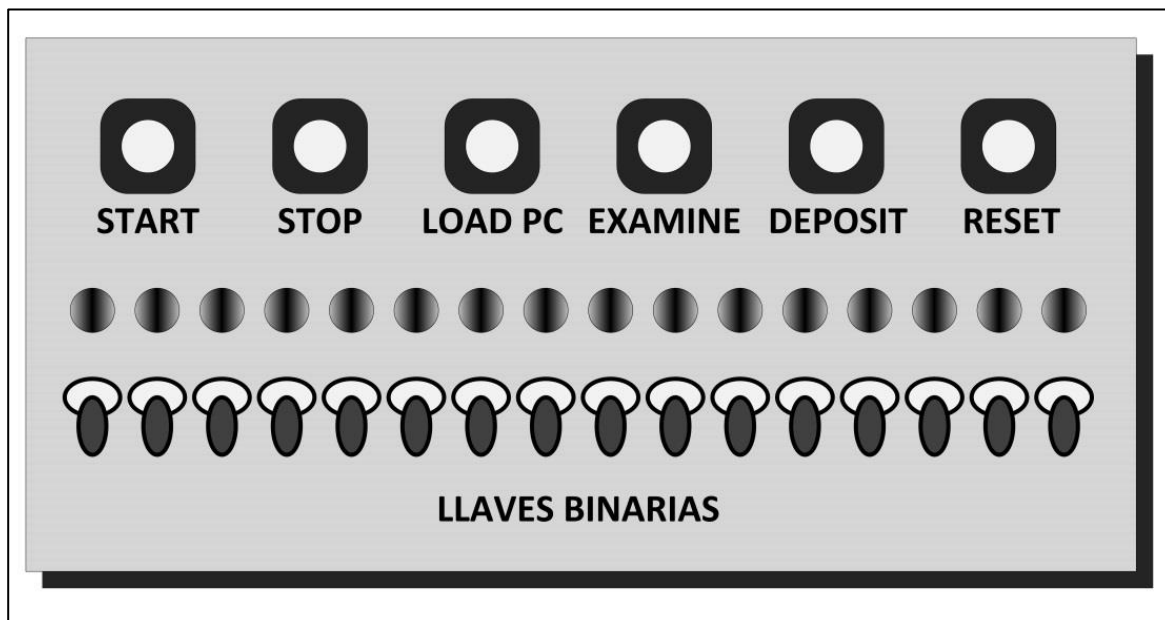
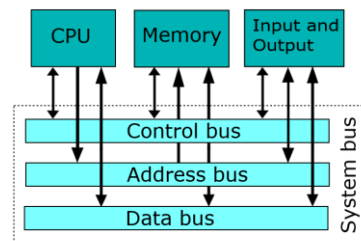
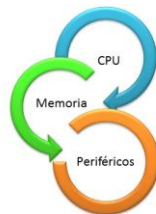


Fig. 4.4 Consola de la máquina elemental

El Diagrama en Bloques de la Máquina Elemental se presenta en la Figura 4.5:

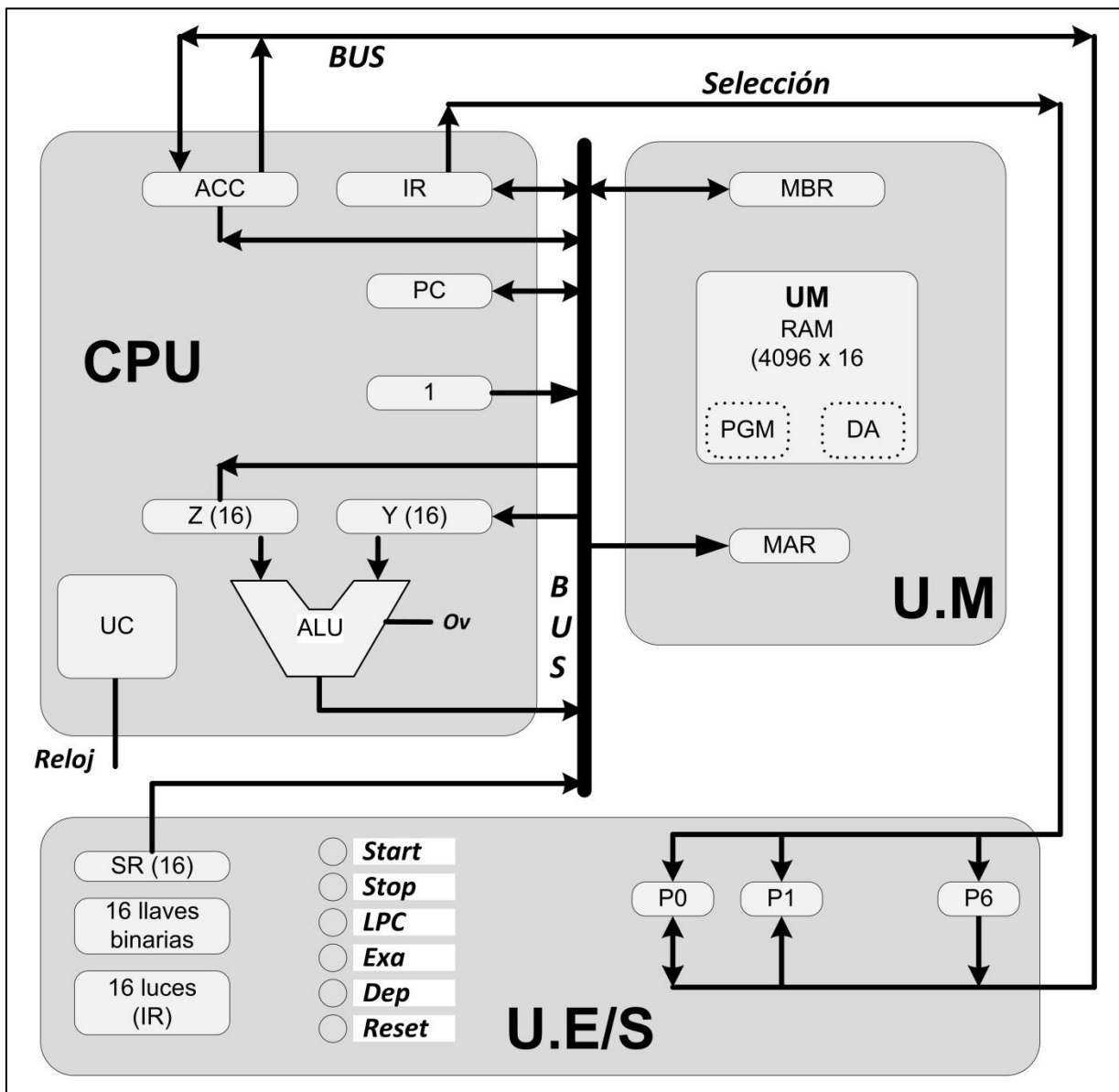
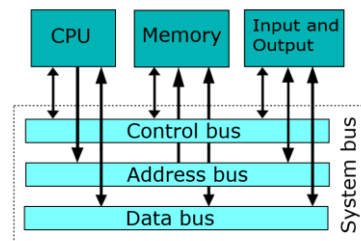
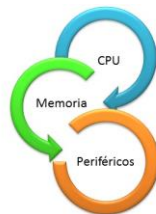
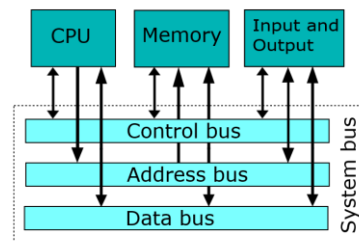
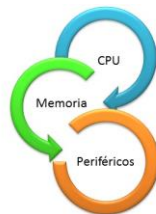


Fig. 4.5 Diagrama en bloques de la máquina elemental



2.2 ARQUITECTURA DE LA COMPUTADORA ELEMENTAL

2.2.1 UNIDAD DE PROCESAMIENTO CENTRAL

La CPU cuenta con:

- REGISTROS:

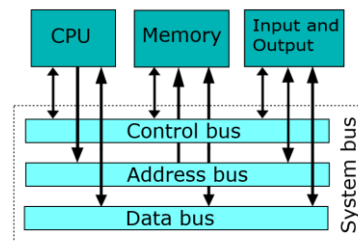
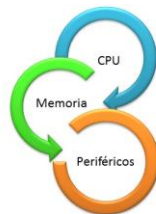
- ACC (16 bits): El registro Acumulador es un registro de propósitos generales, se utiliza para distintos fines
- IR (16 bits): El registro de Instrucciones es un registro de propósito especial. Se utiliza para guardar la instrucción que se ha buscado desde la memoria.
- PC (12 bits): El registro Contador de Programa es un registro de propósitos especiales. Se utiliza para guardar la dirección de memoria de la próxima instrucción a buscar.
- Registro Z (16 bits) y Registro Y (16 bits): Los Registros de Operandos de entrada a la ALU son registros de propósito especial y no son accedidos por el programador.
- 1 (1 bit): Es un registro de propósito especial. Permite generar un 1.

- **ALU:** realiza las operaciones lógico-aritméticas ya mencionadas sobre los operandos ubicados en los registros Z e Y. Genera la señal OV (overflow) si el resultado de la suma aritmética supera la capacidad de representación.

- **UC:** La Unidad de Control tiene una entrada de sincronismo y genera las órdenes en sincronismo con los flancos del Reloj. Existen dos tipos de Unidades de Control: la UC Cableada y la UC Microprogramada.

2.2.2 UNIDAD DE MEMORIA

La UM es una memoria RAM de lectura/escritura como las vistas anteriormente. Posee un registro asociado a sus líneas de datos llamado MBR (16 bits), a través del cual pasará la información que se lea o escriba en la memoria. También posee otro registro asociado a las líneas de dirección llamado MAR (12 bits), cuyo contenido deberá ser la dirección de memoria a la que se pretende acceder. La RAM que usa tiene un Tiempo de Acceso = 400 ns.



2.2.3 UNIDAD DE ENTRADA/SALIDA

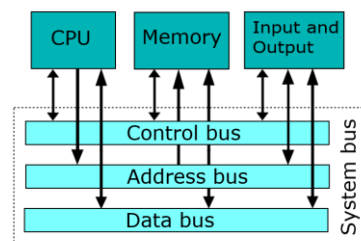
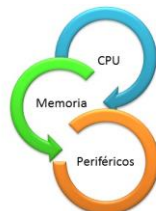
Como dispositivos de entrada/salida se observan dos casos

- La **CONSOLA**: que está conectada directamente del BUS. En su interior posee el hardware que posibilita su comunicación con la CPU a través de los 6 pulsadores, las 16 llaves, las 16 luces y el Registro de Llaves (16bits), que reflejan el estado de las 16 llaves binarias. Las 16 luces indican el contenido del IR.

Los pulsadores:

- ◆ Pulsador de ARRANQUE (Start)
- ◆ Pulsador de PARADA (Stop)
- ◆ Pulsador de CARGAR PC (Load PC)
Acción: Transfiere el contenido del SR al PC
- ◆ Pulsador de DEPOSITAR (Deposit)
Acción: Transfiere el contenido del SR a la posición de memoria indicada por el PC, luego incrementa el PC
- ◆ Pulsador de EXAMINAR (Examine)
Acción: Transfiere el contenido de la posición de memoria indicada por PC al IR, luego incrementa el PC
- ◆ Pulsador de RESET (Reset)
Acción: Borra la RAM y los Registros

- Los **DISPOSITIVOS EXTERNOS**: llamados periféricos (P0 a P63). Tienen destinado un BUS bidireccional dedicado a E/S (8 bits) para los Datos y un BUS (6 bits) para selección de periférico (el hardware de interface no se muestra en la Figura 4.5). Las transferencias de entrada - salida en la Blue son realizadas bajo control de programa (veremos instrucciones para tal fin) y a través del acumulador. Cuando se ejecuta una instrucción de entrada/salida, la Unidad de Control genera la señal Transferencia (TRA) hacia los dispositivos externos. Sólo el dispositivo seleccionado responderá con una señal Ready (R) cuando haya completado su tarea. Este tipo de control se llama HANDSHAKING.



2.3 EL CONJUNTO DE INSTRUCCIONES

Como anticipamos, esta computadora posee un tipo de instrucción de longitud fija de 16 bits con el siguiente formato (Figura 4.6):

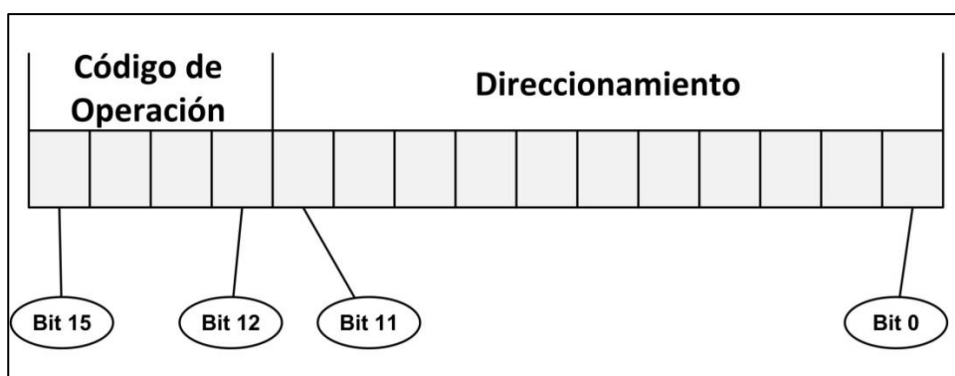
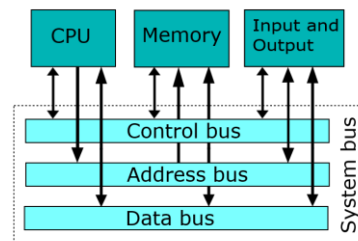
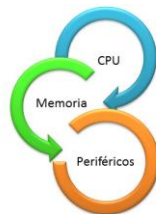


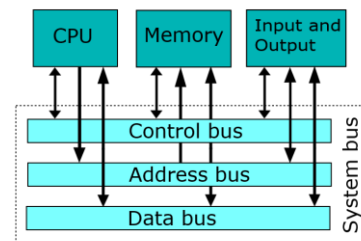
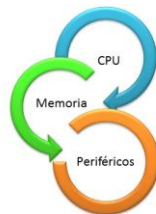
Fig. 4.6 Formato de las instrucciones

Debido a que tenemos solamente 4 bits para el código de operación, se dispone a lo sumo de 16 instrucciones que se presentan en la Tabla 4.1:



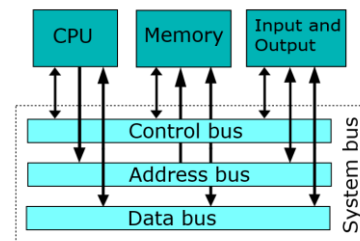
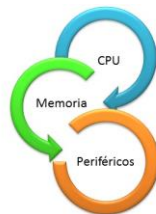
Código binario	Código octal	Nombre	Mnemónico
0000XXXXXXXXXXXXX	00XXXX	HALT (PARAR)	HLT XXXX
0001XXXXXXXXXXXXX	01XXXX	ADD (SUMAR)	ADD XXXX
0010XXXXXXXXXXXXX	02XXXX	OR-EXCLUSIVA	XOR XXXX
0011XXXXXXXXXXXXX	03XXXX	AND	AND XXXX
0100XXXXXXXXXXXXX	04XXXX	OR	IOR XXXX
0101XXXXXXXXXXXXX	05XXXX	NOT (COMPLEM. A 1)	NOT XXXX
0110XXXXXXXXXXXXX	06XXXX	LOAD AC (CARGAR AC)	LDA XXXX
0111XXXXXXXXXXXXX	07XXXX	STORE AC (ALMAC. AC)	STA XXXX
1000XXXXXXXXXXXXX	10XXXX	SALTO SUBROUTINA	SRJ
1001XXXXXXXXXXXXX	11XXXX	SALTO CONDICIONAL	JMA XXXX
1010XXXXXXXXXXXXX	12XXXX	SALTO INCONDICIONAL	JMP XXXX
1011XXXXXXXXXXXXX	13XXXX	ENTRADA DE DATOS	INP XXXX
1100XXXXXXXXXXXXX	14XXXX	SALIDA DE DATOS	OUT XXXX
1101XXXXXXXXXXXXX	15XXXX	ROTACION DE BITS	RAL XXXX
1110XXXXXXXXXXXXX	16XXXX	COPIAR LLAVES	CSA XXXX
1111XXXXXXXXXXXXX	17XXXX	NO OPERACION	NOP XXXX

Tabla 4.1 Conjunto de instrucciones

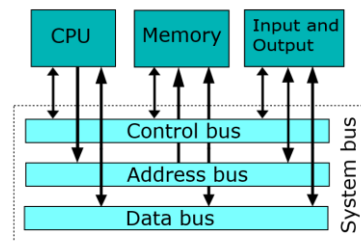
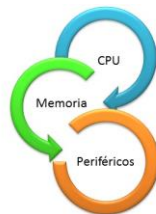


La descripción de cada una de las instrucciones es la siguiente:

- ❖ **HLT XXXX:** Detiene el funcionamiento de la computadora. Presionando el pulsador START de la consola principal la computadora arranca nuevamente siguiendo con la ejecución de la instrucción siguiente al HALT. El campo de direcciones XXXX se ignora.
- ❖ **ADD XXXX:** Realiza la suma aritmética en complemento a dos de los operandos ubicados en el acumulador y en la dirección de memoria expresada en el campo XXXX, y deja el resultado en el acumulador. El contenido de la dirección de memoria XXXX no cambia, mientras que si lo hace el dato que se encontraba en el acumulador. Si el resultado de la suma es mayor que $2^{15}-1$ o menor que -2^{15} la computadora se detiene.
- ❖ **XOR XXXX:** Realiza la OR-Exclusiva bit a bit de los operandos ubicados en el acumulador y en la dirección de memoria expresada en el campo XXXX, y deja el resultado en el acumulador. El contenido de la dirección de memoria XXXX no cambia, mientras que si lo hace el dato que se encontraba en el acumulador.
- ❖ **AND XXXX:** Realiza la AND bit a bit de los operandos ubicados en el acumulador y en la dirección de memoria expresada en el campo XXXX, y deja el resultado en el acumulador. El contenido de la dirección de memoria XXXX no cambia, mientras que si lo hace el dato que se encontraba en el acumulador.
- ❖ **IOR XXXX:** Realiza la OR bit a bit de los operandos ubicados en el acumulador y en la dirección de memoria expresada en el campo XXXX, y deja el resultado en el acumulador. El contenido de la dirección de memoria XXXX no cambia, mientras que si lo hace el dato que se encontraba en el acumulador.
- ❖ **NOT XXXX:** Cada bit del dato en el acumulador es remplazado por su complemento lógico. La dirección de memoria XXXX se ignora.



- ❖ LDA XXXX: El contenido de la ubicación de memoria XXXX se copia en el acumulador. El contenido de la dirección de memoria XXXX no cambia, mientras que si lo hace el dato que se encontraba en el acumulador.
- ❖ STA XXXX: El contenido del acumulador se copia en la dirección de memoria XXXX. El contenido del acumulador no cambia, mientras que si lo hace el dato que se encontraba en la dirección de memoria.
- ❖ SRJ XXXX: Sirve para hacer un salto del programa a una subrutina. Para esto realiza una copia del contador del programa en los 12 bits más bajos del acumulador (en los 4 bits más altos del acumulador se ponen ceros). Luego, se copia el número XXXX en el contador del programa para que la próxima instrucción sea tomada de dicha dirección.
- ❖ JMA XXXX: Produce un salto a otra dirección de programa si el bit de signo del acumulador es uno (es decir si el acumulador contiene un número negativo). Si se cumple dicha condición, copia el número XXXX en el contador de programa y la próxima instrucción se toma de esta dirección. Si no se cumple la condición (el bit de signo del acumulador es cero, es decir, que el dato en el acumulador es positivo o cero), esta instrucción no realiza nada y el programa sigue normalmente.
- ❖ JMP XXXX: Produce un salto incondicional a otra parte del programa, dado que se copia el número XXXX en el contador de programa, y la próxima instrucción a ejecutar se toma de la dirección XXXX.
- ❖ INP XYYY: Los 8 bits de mayor peso del acumulador se colocan a cero, y el próximo carácter de 8 bits que viene del dispositivo externo YY se coloca en la parte baja del acumulador. La parte XX del campo de dirección se ignora. La próxima instrucción no se ejecuta hasta que la transferencia del dato se haya completado.
- ❖ OUT XYYY: Lo 8 bits más significativos del acumulador se envían al dispositivo externo YY. La parte XX del campo de dirección se ignora. Si el dispositivo externo no puede aceptar el dato en ese momento, la computadora espera hasta que se haya podido realizar la transferencia.



- ❖ RAL XXXX: Los bits del acumulador se rotan un lugar hacia la izquierda. El bit AC_{15} se coloca en AC_0 , de modo que el desplazamiento es cíclico. El campo de direcciones XXXX se ignora.
- ❖ CSA XXXX: El número que está en el registro de llaves (introducido por las llaves de la consola) se copia en el acumulador. El campo de direcciones XXXX se ignora.
- ❖ NOP XXXX: Esta instrucción no hace nada. El campo de direcciones XXXX se ignora.

2.4 EL CICLO DE MÁQUINA

Se llama “**ciclo de máquina**” de una computadora, al procedimiento que consta de todas las tareas necesarias para poder buscar y ejecutar completamente una instrucción del programa almacenado en memoria, que podemos sintetizar de la siguiente forma:

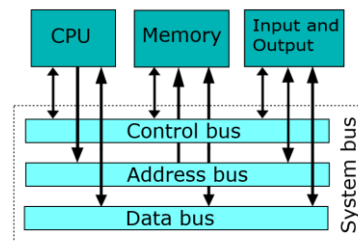
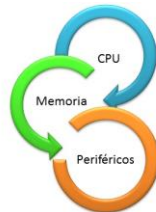
- Búsqueda de una instrucción a memoria
- Lectura e interpretación de esa instrucción
- Ejecución de la misma.
- Almacenamiento de resultados
- Preparación para leer la próxima instrucción.

Esta computadora elemental (BLUE) tiene un ciclo de máquina básico compuesto por dos partes:

- **CICLO DE BÚSQUEDA**
- **CICLO DE EJECUCIÓN**

Durante el **Ciclo de Búsqueda**, la instrucción almacenada en la memoria y apuntada por el **Contador de Programa (PC)** se localiza en la memoria y se copia en el **Registro de Instrucciones (RI)**. Luego, el número almacenado en el **PC** se incrementa en uno, logrando así que ahora apunte a la próxima celda de memoria (o sea, a la siguiente instrucción).

Al completar el **Ciclo de búsqueda**, la instrucción que está en el **IR** se analiza, decodifica y ejecuta. Si la presente instrucción no necesita hacer una nueva



búsqueda a memoria (de algún dato u operando) el **ciclo de máquina** termina en ese momento.

Si es necesario buscar un operando a memoria, entonces comienza el **Ciclo de Ejecución**, para realizar un nuevo acceso a memoria para traer al operando necesario y completar así la instrucción.

2.5 FLUJO DE INFORMACIÓN

Sin analizar las instrucciones en detalle todavía, mostraremos los movimientos de información entre registros dentro de la máquina:

- ♦ **El Flujo de direcciones (addresses)** en la BLUE son movimientos de 12 bits entre registros (Figura 4.7):
 - Load PC: envía los 12 bits más bajos del registro de llaves (R.Sw.) al PC.
 - Saltos (JMP, JMA, SRJ): envían los 12 bits más bajos del RI al PC.
 - SRJ (salto a subrutina): envía los 12 bits del PC al Acumulador (ACC).
 - Búsqueda de una instrucción: envía los 12 bits del PC al MAR.
 - Búsqueda de un Operando: envía los 12 bits más bajos de RI al MAR.
- ♦ **El Flujo de instrucciones y operandos** en la BLUE son movimientos de 16 bits entre registros (Figura 4.8):
 - CSA: copia los 16 bits del R.Sw. al ACC.
 - Deposit: copia los 16 bits del R.Sw. al MBR.
 - Instrucciones: se copian del MBR al RI.
 - LDA: copia los 16 bits del MBR al ACC.
 - STA: copia los 16 bits del ACC al MBR.
 - Operaciones de la ALU (en el ciclo de ejecución):
 - Copia los 16 bits del ACC al Registro Z de la ALU.
 - Copia los 16 bits del MBR al Registro Y de la ALU.
 - El Resultado (o salida de la ALU) se copia al ACC.

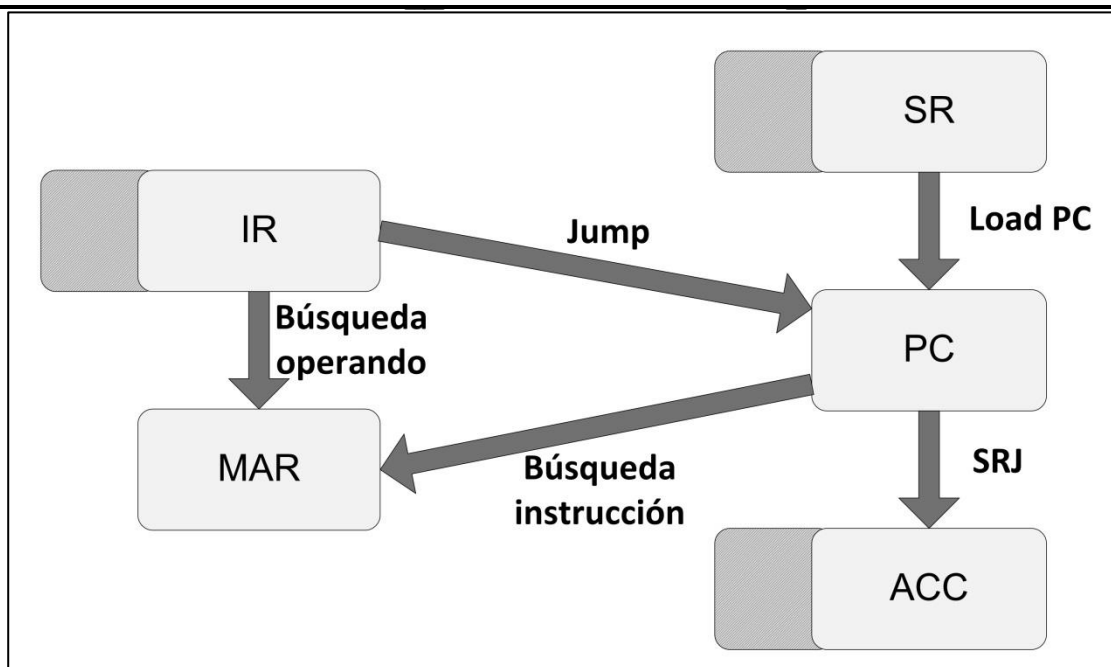
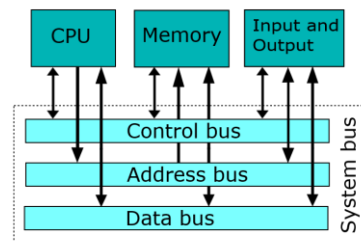
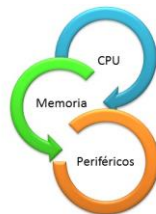


Fig. 4.7 La transmisión de direcciones en la Blue

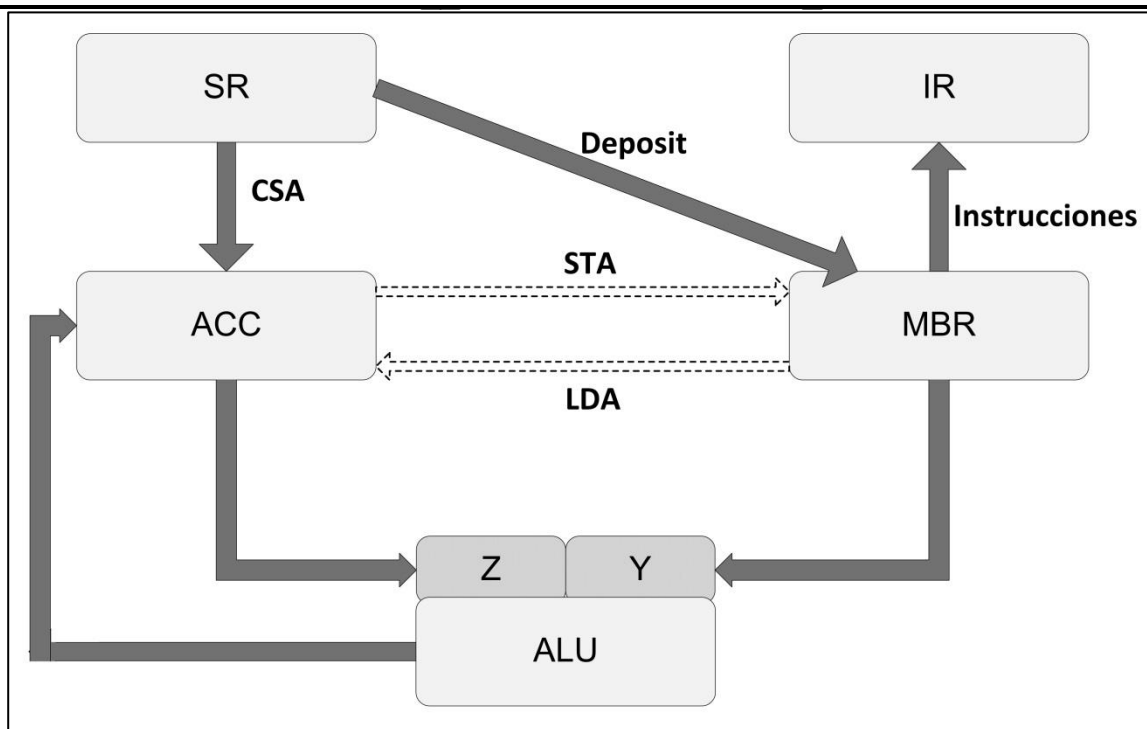
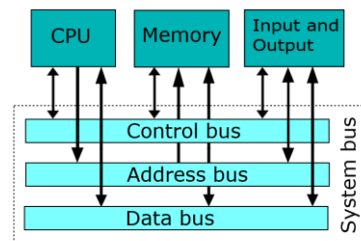
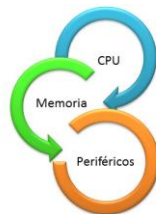


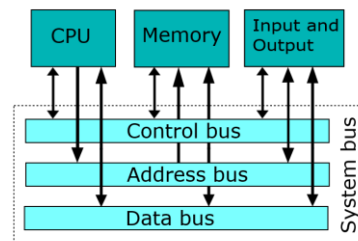
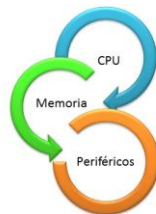
Fig. 4.8 La transmisión de instrucciones y operandos en la Blue

Para poder realizar las posibles transferencias vistas es necesario implementar algún esquema de relación entre los registros, memoria y unidades de E/S, como por ejemplo:

- BUS COMUN (elegido para BLUE)
- PUNTO A PUNTO

2.6 UNIDAD DE CONTROL

Las máquinas síncronas tienen Unidades de Control Síncronas, es decir, que tienen una entrada de sincronismo (reloj) que emite pulsos en periodos de tiempos fijos (frecuencia de reloj), y en cada pulso se realizan una o más tareas. Esto tiene como ventaja que mantiene las distintas tareas de cada dispositivo en orden y secuencia, y con un hardware sencillo. La desventaja es que ninguna tarea se puede realizar en menos tiempo que la duración de un ciclo de reloj.



La tarea de la unidad de control es coordinar todas las acciones de la máquina. Para este trabajo es necesaria una secuencia de pulsos y señales que deben generarse sincrónicamente al reloj.

Existen dos maneras de diseñar la Unidad de Control:

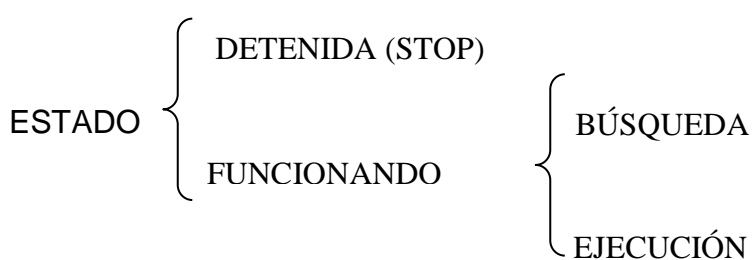
- Unidad de Control Cableada
- Unidad de Control Microprogramada

Ambas pueden verse como una caja negra con exactamente las mismas entradas y las mismas salidas. La diferencia está en su implementación interna.

2.6.1 Unidad de Control Cableada

En la Figura 4.9 se presenta el Diagrama en Bloques de la Unidad de Control Cableada. Está integrada por los siguientes componentes:

- a) **DECODIFICADOR DE INSTRUCCIONES:** Es un Sistema Combinacional. Un decodificador binario de 4 entradas y 16 salidas, que se utiliza para determinar qué instrucción contiene el IR. Por lo tanto, sus entradas serán IR12 a IR15.
- b) **SECUENCIADOR:** Es un Sistema Secuencial. El estado de la Máquina puede ser:



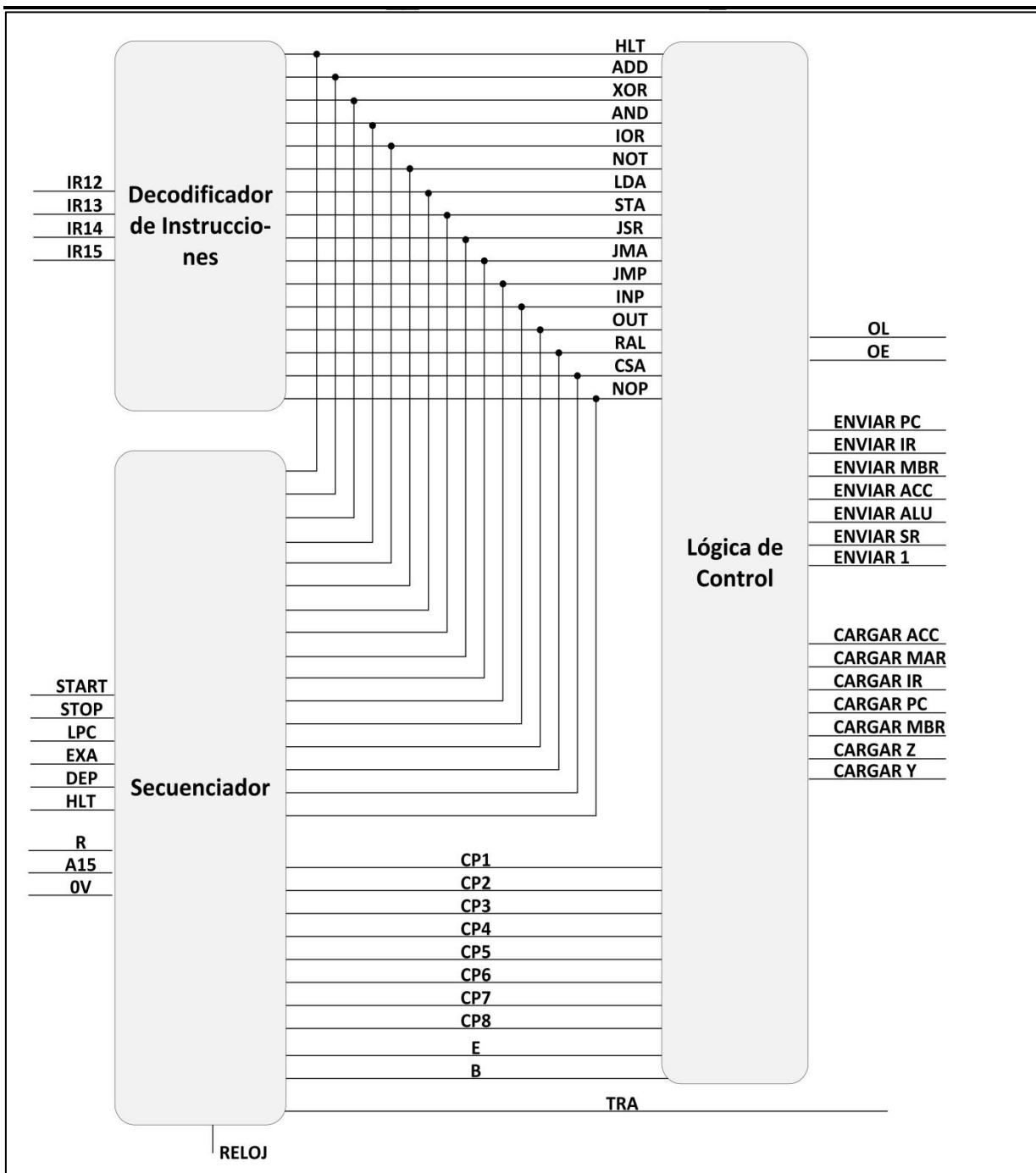
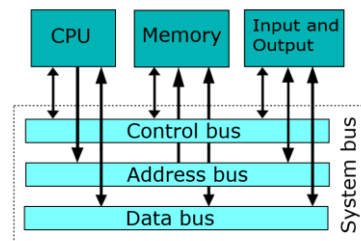
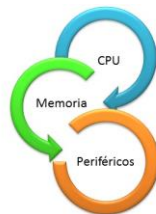
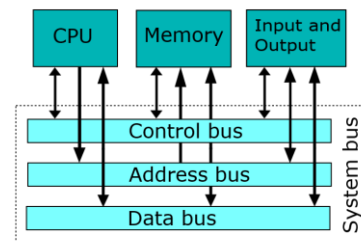
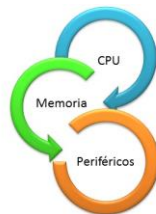


Fig. 4.9 Diagrama en bloques de la Unidad de Control Cableada



El SECUENCIADOR consta de (Figura 4.10):

- 2 (dos) biestables para definir los distintos estados mencionados:
 - **RUN** (arranque): es un biestable Set-Reset que arranca (pone en “1” su salida Q) por el botón de START o se para (pone en “0” su salida Q) por el botón de STOP, por la instrucción HALT o por la detección de un overflow aritmético.
 - **STATE** (estado): es un biestable D tiene dos condiciones (sus salidas) para determinar el ciclo de búsqueda (Fetch) o el ciclo de ejecución (Execute).
 - 1 (un) biestable relacionado con E/S:
 - **TRA** (transferencia): Es un biestable Set-Reset que se pone en “1” cuando se inicia una transferencia de E/S y se pone en “0” cuando se recibe la señal R (Ready) desde algún periférico. La E/S se discute más adelante.
 - 1 (un) biestable para tareas relacionadas con los Pulsadores EXA y DEP.
 - 1 (un) un reloj externo de 8 Mhz.
- c) **LÓGICA DE CONTROL:** Sistema Combinacional que consiste en un conjunto de compuertas que generan las señales de control.

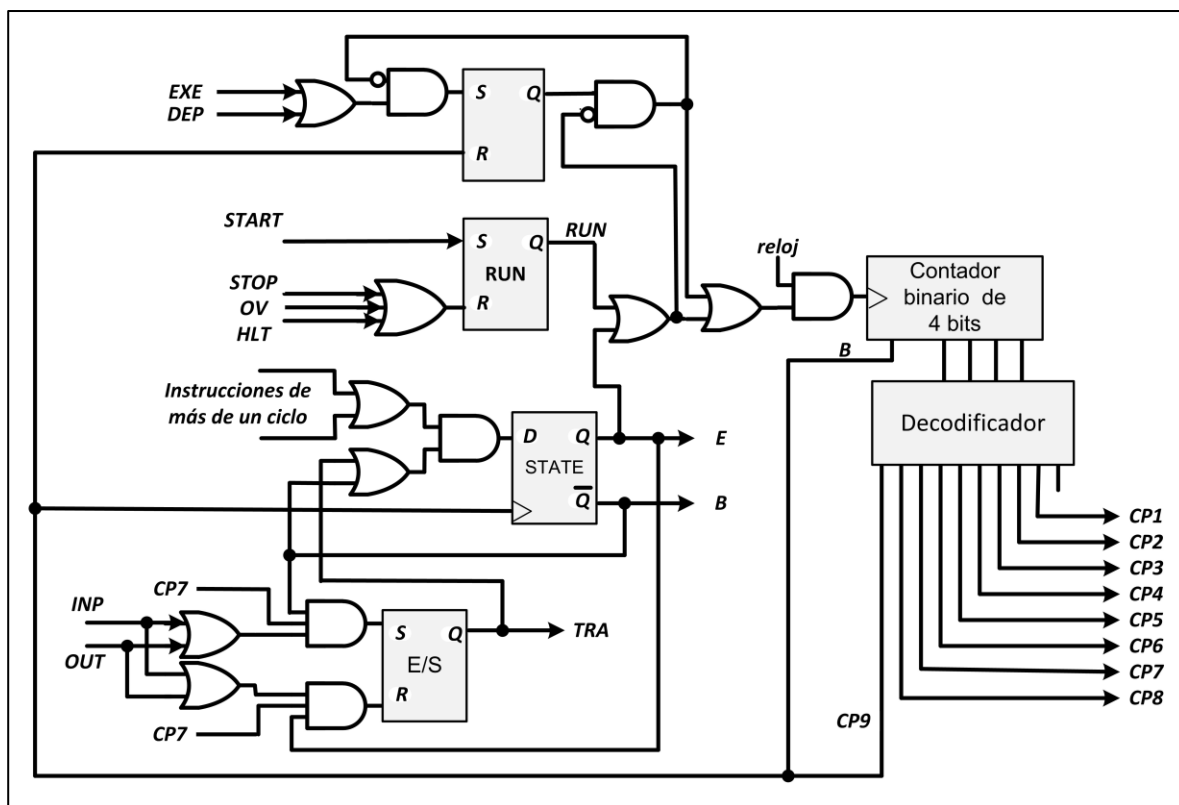
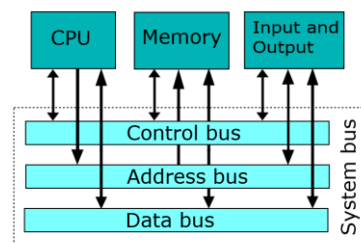
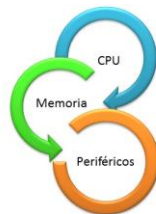


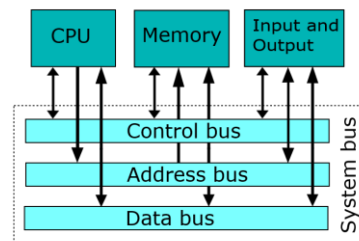
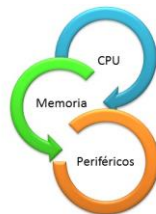
Fig. 4.10 Diagrama en bloques del secuenciador

2.6.2 Secuenciador

La señal de sincronismo es un oscilador de 8 Mhz llamado RELOJ que se divide en una secuencia de 8 pulsos, llamado CICLO DE MEMORIA, en líneas distintas separadas en tiempo por 125 ns.

Si $RUN = 1$ (máquina funcionando) el contador cuenta los pulsos de reloj, comenzando por 1 (ya que inicialmente estaba en cero). Las salidas del contador están conectadas a las entradas de un decodificador binario cuyas salidas son los pulsos de 125 ns.

Cuando la cuenta llega a 1001 (CP9), el contador se borra y comienza un nuevo CICLO DE MEMORIA. Además, este pulso CP9 se usa para sincronizar el biestable ESTADO y borrar el biestable SR sin nombre. Si la instrucción en proceso es de un ciclo $D = 0$; en el caso de instrucciones de más de un ciclo $D = 1$. De esta forma la



máquina pasa de CICLO DE BÚSQUEDA a CICLO DE EJECUCIÓN automáticamente después del CP8.

Nótese que la máquina arranca con $RUN = 1$, al presionar el pulsador START y se detiene al presionar STOP, con un OV o con la instrucción HLT, siempre y cuando ESTADO = 0 (BÚSQUEDA).

El biestable SR E/S genera la señal TRA hacia los periféricos en las condiciones que se ven en la Figura 4.10. Además, recibe la señal R (READY) desde los periféricos y se pone en cero.

Los pulsadores EXA y DEP funcionan sólo si $RUN = 0$ y ESTADO = 0, y disparan un único CICLO DE MEMORIA. En la Figura 4.11 se muestra el diagrama de tiempo del secuenciador, y aparecen los CP1 a CP8. El CP9, que no se usa en el CICLO DE MEMORIA, es de menor duración.

El Ciclo de Búsqueda

Si el biestable $RUN = 1$, arranca el reloj, el biestable de STATE = 0 está en Búsqueda y se inicia el ciclo de búsqueda de la máquina, en el cual la máquina carga la instrucción cuya dirección está en el PC, en el registro de instrucciones. Es necesario que el operador haya cargado desde la Consola un PROGRAMA en Memoria y la dirección de la primera instrucción en el PC.

La secuencia del ciclo de búsqueda se presenta en la Tabla 4.2:

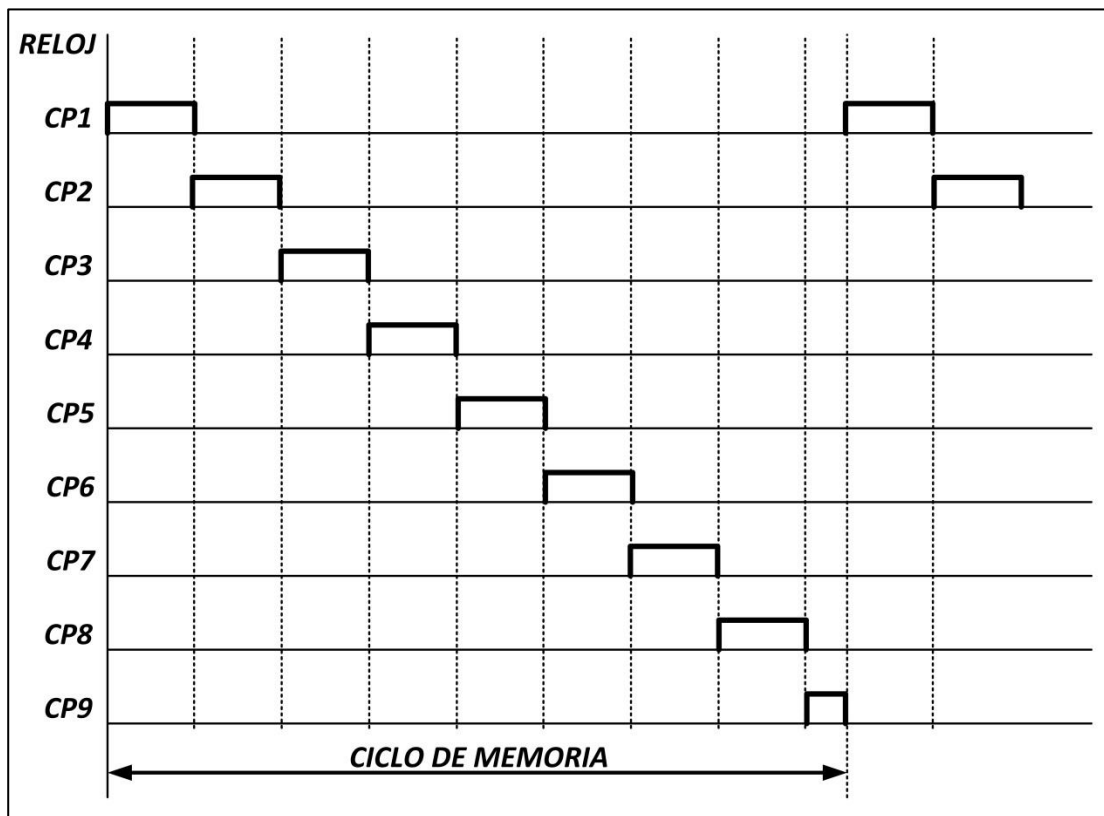
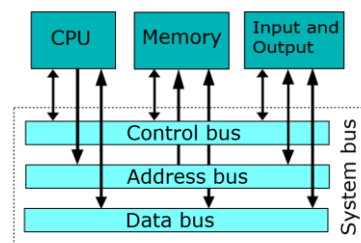
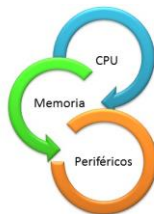
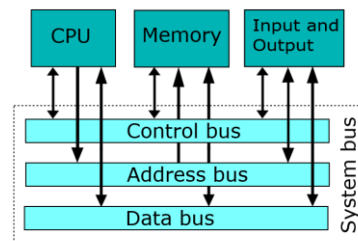
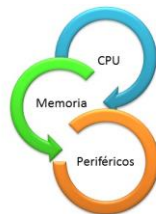


Fig. 4.11 Diagrama de tiempos del secuenciador

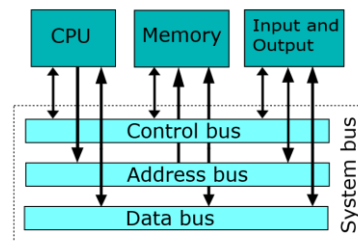
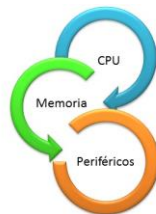


CP	ACCIÓN	COMENTARIO
1	PC → MAR PC → Z	Transfiere el PC al MAR y a Z, y comienza a leer la instrucción
2	+1 → Y	Coloca +1 en Y
3		Tiempo de espera
4	ALU → PC	Hace la suma de PC + 1 y coloca el resultado en el PC
5	M → MBR	Coloca el dato leído de la memoria en el MBR
6	MBR → IR	Transfiere el contenido del MBR al IR y comienza la decodificación de la instrucción
7		Disponible para decodificación y ejecución
8		Se define si el próximo ciclo es de búsqueda o de ejecución

Tabla 4.2 Ciclo de búsqueda

En el pulso de reloj 1 se copia el contenido del contador de programa (PC) al registro de direcciones de memoria (MAR) y al registro Z de la ALU, y se inicia el ciclo de lectura de la memoria. En el pulso de reloj 2 se coloca el número +1 en el registro Y de la ALU. En el pulso de reloj 3 esperamos que se realice la suma PC + 1. En el pulso 4 se copia el resultado de la suma (la salida de la ALU) al contador de Programa (esto incrementa el PC y está listo para indicar la próxima instrucción). En el pulso 5 se copia el dato de la memoria al registro buffer de memoria (MBR). En el pulso 6 se copia el contenido del MBR al registro de instrucción (IR). Así culmina el Ciclo de Búsqueda.

Si la instrucción en cuestión no requiere en su ejecución acceder a la memoria por un dato, se pueden utilizar los pulsos de reloj 7 y 8 para ejecutar algunas



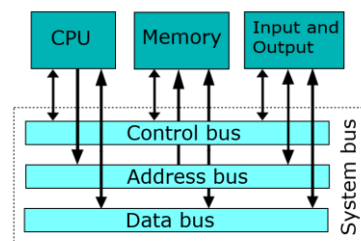
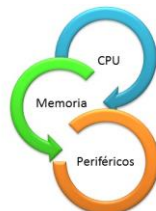
instrucciones de la BLUE. Este es el caso de las instrucciones HALT, NOP, JMP, JMA, SRJ, CSA, NOT, RAL como indica la Tabla 4.2. Las mismas finalizan en el ciclo de memoria asignado a la Búsqueda (no necesitan otro ciclo de memoria) y las llamamos instrucciones de un ciclo. Es decir, la búsqueda y ejecución de estas instrucciones se realizan en un mismo ciclo de memoria. Al finalizar el ciclo de búsqueda, empieza un nuevo ciclo que debe asignarse nuevamente a Búsqueda.

Si la instrucción en cuestión requiere en su ejecución acceder a la memoria, o si se trata de las instrucciones INP o OUT, será necesario asignar el próximo ciclo a Ejecución.

El Ciclo de Ejecución

Las instrucciones que requieren de un dato a memoria para realizar una operación lógica o matemática, o realizar una transferencia de datos hacia o desde la memoria, requieren un segundo ciclo de memoria. Estas son: LDA, STA, ADD, XOR, AND, IOR. Para acceder a memoria es necesario un segundo ciclo de memoria como se observa en la Tabla 4.3.

Pero también existen instrucciones que requieren más de un ciclo y no acceden a memoria en su ejecución. Es el caso particular de las instrucciones de E/S: INP y OUT. Utilizan nuevos ciclos para esperar al periférico involucrado. Y serán tantos ciclos como el tiempo de espera (Tabla 4.4).



CP	HALT	NOP	JMP	JMA	SRJ	CSA	NOT	RAL
7	1 → TRA	-----	-----	-----	PC → A	-----	A → Z	A → Z
8	Off → RUN	-----	IR → PC	If A ₁₅ = 1 IR → PC	IR → PC	SR → A	Z → A	2*Z → A

a)

CP	LDA	STA	ADD	XOR	AND	IOR
8	Al finalizar este pulso del ciclo de búsqueda, el biestable de ESTADO cambia a EJECUCIÓN					
1	IR → MAR; OL	IR → MAR	IR → MAR; OL	IR → MAR; OL	IR → MAR; OL	IR → MAR; OL
2	-----	A → MBR; OE	A → Z	A → Z	A → Z	A → Z
3	-----	-----	-----	-----	-----	-----
4	-----	-----	-----	-----	-----	-----
5	Mem → MBR	-----	Mem → MBR	Mem → MBR	Mem → MBR	Mem → MBR
6	MBR → A	-----	MBR → Y	MBR → Y	MBR → Y	MBR → Y
7	-----	-----	-----	-----	-----	-----
8	STATE → F	STATE → F	SUM STATE → A STATE → F	XOR STATE → A STATE → F	AND STATE → A STATE → F	OR STATE → A STATE → F

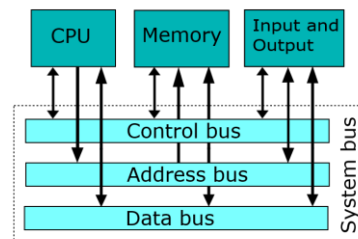
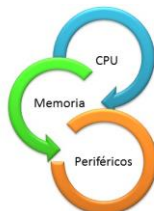
b)

Tabla 4.3 Ejecución de las instrucciones de uno y dos ciclos



UTN
virtual
MENDOZA

UTN
FACULTAD
REGIONAL
MENDOZA
UNIVERSIDAD TECNOLÓGICA NACIONAL



CP	INPUT	OUTPUT	COMENTARIO
7	1 → TRA	1 → TRA	Los bits IR ₅₋₀ seleccionan el dispositivo de E/S
8	E → STATE	E → STATE	Fin ciclo de búsqueda y comienza el de ejecución
1	-----	-----	
2	-----	-----	
3	-----	-----	
4	-----	-----	
5	-----	-----	
6	-----	-----	
7	Si R = 1 Inp → A 0 → TRA	Si R = 1 0 → TRA	Si el Flag R=1 en INP se copian los datos del periférico a A, y en ambos se pone TRA=0
8	Si TRA = 0 F → STATE Si TRA = 1 E → STATE	Si TRA = 0 F → STATE Si TRA = 1 E → STATE	Si se completó la transferencia se inicia un nuevo ciclo de búsqueda Si NO se completó la transferencia se inicia un nuevo ciclo de ejecución

Tabla 4.4 Ciclo de búsqueda de las instrucciones INP y OUT

En la Figura 4.12 se puede ver el hardware relacionado con la Entrada/Salida con cierto detalle.

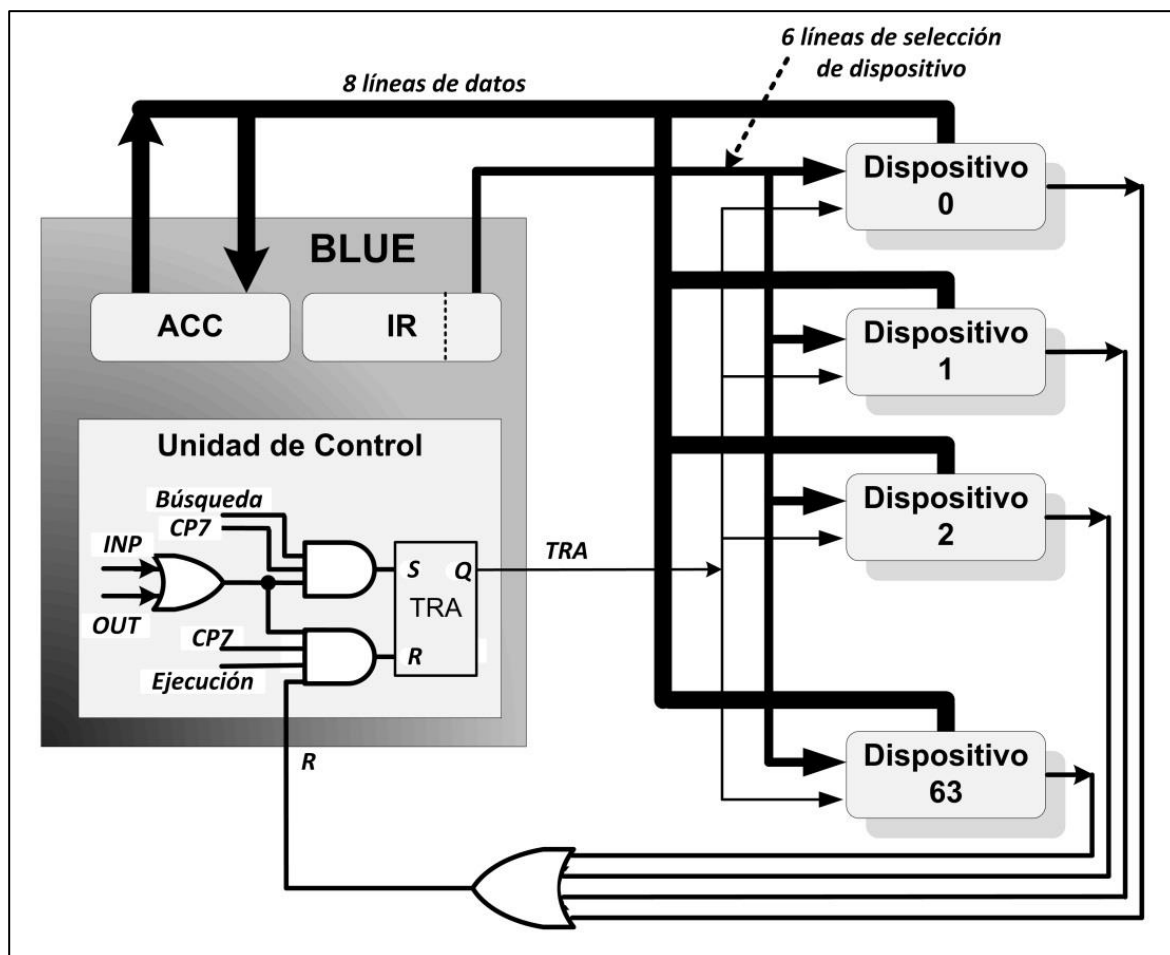
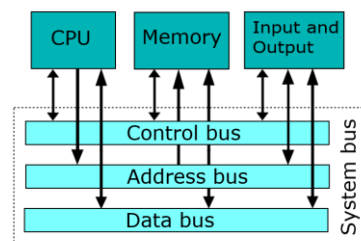
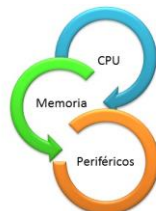


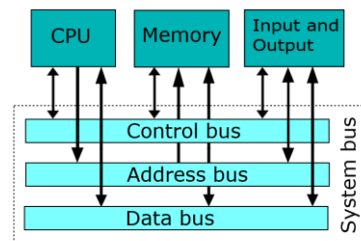
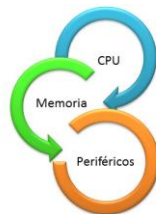
Fig. 4.12 Lógica para el funcionamiento de las E/S

2.6.3 Lógica de Control

Como se observa en la Figura 4.13, las entradas a este bloque combinacional son los códigos de operación decodificados, los pulsos de reloj y el estado de Búsqueda o Ejecución. Las salidas deben ser las órdenes concretas que emite la unidad de control. A fin de aclarar en qué consiste el hardware dentro de este bloque LÓGICA DE CONTROL veamos un ejemplo:

¿En qué casos la UC debe emitir la orden CARGAR MAR?

- 1) Si la máquina está parada y se presiona el pulsador EXA o DEP durante el CP1,
- 2) Si la máquina está funcionando, y está en Búsqueda y durante el CP1, o



- 3) Si la máquina está funcionando y está en Ejecución de las instrucciones LDA, STA, ADD, IOR, AND o XOR durante el CP1.

Entonces el circuito de control será, teniendo en cuenta que si la máquina esta parada el biestable Estado = 0 (esto significa que $B = 1$ y $E = 0$) y se producirá una única secuencia de pulsos CP1 a CP8 sólo si el operador presionó EXA o DEP.

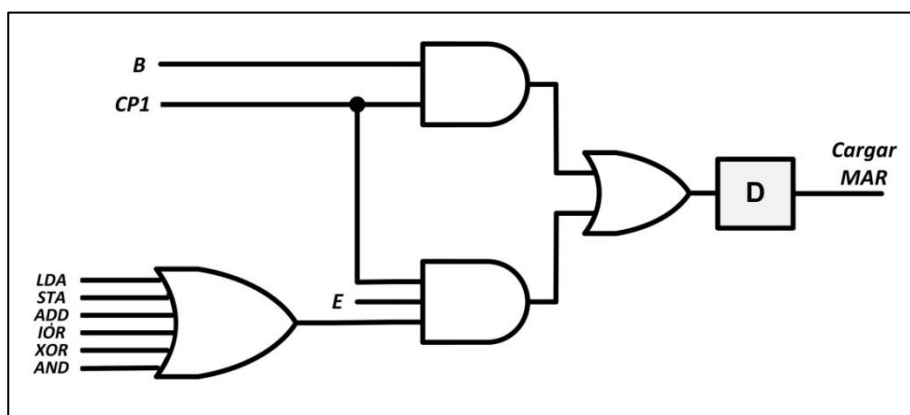


Fig. 4.13 Vista parcial de la lógica control

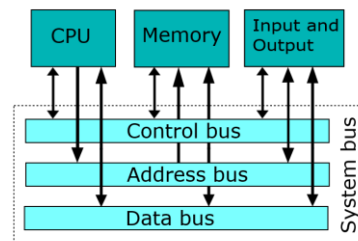
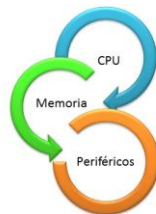
El recuadro D indica un retardo de 50 ns a fin de permitir que se establezca el BUS como se verá posteriormente. Circuitos similares al planteado se corresponderán con todas las señales emitidas por la Unidad de Control.

2.7 UNIDAD DE CONTROL MICROPROGRAMADA

El esquema general de la máquina elemental con unidad de control microprogramada o simplemente microprogramada, se observa en la Figura 4.14.

Obsérvese que lo único que ha cambiado en la máquina elemental es la Unidad de Control que, por su parecido a la arquitectura de von Neumman, se la ha llamado MÁQUINA INTERIOR.

La búsqueda y ejecución de cada macroinstrucción son realizadas por los microprogramas residentes en la micro-ROM (al contenido de la ROM, es decir, los microprogramas y la propia ROM se le llama FIRMWARE). La dirección de la primera microinstrucción a ejecutar, es proporcionada por el código de operación de la macroinstrucción, es decir, será alguna de las 16 primeras posiciones.



Cada microinstrucción está compuesta de 45 bits, divididos en seis campos: Acción, Test, Envíe, Reciba, Falso, y Éxito.

El significado de cada uno puede verse en la Tabla 4.4. El campo Acción está relacionado con las órdenes que debe dar la Unidad de Control (leer la memoria, escribir la memoria, etc.), el campo Test se relaciona con la necesidad de chequear el estado de la máquina en un momento dado (Bit 15 del acumulador, señal de overflow, etc.). Los campos Envíe y Reciba tienen que ver con enviar los contenidos de los registros al bus o levantarlos del mismo. Por último, los campos falso y éxito están relacionados con el resultado del chequeo indicado por el campo Test y definen la próxima microinstrucción a ejecutar. Nótese que la máquina interior no posee contador de programa. En la Tabla 4.5 puede verse el contenido de la ROM, es decir, los microprogramas correspondientes a cada instrucción.

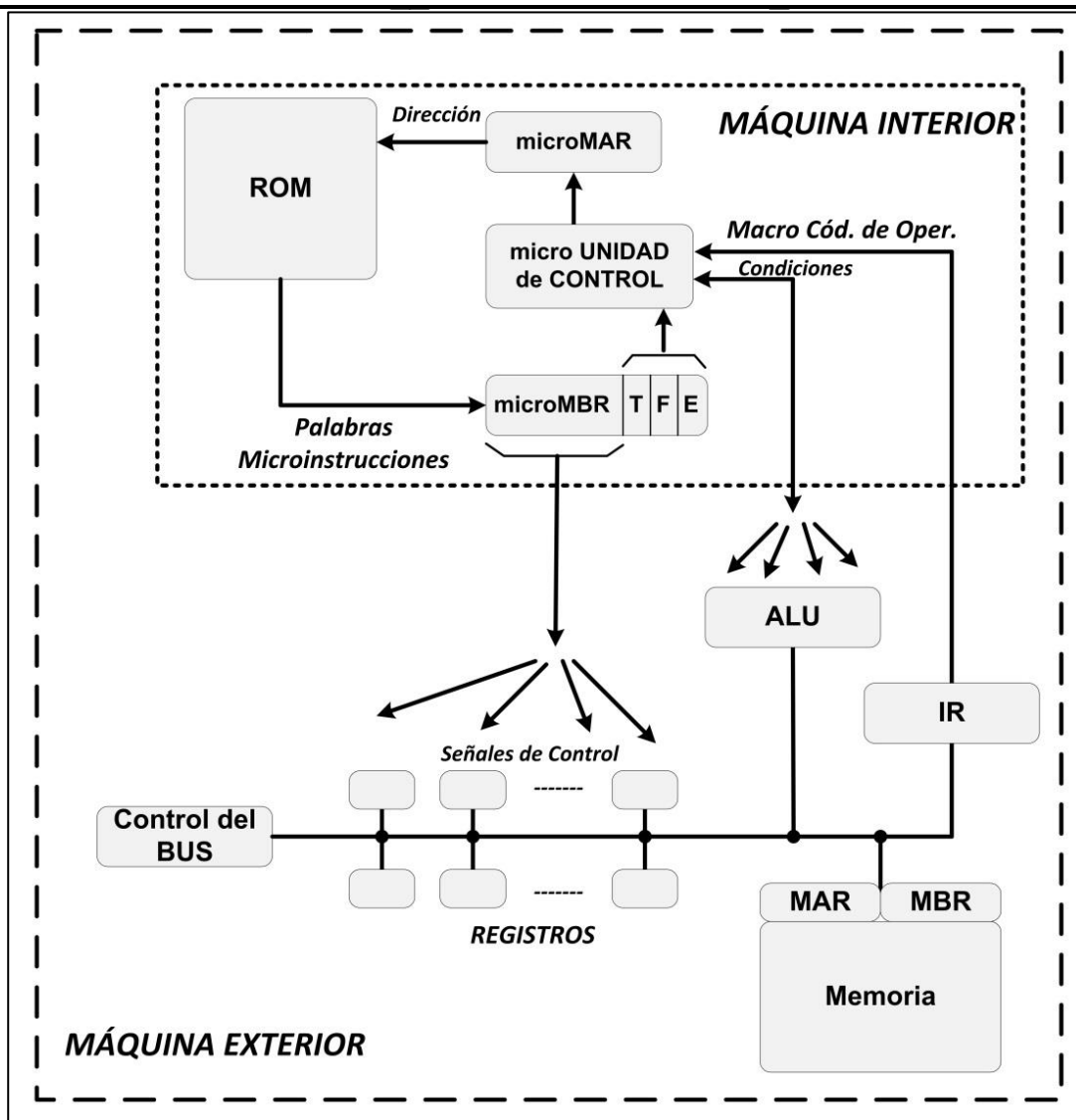
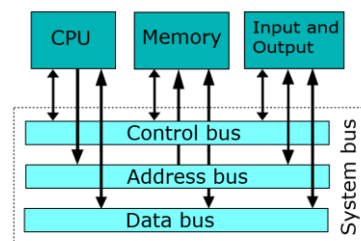
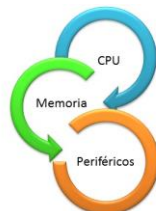
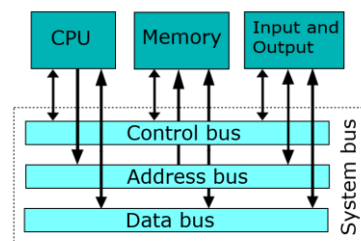
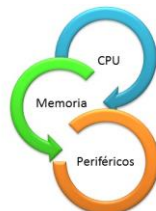


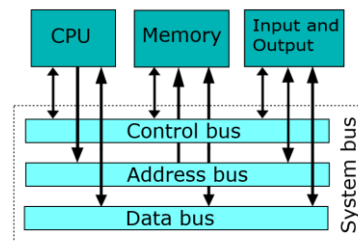
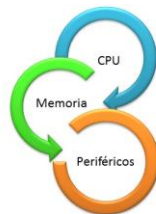
Figura 4.14 Máquina elemental con unidad de control microprogramada



MICROINSTRUCCIÓN DE LA MÁQUINA MICROPROGRAMADA

Acción (5)	Test (4)	Enviar (13)	Recibir (7)	Falso (8)	Éxito (8)
Campo ACCIÓN: (5 BITS): Realiza una de las siguientes acciones:					
1º bit	Inicie ciclo de lectura (LEER)				
2º bit	Inicie ciclo de escritura (ESCRIBIR)				
3º bit	TRA = 1				
4º bit	TRA = 0				
5º bit	BOOC				
Campo ENVIAR: (13 BITS): Envía al BUS uno de los siguientes Registros:					
1º bit	Acumulador (A)				
2º bit	MBR				
3º bit	PC				
4º bit	SR				
5º bit	+1				
6º bit	Campo de direccionamiento de IR				
7º bit	Líneas de entrada de datos (LED)				
8º bit	SUM				
9º bit	OR				
10º bit	XOR				
11º bit	AND				
12º bit	2 * Z				
13º bit	C ₁ (z)				
Campo RECIBIR: (7 BITS): Carga uno de los siguientes Registros desde el BUS:					
1º bit	Acumulador (A)				
2º bit	MBR				
3º bit	PC				
4º bit	Z				
5º bit	IR				
6º bit	Y				
7º bit	MAR				
Campo TEST: (4 bits): Si la condición especificada es acertada, toma la próxima microinstrucción desde la dirección especificada por el campo ÉXITO. Sino la toma desde el campo FALSO. Las condiciones son las siguientes:					
0000	No realizar TEST				
0001	A ₁₅ = 1 Acumulador negativo)				
0010	R = 1 (Dispositivo de E/S listo)				
0011	Memoria LIBRE (la memoria dispone de una línea que indica la finalización de una lectura o una escritura)				
0100	Botón STOP				
0101	Botón START				
0110	Botón Load PC				
0111	Botón EXAMINE				
1000	Botón Deposit				
1001	OVERFLOW				
1010 al 1111	NO DEFINIDOS				

Tabla 4.5 Formato de la microinstrucción de la máquina elemental microprogramada



El circuito de la UNIDAD de CONTROL MICROPROGRAMADA se observa en la Figura 4.15. Obsérvese que el contenido del micro-MAR (los 8 biestables D de la Figura) puede tener cuatro orígenes:

- Contenido del campo FALSO, en caso que la salida del multiplexor sea cero.
- Contenido del campo ÉXITO, en caso que la salida del multiplexor sea 1.
- El código de operación de la macroinstrucción residente en el registro de instrucciones, en el caso que el bit 5 del campo acción sea 1
- Cero, en el caso que RESET sea 1 (esta señal proviene del botón Master Reset en la consola del operador).

Los dos primeros casos se dan cuando la Unidad de Control está ejecutando un microprograma, ya sea correspondiente a una macroinstrucción o al ciclo de búsqueda.

En la Tabla 4.6 se muestra el contenido de la ROM (256 x 45). A modo de ejemplo, se describirá el ciclo de búsqueda en nuestra nueva máquina. El microprograma correspondiente comienza en la dirección rotulada RNI. El campo Acción no tiene contenido indicando que ninguna acción es necesaria, El campo Envíe envía el PC al bus, el campo Reciba carga el MAR y el Z desde el bus, el campo teste verifica si alguien ha apretado el botón STOP de la consola, si está apretado la próxima microinstrucción es la que está en la dirección rotulada HALT, si no está apretado, la próxima microinstrucción es RNI1. De RNI1 a RNI5, se realiza el incremento del PC y la carga del IR con el contenido de la posición de memoria direccionada. La RNI6 da la orden de cargar el micro-MAR con el código de operación de la macroinstrucción ya almacenada en el IR, esto lo hace con el bit 5 en 1 (ver el circuito de control del micro-MAR).

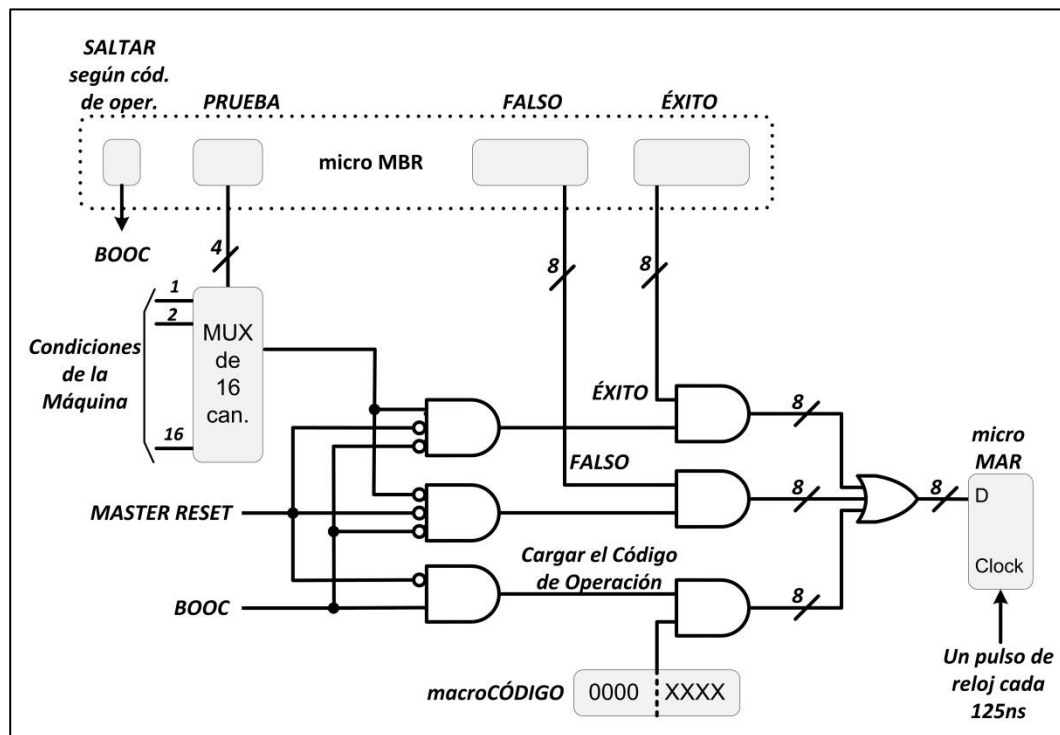
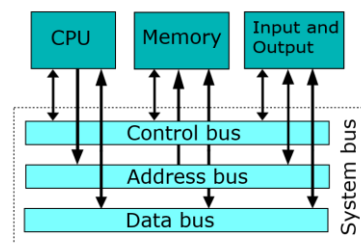
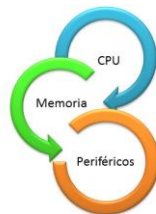
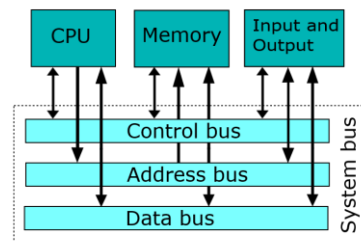
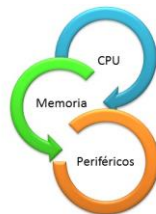
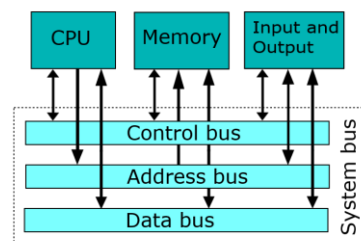
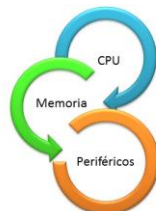


Fig. 4.15 Unidad de control microprogramada

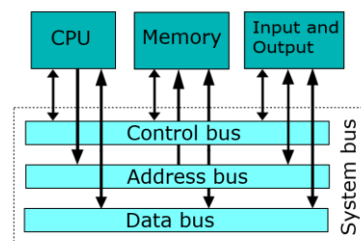
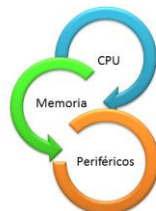


Dirección	Acción	Envíe	Recibe	Test	Falso	Éxito
0	----	----	----	----	HALT	----
1	----	----	----	----	ADD	----
2	----	----	----	----	XOR	----
3	----	----	----	----	AND	----
4	----	----	----	----	IOR	----
5	----	----	----	----	NOT	----
6	----	----	----	----	LDA	----
7	----	----	----	----	STA	----
10	----	----	----	----	SRJ	----
11	----	----	----	----	JMA	----
12	----	----	----	----	JMP	----
13	----	----	----	----	INP	----
14	----	----	----	----	OUT	----
15	----	----	----	----	RAL	----
16	----	----	----	----	CSA	----
17	----	----	----	----	NOP	----
Microprograma de Búsqueda						
RNI	LEER	PC	MAR, Z	STOP	RNI 1	HALT
RNI 1	----	+1	Y	----	RNI 2	----
RNI 2	----	----	----	----	RNI 3	----
RNI 3	----	SUM	PC	LIBRE	RNI 4	RNI 5
RNI 4	----	----	----	LIBRE	RNI 4	RNI 5
RNI 5	----	MBR	IR	----	RNI 6	----
RNI 6	BOOC	----	----	----	----	----
Microprograma de la instrucción HALT						
HALT	TRA = 0	----	----	START	H 1	RNI
H 1	----	----	----	EXAM	H 2	EXAM
H 2	----	----	----	DEPOSIT	H 3	DEPOSIT
H 3	----	----	----	Load PC	HALT	LPC
Microprograma del Botón EXAMINAR						
EXA	LEER	PC	MAR, Z	----	E 1	----
E 1	----	+1	Y	----	E 2	----
E 2	----	----	----	----	E 3	----
E 3	----	SUM	PC	LIBRE	E 4	E 5
E 4	----	----	----	LIBRE	E 4	E 5
E 5	----	MBR	IR	----	E 6	----
E 6	----	----	----	EXAM	H 6	E 6

a)



Dirección	Acción	Envíe	Recibe	Test	Falso	Éxito
Microprograma del Botón DEPOSITAR						
DEP	----	PC	MAR, Z	----	D 1	----
D 2	----	+ 1	Y	----	D 2	----
D 2	----	SUM	PC	----	D 3	----
D 3	ESCRIBIR	SR	MBR	----	D 4	----
D 4	----	----	----	LIBRE	D 4	D 5
D 5	----	----	----	DEPOSIT	H 3	D 5
Microprograma del Botón CARGUE PC						
LPC	----	SR	PC	----	L 1	----
L 1	----	----	----	LOAD PC	HALT	L 1
Microprograma de la instrucción NOP						
NOP	----	----	----	----	RNI	----
Microprograma de la instrucción CSA						
CSA	----	SR	A	----	RNI	----
Microprograma de la instrucción RAL						
RAL	----	A	Z	----	RAL 1	----
RAL 1	----	2*Z	A	----	RAL 2	----
RAL 2	----	----	----	----	RNI	----
Microprograma de la instrucción JMP						
JMP	----	IR	PC	----	RNI	----
Microprograma de la instrucción JMA						
JMA	----	----	----	A 15	RNI	JMP
Microprograma de la instrucción SRJ						
SRJ	----	----	PC	A	JMP	----
Microprograma de la instrucción NOT						
NOT	----	A	Z	----	NOT 1	----
NOT 1	----	C _i (Z)	A	----	NOT 2	----
NOT 2	----	----	----	----	RNI	----
Microprograma de la instrucción INP						
INP	TRA = 1	----	----	R = 1	INP	INP 1
INP 1	----	LED	A	----	INP 2	----
INP 2	TRA = 0	----	----	----	RNI	----
Microprograma de la instrucción OUT						
OUT	TRA = 1	----	----	R = 1	OUT	OUT 1
OUT 1	TRA = 0	----	----	----	RNI	----
b)						

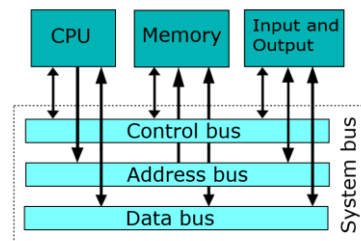
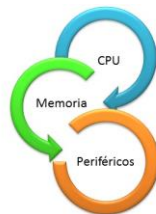


Dirección	Acción	Envíe	Recibe	Test	Falso	Éxito
Microprograma de la instrucción LDA						
LDA	LEER	IR	MAR	----	LDA 1	----
LDA 1	----	----	----	LIBRE	LDA 1	LDA 2
LDA 2	----	MBR	A	----	RNI	----
Microprograma de la instrucción STA						
STA	----	IR	MAR	----	STA 1	----
STA 1	ESCRIBIR	A	MBR	----	STA 2	----
STA 2	----	----	----	LIBRE	STA 2	RNI
Microprograma de la instrucción ADD						
ADD	LEER	IR	MAR	----	ADD 1	----
ADD 1	----	A	Z	LIBRE	ADD 2	ADD 3
ADD 2	----	----	----	LIBRE	ADD 2	ADD 3
ADD 3	----	MBR	Y	----	ADD 4	----
ADD 4	----	----	----	----	ADD 5	----
ADD 5	----	SUM	A	----	RNI	----
Microprograma de la instrucción IOR						
IOR	LEER	IR	MAR	----	IOR 1	----
IOR 1	----	A	Z	LIBRE	IOR 2	IOR 3
IOR 2	----	----	----	LIBRE	IOR 2	IOR 3
IOR 3	----	MBR	Y	----	IOR 4	----
IOR 4	----	----	----	----	IOR 5	----
IOR 5	----	OR	A	----	RNI	----

c)

Tabla 4.6 Contenido de la ROM (256 x 45)

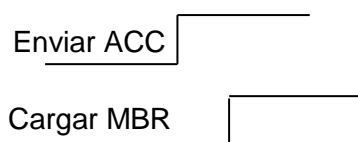
Es conveniente realizar el análisis de todos los microprogramas a fin de comprender el funcionamiento completo. Al final de la Guía Didáctica se discuten las ventajas y desventajas de las máquinas microprogramadas que, debido a la facilidad que da el firmware, poseen gran cantidad de instrucciones. En contraposición, las máquinas no microprogramadas poseen, entre otras características, pocas instrucciones.



2.8 BUS EN LA MÁQUINA ELEMENTAL

Si analizamos las órdenes que emite la UC (Cableada o Microprogramada), podemos concluir que la gran mayoría implica una transferencia entre registros. Con exactitud son 20 señales entre enviar al BUS y cargar desde el BUS. Sólo 4 órdenes no implican transferencias: LEER, ESCRIBIR, TRA = 0 y TRA = 1. Gran parte de la actividad de la máquina se resuelve transfiriendo registros.

Por lo mencionado, los registros deben estar eficazmente interconectados y esto puede lograrse con la arquitectura de BUS COMÚN. Este término se refiere a que existirá un único BUS que interconecta a las distintas partes de la Máquina. Este BUS COMÚN puede transportar un dato (16 bits), una instrucción (16 bits) o una dirección (12 bits) en distintos momentos, razón por la cual se lo llama bus multiplexado. Las señales de control son diseminadas por la Máquina por un BUS especial a ese fin, que llamamos BUS de CONTROL. En la Figura 4.16 se observa un circuito que resuelve los requerimientos y las señales de control.



La señal Cargar MBR debe estar retardada un tiempo mayor al de respuesta del biestable y las compuertas AND y OR

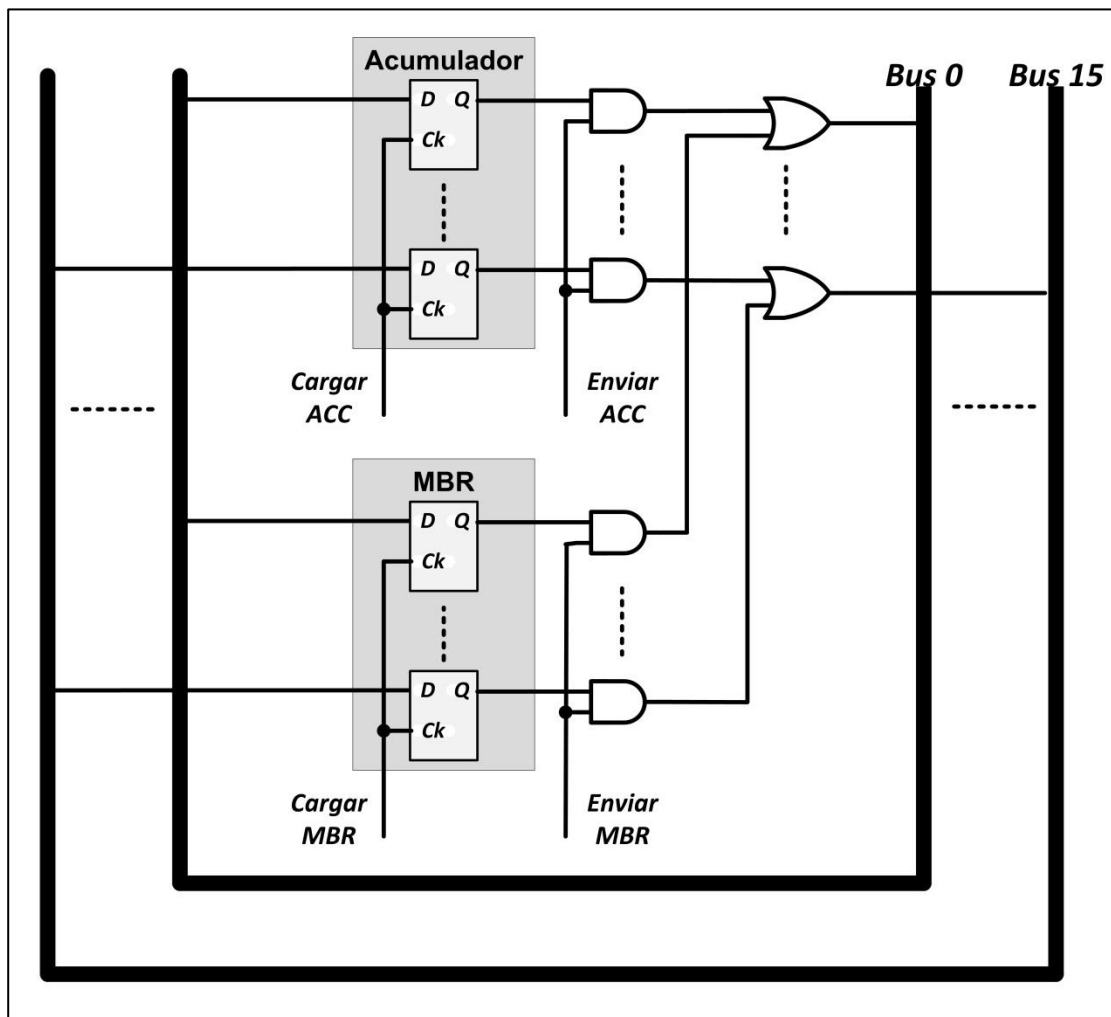
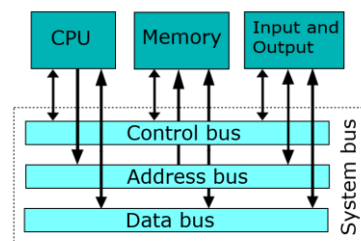
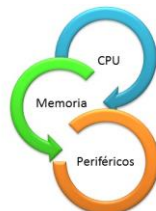


Figura 4.16 Circuito que resuelve los requerimientos y las señales de control.

Obsérvese que sólo se debe enviar un registro al BUS. Sin embargo, es posible cargar más de un registro desde el BUS simultáneamente, un ejemplo de esto es la orden:

ENVÍE PC, CARGUE MAR y Z

La conexión del Registro MBR al BUS requiere un circuito adicional por cuanto este Registro también está conectado la memoria. La figura 4.17 indica esta conexión

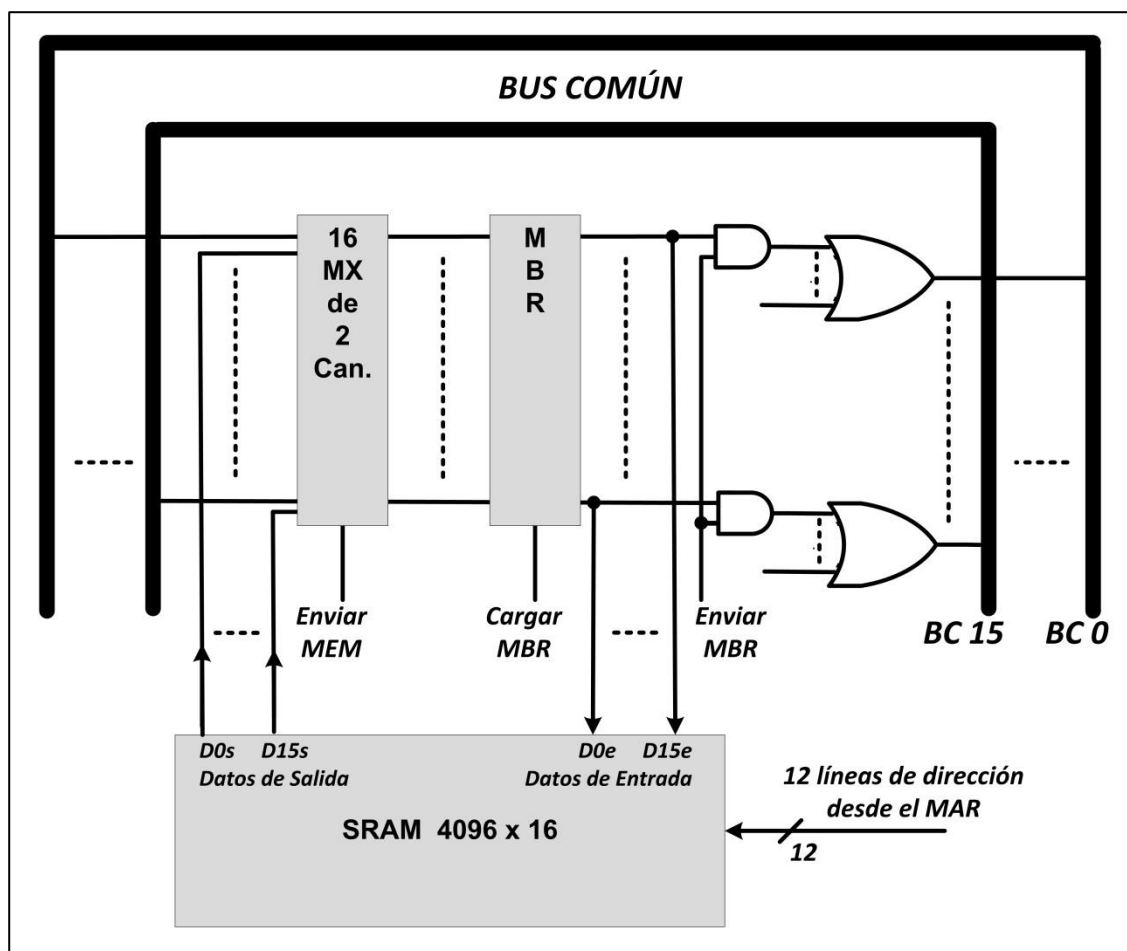
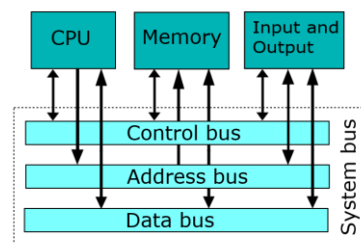
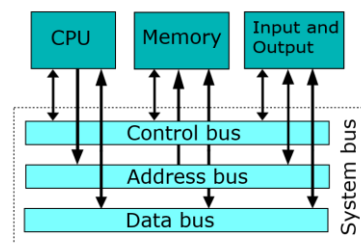
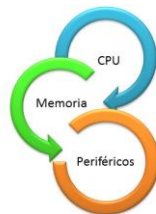


Figura 4.17 Circuito asociado al Registro MBR.

2.9 UNIDAD ARITMÉTICA Y LÓGICA

La ALU de la Máquina Elemental realiza las operaciones AND, OR, IOR, XOR y ADD sobre los operandos contenidos en los registros Z e Y. Las operaciones lógicas se realizan bit a bit y la suma aritmética se realiza con convenio de Complemento a 2 (el acarreo se desprecia). Adicionalmente, la ALU realiza la operación NOT (C1) y RAL (rotación de bits hacia la izquierda) sobre el contenido de Z.



Las operaciones mencionadas las realiza simultáneamente y permanentemente, es decir, que en todo momento está calculando los resultados sobre el contenido de los registros Z e Y. La unidad de control sólo tiene que cargar los registros Z e Y con los operandos y seleccionar cual es la salida que necesita. El diagrama en bloques de la ALU se observa en la Figura 4.18.

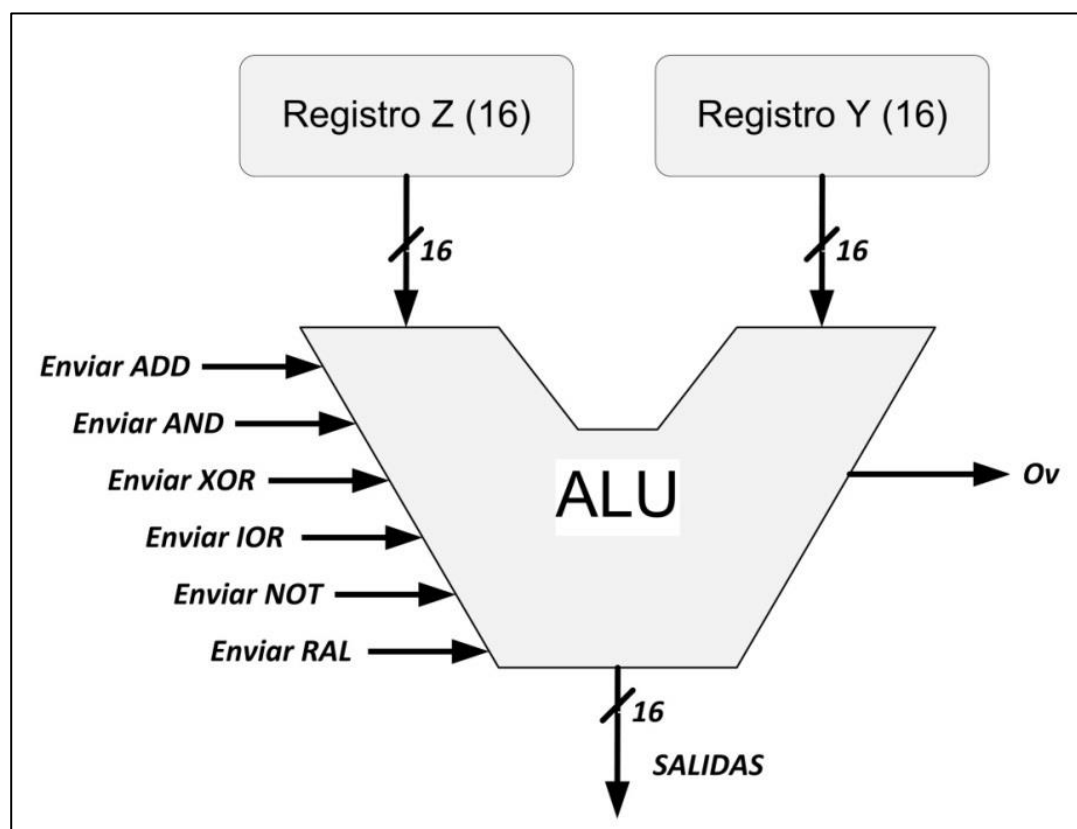
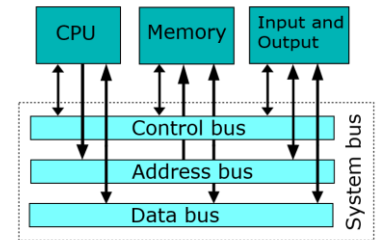
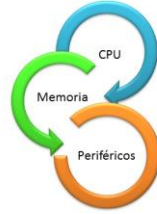


Fig. 4.18 Diagrama en bloques de la unidad aritmética y lógica

Se recomienda al lector realizar el circuito de la ALU para ejercitar lo aprendido. La ALU tendrá:

- 16 Compuertas AND,
- 16 Compuertas XOR,
- 16 Compuertas IOR,
- 16 Compuertas NOT,
- 1 Sumador de 16 bits con circuito de rebasamiento (OV) sin acarreo (C2), y
- Compuertas AND y OR para implementar la selección de la salida.



Es conveniente mencionar que el tiempo que tarda la ALU para realizar la suma aritmética (ADD), es mayor que el necesario para las operaciones lógicas, ya que para estas últimas requieren menos hardware. Para los ciclos de máquina se ha supuesto que la ALU tarda para sumar (ADD), como máximo, 200 ns y para el resto de las operaciones 80 ns.

Por lo tanto, cuando la ALU suma, la UC debería esperar un pulso de reloj (para el caso de UC Cableada) o una microinstrucción (para el caso de UC Microprogramada). Observe las Figuras 4.10 y 4.15.