

UNA MAQUINA ELEMENTAL

Extraído de: Caxton Foster "Computer Architecture" Ed 2da , Van Nostrand Reinhold

Traducido por: Gustavo Mercado para la cátedra Técnicas Digitales II UTN FRM (1994)

INTRODUCCION

"Cuando desees producir un resultado por medio de un instrumento, no te permitas complicarlo"
(LEONARDO DA VINCI)

En este trabajo se va a describir una computadora muy simple, que se vendería por unos 1000 dólares. 8 años atrás, yo estimé su costo en 1000 dólares, pero si uno fuera a producir suficientes unidades como para recuperar el alto costo inicial de la integración a gran escala, se podría fabricar una máquina de 3 o 4 chips por unos 100 dólares. Primero se describe como la máquina aparece al los ojos del programador, y luego examinaremos cada una de sus partes y veremos como está hecha. Los lectores que están familiarizados con los aspectos básicos respecto al diseño de máquinas, probablemente se saltarán muchos de los temas. Los lectores novicios deberán estudiar este capítulo con el máximo detenimiento, puesto que será el único en el que se presente un diseño completo.

DESCRIPCION PARA EL PROGRAMADOR

BLUE¹ es una computadora digital automática, binaria, de complemento a dos, programa almacenado y longitud de palabra fija. Puede direccionar una memoria central de hasta 4096 celdas. Cada celda contiene 16 bit y el tiempo de acceso de memoria es de un micro segundo. Cuando se la usa para almacenar datos estas celdas son tratadas como palabras de dato entero de 15 bit más el signo. Cuando se las usa para almacenar instrucciones, se las trata como un campo de código operación de 4 bit seguida por un campo de dirección de 12 bit (ver Fig. 1). Existe un Acumulador (AC - Accumulator) de 16 bit, un Registro de Instrucciones (IR - Instruction Register) de 16 bit, y un Contador de Programa (PC - Program Counter) de 12 bit. El contenido de los registros mencionados se exhibe en una consola con pequeñas luces que están encendidas para el 1, y apagadas para el 0.

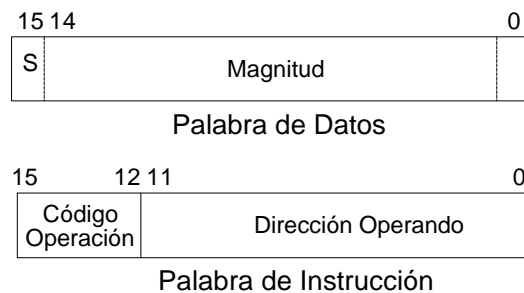


Fig. 1: Formato de las palabras de datos e instrucciones.

También en la consola hay un Registro de Llaves (SW - Switch Register) de 16 bits que el operador puede manipular, Teclas de Arranque (START), Detención (STOP), Cargar el Contador de Programa (LOAD PC, Examinar (EXAMINE), Depositar (DEPOSIT) y un botón de Borrado Total (MASTER CLEAR). Los botones de arranque y de detención tienen un significado obvio. El botón de Cargar el Contador de Programa hace que los 12 bits del registro de llaves se copien en el contador de programa. El botón de Examinar hace que el contenido de la celda señalada por el contador de programa se cargue en el registro de instrucciones para ser examinado. El botón Depositar hace que el número de 16 bit que está en el registro de llaves se almacene en la celda de memoria señalada por el contador de programas. Tanto las llaves de Examinar y Depositar hacen que el contador de programa se incremente en uno al completar su ciclo. El botón de borrado total genera una señal que pone la máquina y sus periféricos en un estado inicial de alistamiento, lista para empezar a trabajar. Además de los tres registros dispuestos en la consola, existe un Registro de Direcciones de Memoria (MAR - Memory Address Register) de 12 bits, un Registro Buffer de Memoria (MBR - Memory Buffer Register) de 16 bits y los registros "Y" y "Z", a los que el programador no puede acceder.

¹ BLUE (azul) es el color del gabinete de la máquina

Los cuatro bits del código de operación proveen 16 instrucciones distintas. Las presentamos en orden, dando primero su equivalente numérico octal, luego una descripción mnemónico de 3 letras y finalmente una breve descripción de sus efectos. La Fig. .2 muestra una vista general de la máquina.

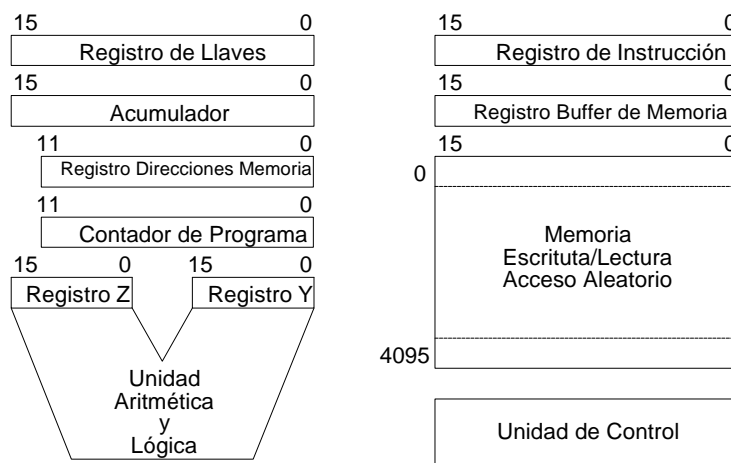


Fig. 2: Una vista general de la BLUE.

EL JUEGO DE INSTRUCCIONES

Ahora se presentan las 16 instrucciones de que dispone la BLUE. En general, el código numérico que representa a la instrucción, en numeración octal, fue elegido al azar. Las dos instrucciones HLT y NOP, a las que se asignan las combinaciones 0000 (00 octal) y 1111 (17 octal) son la excepción. Esto fue hecho de modo que cualquier intento de ejecutar una palabra que contenga todos ceros, detenga la máquina. Así, un salto ocasional dentro de un área que haya sido borrado a cero sería detectada al instante. De igual modo, tratar de ejecutar una palabra dato que este compuesta solo de unos (condición común de la palabra de detención) haría que, si no se detiene la máquina, al menos no se produzcan alteraciones.

A continuación la lista de instrucciones:

Código Máquina	Mnemónico	Operando	Descripción	
00	HLT	XXXX	Esta instrucción hace que la computadora se detenga. Apretar el botón START en la consola hará que la máquina arranque de nuevo; comenzando con la instrucción siguiente. El campo del dirección XXXX es ignorado.	
01	ADD	XXXX	Los contenidos de XXXX se agregan a los contenidos del acumulador, y la suma se introduce en el acumulador. Si la suma resultante es mayor que $2^{15}-1$ o menor que -2^{15} la máquina se detiene. El contenido de XXXX no se cambia ni se altera.	SUMA ARITMÉTICA $AC \leftarrow (XXXX) + (AC)$
02	XOR	XXXX	El contenido de la dirección XXXX, efectúa la máquina una OR exclusiva bit a bit entre el contenido de XXXX y el contenido del acumulador, el resultado de la operación reemplaza a los contenidos anteriores del acumulador. El contenido de XXXX no se altera.	O EXCLUSIVA LÓGICA $AC \leftarrow (XXXX) \oplus (AC)$
03	AND	XXXX	El computador realiza una operación AND bit a bit entre los contenidos de la dirección XXXX y los contenidos del acumulador. El resultado de la operación es puesto en el acumulador y los contenidos de la dirección XXXX no son alterados.	Y LÓGICA $AC \leftarrow (XXXX) \cdot (AC)$

04	IOR	XXXX	Se realiza una operación OR entre los contenidos de la dirección XXXX y los contenidos del acumulador y el resultado se guarda en el acumulador, los contenidos de XXXX no son cambiados.	O LÓGICA $AC \leftarrow (XXXX) + (AC)$
05	NOT	XXXX	Cada bit del contenido del acumulador es reemplazado por su complemento lógico. El dominio de la dirección XXXX es ignorado.	NO LÓGICA $AC \leftarrow \text{NOT}(AC)$
06	LDA	XXXX	Los contenidos de XXXX se copian en el acumulador. Los anteriores contenidos del mismo se pierden. Los contenidos de XXXX no se alteran.	CARGAR ACUMULADOR $AC \leftarrow (XXXX)$
07	STA	XXXX	Los contenidos del acumulador se copian en XXXX. El anterior contenido de XXXX se pierde. Los contenidos del acumulador no se alteran.	ALMACENAR ACUMULADOR $XXXX \leftarrow (AC)$
10	SRJ	XXXX	Los contenidos del contador de programa (esto es la instrucción actual más uno) se copian en los últimos 12 bits del acumulador y los 4 bits restantes se ponen a cero luego el número XXXX se copia en el contador de programa de modo que la próxima instrucción será tomada desde la ubicación XXXX. Esta instrucción se usa para saltar a una subrutina.	SALTO A SUBRUTINA $AC \langle 11..0 \rangle \leftarrow (PC) + 1$ $AC \langle 15..12 \rangle \leftarrow 0$ $PC \leftarrow XXXX$
11	JMA	XXXX	Si el bit de signo del acumulador es 1 (acumulador negativo) el número XXXX es ubicado en el contador de programa, y la siguiente instrucción se toma desde la ubicación XXXX. Si el bit de signo del acumulador es 0 (acumulador cero o positivo), esta instrucción no hace nada y la próxima a ser ejecutada se toma de la ubicación actual más uno.	SALTO CONDICIONAL Si $AC \langle 15 \rangle = 1 \therefore PC \leftarrow XXXX$ Si $AC \langle 15 \rangle \neq 1 \therefore PC \leftarrow (PC + 1)$
12	JMP	XXXX	El número XXXX se copia en el contador de programa. La siguiente instrucción es por lo tanto tomada siempre desde la ubicación XXXX.	SALTO INCONDICIONAL $PC \leftarrow XXXX$
13	INP	XXYY	Los 8 bits superiores del acumulador se borran a 0 y los caracteres de los 8 bits restantes que provienen del dispositivo de entrada YY se ingresan en la parte inferior del acumulador. La parte XX del dominio de la dirección es ignorada. La siguiente instrucción no se ejecuta hasta que la transferencia de datos se completa.	ENTRADA DE DATOS DESDE PERIFERICO $AC \langle 15..8 \rangle \leftarrow 0$ $AC \langle 7..0 \rangle \leftarrow (YY)$
14	OUT	XXYY	Los 8 bits más significativos del acumulador se envían al depósito de salida YY. La parte XX del dominio de la dirección es ignorado. Si el dispositivo de salida no puede aceptar los datos en ese momento, la máquina espera hasta que los datos sean aceptados antes de comenzar la ejecución de la siguiente instrucción.	SALIDA DE DATOS HACIA PERIFERICO $(YY) \leftarrow AC \langle 15..8 \rangle$
15	RAL	XXXX	Los contenidos del acumulador son rotados un lugar a la izquierda. El bit cambiado de AC_{15} se ingresa dentro de AC_0 , de modo que el cambio es cíclico. El dominio de la dirección es ignorado.	ROTACION A LA IZQUIERDA
16	CSA	XXXX	El juego de números dispuesto en el registro de llaves de la consola reemplaza al contenido del acumulador. El dominio de la dirección XXXX es ignorado.	COPIAR LLAVES EN EL ACUMULADOR $AC \leftarrow (SR)$
17	NOP	XXXX	Esta instrucción no tiene ningún efecto. El dominio de la dirección XXXX es ignorado.	NO OPERACION

Tabla 1: Conjunto de Instrucciones

Como el lector puede ver el juego de instrucciones disponible en esta máquina es limitado pero completo. Se pueden conectar más de 64 dispositivos de entrada y salida, cada uno es capaz de manejar caracteres de 8 bits. No posee facilidades tales como registros de índice, ubicación indirecta de direcciones, ni manejo de interrupciones. De todos modos, la máquina en cuestión será bastante complicada de analizar.

EL CICLO DE MÁQUINA BÁSICO

La computadora BLUE como muchas otras computadoras actuales, tienen un ciclo básico de dos partes. Ellas son los de BUSQUEDA (Fetch) y EJECUCION(Execute). Durante el ciclo de BUSQUEDA, la instrucción señalada por el contador de programa es extraída de la memoria y ubicada en el Registro de Instrucción (IR). Luego el número almacenado en el Contador de Programa (PC) se incrementa en uno, de modo que señala la celda de memoria siguiente a la que contiene la instrucción ya extraída de la memoria.

Al completar el ciclo de BUSQUEDA, la instrucción en el IR se analiza, se Decodifica y se Ejecuta.

EL FLUJO DE LA INFORMACIÓN

Sin ponernos a analizar en detalles las instrucciones, se puede tener una idea de cual información debe ser transferida dentro de la computadora. Consideremos primero la transferencia de direcciones. Los pasos a seguir se indican en la Figura .3. La Figura 4 muestra el flujo de los operandos e instrucciones, incluyendo los generados por la operación de los controles de la consola. Se destaca que la unidad aritmética tiene dos entradas: los registros Y y Z. La Figura 5 muestra las transferencias de entrada y salida de datos entre los distintos dispositivos. Contando las transferencias que se muestran en las Figuras 3,4 y 5, encontramos 5 pasajes posibles para transferencia de direcciones, 8 pasajes para operandos e instrucciones, y 3 pasajes (sin contar las líneas de control) para operaciones de entrada y salida, lo que hace un total de 16 pasajes diferentes.

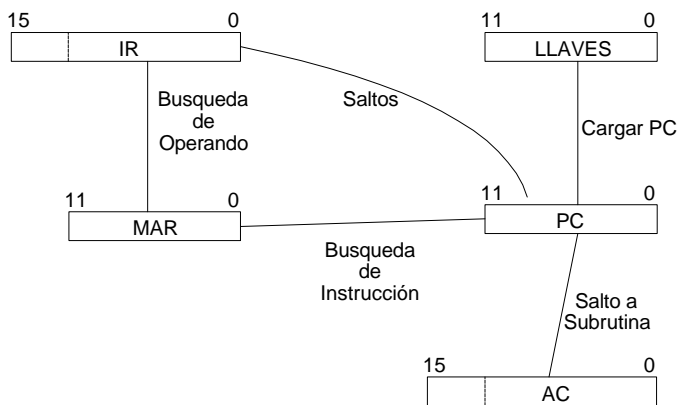


Fig.3: La transferencia de direcciones

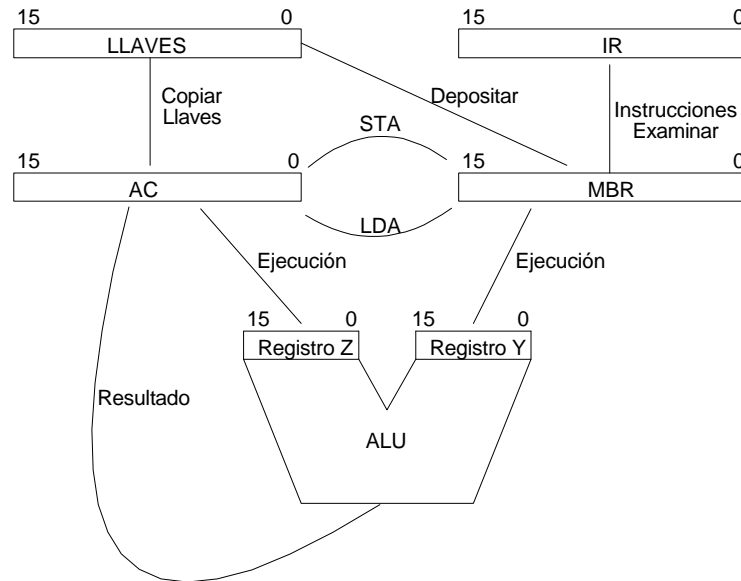


Fig.4: La transmisión de sentencias y operandos

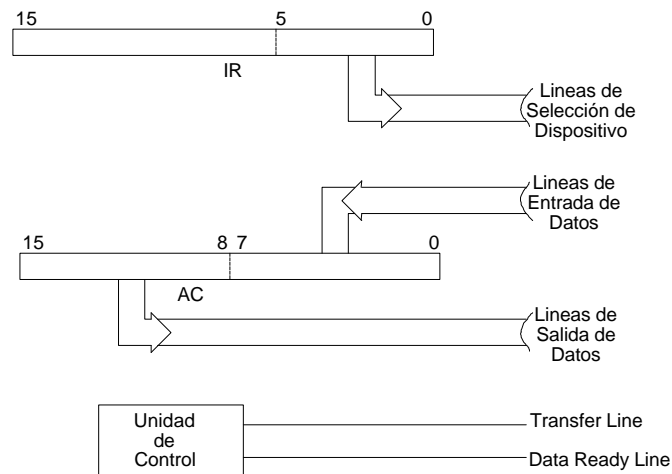


Fig.5: Las líneas de entrada-salida y selección y control

La Figura 6 muestra un esquema de la máquina rodeada de memoria y de entrada-salida incluida la consola. Se nota que existen conexiones diferentes entre CPU-Memoria de las conexiones CPU-Entrada-Salida

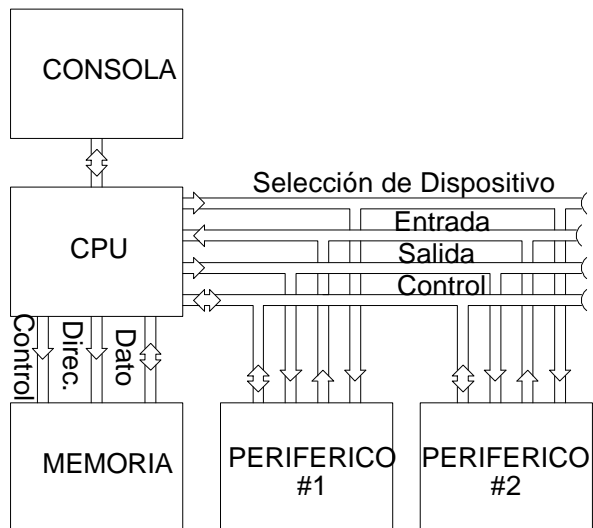


Fig 6: Esquema completo de la maquina

Estas transferencias se pueden efectuar de acuerdo a tres esquemas por lo menos. Ellos se denominan "Ruta de Información común" (Comon Bus), " Ruta de Información múltiple" (Multiple Bus), y " Ruta de Información Punto a punto" (Point to Point).

El esquema de "Bus Común" se ilustra en la Figura .7. Hay en el una "autopista" de uso común. Cualquier registro puede poner su información en esa autopista, y también copiar información de ella. Cuando se deben realizar muchas transferencias, es fácil ver que el número de ramales requeridos por el esquema de bus común será mucho menor que el que hubiera requerido el método punto a punto.

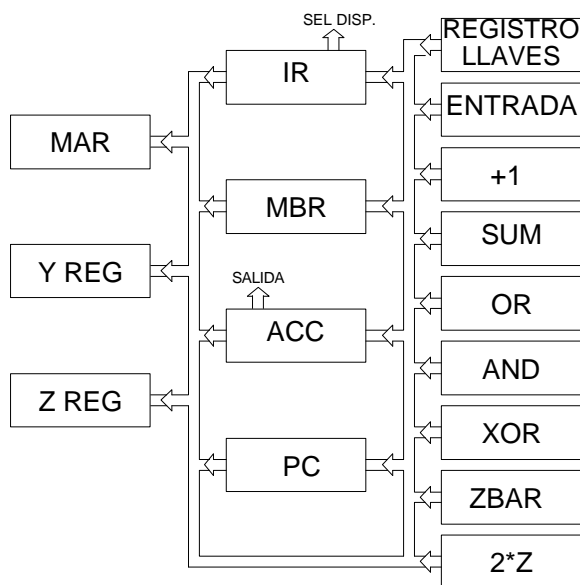


Fig. 7 El esquema de bus común usado en "BLUE". Notese que las líneas de selección de dispositivos desde el registro de instrucciones y las líneas de salida de datos desde el acumulador están, siempre energizados. Las seis salidas de la unidad aritmética se muestran por separado.

Pregunta 1 Supóngase que una máquina tiene n registros, cada uno de los cuales podría ser origen o destino para una transferencia de información. ¿Cual es el máximo número de ramales que se requerirían para implementar un esquema punto a punto? ¿y en un esquema de bus común?.

El pasaje desde el registro de instrucciones hacia las líneas de selección de dispositivos es externa a la estructura del vehículo, de modo que este puede ser usado para transmisión de datos mientras se selecciona un dispositivo.

IMPLEMENTACIÓN DE UN ESQUEMA DE BUS COMÚN

En un esquema de bus común se tiene una "Autopista", o grupo de cables que conecta a todos los registros. Se puede, por ejemplo, conectar el Acumulador al Registro Buffer de Memoria. En otro momento podría conectarse el RBM al Registro de Instrucciones, pero no se puede hacer ambas conexiones al mismo tiempo, debido que hay un solo camino del flujo de información.

En la Figura 8 se muestra tres registros de 2 bits y un "vehículo común" que los conecta entre si. Este BUS es diseñado usando flip-flop tipo D disparado por flanco ascendente. Toda información que aparezca en la autopista común podrá copiar por ejemplo, en el registro A, con solo activar brevemente la línea denominada Cargar A (Load A). Para poner información en el BUS, se activa alguna de las líneas de envío (Send).

Pregunta 2: ¿Qué ocurre si activamos 2 líneas de envío al mismo tiempo?

Para copiar información del registro C al registro B, se activa la línea de envío de B, y mientras esto ocurre, elevamos también la línea de carga de B. La Figura 9 muestra un diagrama de tiempos.

Pregunta 3: ¿Importa en que orden dichas líneas vuelven a desactivarse?

Pregunta 4: ¿Qué dato se introduce en B, si activamos la línea de carga de B cuando ninguna línea de envío está activada?

Pregunta 5: El esquema de la Figura 8 fue diseñado usando flip-flops de tipo D. Diseñe un circuito equivalente al de la Figura 7 usando flip-flops tipo J-K.

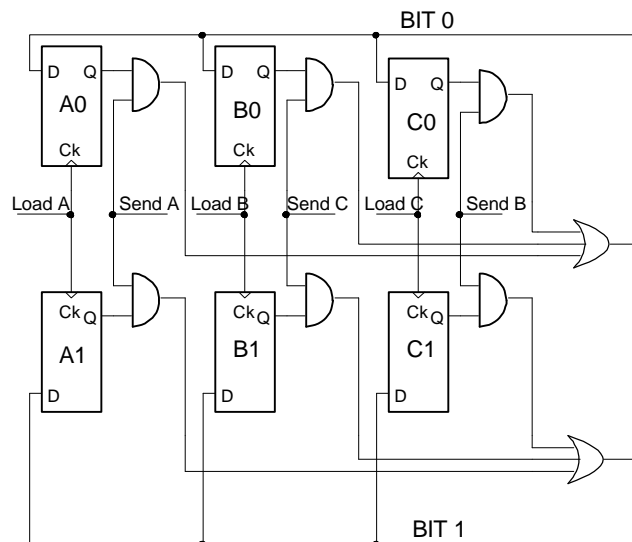


Fig. 8: 3 registros de 2 bits conectados por un "bus común"

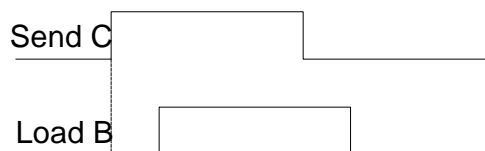


Fig. 9: Diagrama de los pulsos requeridos para copiar información desde el Registro C al Registro B.

ELEMENTOS DEL CIRCUITO DE LA "BLUE "

La 'BLUE' se construye con circuitos integrados convencionales, excepto los registros, que serán hechos con FLIP-FLOP tipo D. Cada registro tiene una línea común de 'reloj' (también llamada 'Cargar Registro'), la cual tiene un pequeño retardo (delay) de 10-15 nseg incorporado. El objeto del retardo es permitirnos presentar un dato a las entradas de registro, antes de ordenar que se cargue dicho registro. de manera de estabilizar los circuitos de entrada y así asegurar una copia exacta de la información.

Cuando el contador de programa (PC) se transfiere al registro Acumulador y puesto que los 4 bits superiores del PC (PC 15-12) no existen, se cargaran ceros en A(15-12) debido que no se pone señal alguna en las líneas B15 a B12.

UNIDAD DE CONTROL

Ahora se debe considerar la generación de las distintas secuencias de pulsos y señales requeridas para el funcionamiento de la computadora. Esta es una tarea de la Unidad de Control, cuyo trabajo es coordinar las acciones de la maquina.

De acuerdo al modo de generación de la secuencia de pulsos, las computadoras se clasifican en **asíncronas y síncrona,**

En el diseño asíncrono, que no estudiaremos aquí, cada unidad es cargada con los resultados de su antecesor cuando esta lista para recibirlos. Esto metodología tiene la ventaja que como mucho el tiempo de ejecución de una tarea es el tiempo que ella misma necesita para completarse La desventaja es que cada unidad debe tener la suficiente inteligencia como para conocer cuando la tarea esta completa.

El diseño síncrono o 'cronometrado', fracciona el tiempo en periodos fijos, llamados 'ciclos menores'. Un suceso solo ocurre cuando el reloj emite un pulso. Este método tiene la ventaja de mantener diversas unidades en secuencia y otra ventaja adicional es que se requiere menos lógica de los dispositivos. La obvia desventaja es que ningún suceso puede tener lugar en menos tiempo que un ciclo menor completo.

Se hizo de la BLUE una maquina síncrona porque es mas fácil de comprender su diseño e implementación.

Se va ahora a comenzar al diseño de la unidad de control. La Unidad de Control tiene 3 flip-flop, denominados flip-flop de Corrida (RUN),de Estado (STATE) y de Transferencia (TRA). El primero se activa (Q=1) cuando se presiona la tecla de Comienzo (START) y se desactiva con la pulsación de la tecla de Parada (STOP), con el comando Parar (HLT) y cuando se detecta un rebasamiento en la capacidad aritmética de la maquina.

El F-F STATE tiene dos condiciones, que corresponden a los ciclos de Búsqueda y Ejecución de una Instrucción. El F-F STATE se coloca en la posición de búsqueda F (FETCH) por medio del tecla START o al concluir un ciclo de ejecución. Según cual instrucción esté en el registro de instrucciones, podrá o no ubicarse en la posición de ejecución E (EXECUTE) al final del ciclo de búsqueda.

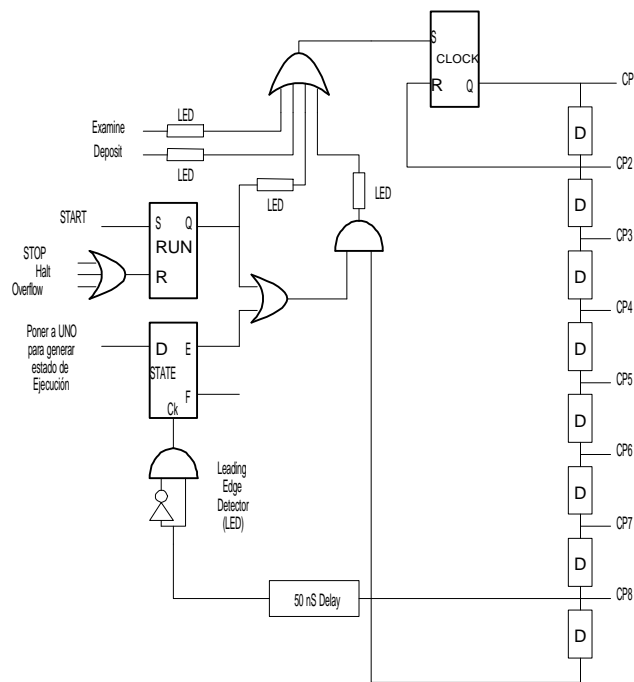


Fig. 10: Circuito de control y de reloj.

Cada ciclo de maquina se divide en 8 intervalos de 125 nseg. denominados ciclos menores. El circuito de reloj maestro tiene 8 líneas de salida, cada una de las cuales contiene un pulso del ciclo de maquina. La Fig 10 muestra una de las varias maneras de diseñar un reloj maestro, y la Fig 11 muestra la relación de tiempo entre los pulsos que van por las líneas CP mientras un pulso de 125 nseg. viaja por la cadena retardos.

Cuando el F-F RUN pasa a la posición ONE, un circuito detector de flancos genera un pulso positivo que enciende al flip-flop de la Fig 10. Si el F-F RUN esta encendido, apretar el botón START no causara ningún efecto. Esto energiza la salida de un flip-flop y el cambio de estado se propaga hacia el retardo D. 125 nseg., mas tarde el estado ONE emerge del primer retardo y es realimentado para apagar el flip-flop. El flanco de subida de este pulso, de 125 nseg. de ancho, se propaga a través de la cadena de retardos, seguido por el consecuente flanco de bajada 125 nseg. mas tarde. Esto genera pulsos de ancho uniforme en las salidas CP1 al CP8, cada uno demorado 125 nseg. respecto del anterior.

Finalmente, el pulso sale al final de la cadena de retardo y enciende el flip-flop de nuevo (siempre y cuando RUN este encendido o STATE esté en el estado de ejecución).

Pregunta 6: Hay muchas otras maneras en que podríamos haber generado estos pulso de reloj ¿Cuántas puede Ud. imaginar?.

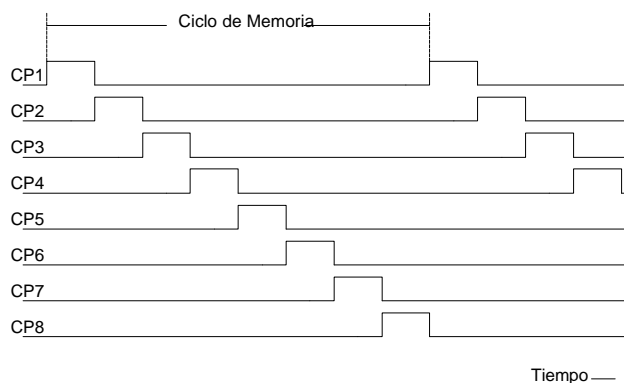


Fig. 11 Diagrama de tiempos de las señales de las líneas CP.

EL CICLO DE BUSQUEDA

Cuando el F-F RUN está encendido, se habilita al reloj (también lo pone en marcha) y si además el F-F STATE esta en la posición FETCH, la acción que desarrolla la maquina es la de cargar el Registro de Instrucciones con la información de las celda apuntada por el Contador de Programa (Program Counter PC), y luego agrega un 1 al contenido PC. La acción antes descripta se la conoce como Ciclo de Búsqueda.

La Tabla 2 muestra la secuencia de dichos sucesos. Con el pulso de reloj 1 (CP-1), se copia el contenido del Contador de Programa en el Registro de Dirección de Memoria y en el Registro Z, y se inicia el ciclo de lectura de memoria. En el CP-2 se pone un +1 en Registro Y. Durante CP-3 se concede tiempo para que la suma se desarrolle, y en CP4 se transfiere la suma al Contador de Programa, usando así la unidad aritmética, que de otro modo quedaría ociosa. Hacia el final de CP-4, la próxima instrucción (el dato traído desde la memoria) estará lista para ingresar en el MBR, lo que ocurrirá en el CP-5. En CP-6 se copia el MBR en el Registro de Instrucciones y se comienza la decodificación de la instrucción.

Reloj	Acción	Comentarios
1	Send PC, Load MAR, Load Z Memory Read	Copia el PC en MAR y Z. Inicia ciclo de lectura de memoria.
2	Send +1, Load Y	Hace Y igual a +1
3		
4	Send SUM, Load PC	Incrementa el PC en uno, de manera que apunte a la siguiente instrucción.
5	Strobe MBR	Carga el MBR con el dato proveniente de memoria.
6	Send MBR, Load IR	Copia la nueva instrucción en el registro de intrucciones y comienza su decodificación.
7		Disponibles para la decodificación de la instrucción y tal vez para la ejecución.
8		

Tabla 2: El ciclo de búsqueda.

Al comienzo de CP7 ya la máquina está lista para ejecutar la nueva instrucción. Pero puesto que el ciclo de maquina tiene 8 pulsos² se dispone de los pulsos CP-7 y CP-8 para realizar alguna ejecución.

Para algunas de las instrucciones de BLUE, este tiempo es suficiente para completar todo los pasos necesarios. Ello se muestra en la Tabla 3, con una descripción de las acciones que producen.

² NT: La versión original de la BLUE tenía una memoria de acceso aleatorio con tecnología de ferrite. Estas memorias tienen la particularidad que se debe volver a escribir la información leída, pues de otra manera el dato de la celda se pierde. Por este motivo es que se dispuso de un 8 pulsos de reloj, necesarios para la reescritura de la memoria. En la presente traducción no se hace referencia a las memorias de ferrite, sino que se utilizan las mas actuales memorias de semiconductores.

Reloj	HALT	NOP	JMP	JMA	SRJ	CSA	NOT	RAL
7	-	-	-	-	Send PC Load A	-	Send A Load Z	Send A Load Z
8	Off→ RUN	-	Send IR Load PC	Si A15=1 Send IR Load PC	Send IR Load PC	Send SR Load A	Send Z Bar Load A	Send 2*Z Load A

Tabla.3 Las ocho instrucciones de un ciclo en el final del ciclo de búsqueda.

Puesto que ninguna de estas instrucciones requiere de un ciclo de ejecución, no cambian el F-F de STATE, dejándolo en FETCH. Así el siguiente ciclo será otra búsqueda de una nueva instrucción.

INSTRUCCIONES DE ENTRADA Y SALIDA

Hay dos instrucciones que no requieren de referencias de la memoria, pero cumplimentarlas toma más de un ciclo. Ellas se muestran en la Tabla. 4. Para ver como funcionan las instrucciones de entrada y salida, se debe observar como son los periféricos conectados a la computadora. Por ejemplo asumamos que el dispositivo periférico Nro. 1 es una lectura de cinta de papel, y el Nro. 2 es una perforadora de cinta de papel. La lectura de cinta estará LISTA (READY) si la cinta esta montada y si un caracter (fila transversal de perforaciones) está ubicado sobre la cabeza lectora. Esta condición puede ser reconocida por una célula fotoeléctrica, que 'busca' la perforación guía de la cinta. La lectora de cinta estará NO LISTA (NOT READY) bajo cualquier otra condición. Cuando la impresora está NO LISTA, el embrague hará mover la cinta para leer el siguiente caracter. Cuando éste se ubica bajo la cabeza lectora, el borde del agujero guía asociado a este caracter, genera un pulso que detendrá la cinta y pondrá la lectora en condición de LISTA. Ver Fig.12. Desde la computadora hacia cada dispositivo de entrada y salida, hay 6 líneas llamadas Líneas de Selección de Dispositivo (Device Selection Lines). Ellas están conectadas a los 6 bits más bajos del Registro de Instrucciones y continuamente se transmiten a todos los dispositivos de entrada y salida. En el CP-7 del Ciclo de Búsqueda de una instrucción de entrada, se enciende el F-F de Transferencia (TRA). Esto indica a cada dispositivo que será examinado para comprobar si es él el requerido, por medio de la compuerta AND A de cada uno. Un dispositivo se puede seleccionar solamente durante una sentencia de entrada o salida porque solamente en esta ocasión el F-F TRA estará activado. Cuando el F-F READY se enciende, la puerta B se activa y las 8 celdas fotoeléctricas pondrán el dato leído en las Líneas de Entrada de Datos (Data Input Lines) al mismo tiempo que la línea R es llevada a 1. Si el F-F READY no está activado, nada ocurre hasta que se detecta un agujero.

Reloj	Acción		Comentarios
	Entrada INP	Salida OUT	
7	1→ TRA	1→ TRA	Transmite IR 5-0 a través de las líneas de selección de dispositivo
8	E→ State	E→ State	A continuación se procede con un estado de ejecución.
Ciclo de Búsqueda			
Ciclo de Ejecución			
1	-	-	
2	-	-	
3	-	-	
4	-	-	
5	-	-	
6	-	-	
7	Si R=1 Send Data Inp Lines Load A 0 → TRA	Si R=1 0 → TRA	Si el dispositivo está listo (R=1), se copia el dato y se limpia TRA
8	Si TRA=0 F → State	Si TRA=0 F → State	Si se ha completado el ciclo de Entrada/Salida, se procede con la búsqueda de una nueva instrucción. De otra manera, se repite el ciclo de ejecución.

Tabla.4 Las instrucciones de entrada/salida.

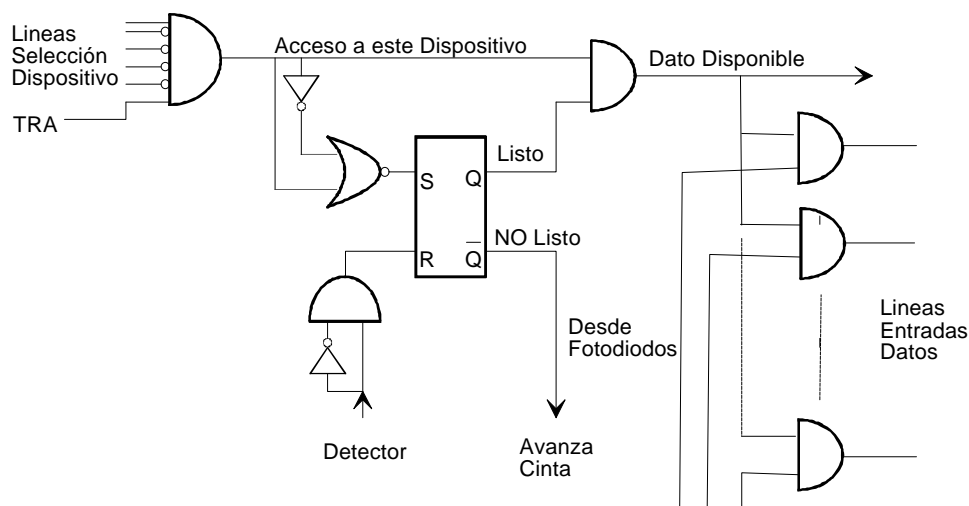


Fig 12: Los circuitos de la lectora de cinta de papel en BLUE.

Cuando el F-F se resetea, el embrague se energiza y la cinta avanza hasta el siguiente caracter. Volviendo atrás en el CP-8 del Ciclo de Búsqueda, el F-F STATE fue puesto en posición EXECUTE. En el CP-7 del Ciclo de Ejecución de la instrucción de entrada, se prueba la línea R. Si está en 0 (cero), la computadora espera otro ciclo mayor y prueba de nuevo. Cuando el dispositivo está LISTO, la línea R se hace 1 y, en el CP-7, las líneas de entrada de datos se ingresan en los 8 bits más bajos del acumulador. Ahora se tiene el caracter 'CAPTURADO', y la máquina está lista para: 1) avanzar la lectora (de modo de obtener el siguiente caracter para la siguiente instrucción de entrada, cuando quiera que esta venga) y 2) Terminar esta instrucción.

Por eso, en el CP 7, si $R=1$, volvemos el flip-flop TRA a 0 (cero). Esto remueve el número de dispositivos de las Líneas de Selección de Dispositivo, haciendo que la línea 'lea este dispositivo' de la lectora de cintas vuelva a 0 (cero). Un detector de flancos de bajada, denominado C en la Fig 12, genera un pulso positivo que pone al F-F READY en la posición NO LISTO. Esto energiza el embrague y la cinta avanza.

Finalmente, en el CP-8 del Ciclo de Ejecución de la instrucción de entrada, si el F-F TRA= 0, se pone al sistema en el estado de búsqueda.

La operatoria de la instrucción de salida es muy similar a la de entrada. Los circuitos asociados a una perforadora de cinta de papel se muestran en la Fig. 12. La perforadora sigue una operación como la describe la Tabla 4 y la Fig 12. Nótese que puesto que la computadora origina la sentencia OUT (y la termina) tan pronto como la perforadora queda ociosa, la perforadora debe tener un buffer de almacenamiento donde mantener la información que mueve los imanes. Esto se logra por medio de 8 flip-flop cuya información se ingresa desde las 8 líneas de salida de datos.

PREGUNTA 7: Se podría haber eliminado el buffer de la perforadora haciendo que la máquina 'se colgara' en la instrucción OUT hasta que el ciclo de perforación se completara. Arme los circuitos y los detalles de la instrucción que se hubieran requerido para este caso.¿ Puede la perforadora estar 'ocupada' cuando comienza la instrucción OUT?.

INSTRUCCIONES DE DOS CICLOS

Las siguientes intrucciones que estudiaremos serán: LDA y STA, ambas son instrucciones no-aritméticas con referencia a memoria. Se considera primero la instrucción Cargar (Load) (ver Tabla.5). En el CP-8 de la Fase de Búsqueda, se pone F-F STATE a la posición Ejecutar. En el CP-1 de la fase de Ejecución se carga el Registro de Direcciones de Memoria (MAR) desde el Registro de Instrucciones (IR) y se inicia el ciclo de lectura de memoria. Hacia el comienzo del CP-5 se lee el dato proveniente de la memoria y se lo introduce en el Registro Buffer de Memoria (MBR). En el CP-6 el contenido del MBR se copia en el Acumulador (ACC). En el CP-8 se pone F-F STATE en Búsqueda.

La instrucción de almacenar (Store) el acumulador es casi igual, pero no se pone el contenido de la celda de memoria en el MBR, sino que en su lugar, en el CP- 5 cargamos el MBR desde el acumulador.

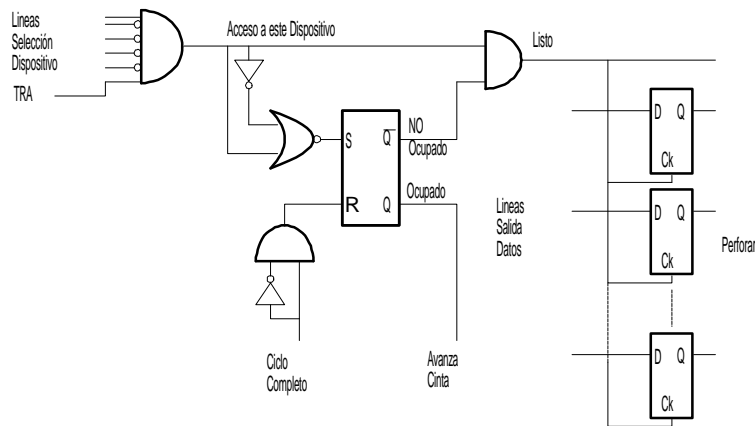


Fig.13 Circuito de la perforadora de papel

Reloj	Acción		Comentarios
	LDA	STA	
7			
8	E → State	E → State	A continuación se procede con un estado de ejecución.
Ciclo de Búsqueda			
Ciclo de Ejecución			
1	Send IR Load MAR Memory Read	Send IR Load MAR	Copia el IR ₁₂₋₀ en MAR. Inicia ciclo de lectura de memoria.
2	-	-	
3	-	-	
4	Strobe MBR	Send A Load MBR Memory Write	Carga MBR con el dato proveniente de memoria/acumulador. Inicia ciclo de escritura.
5	Send MBR Load A	-	Copia MBR (Dato) en acumulador.
6	-	-	
7	-	-	
8	F → State	F → State	Se procede con la búsqueda de una nueva instrucción.

Tabla 5 Instrucciones de carga y almacenamiento del acumulador.

LA UNIDAD ARITMETICA

Para describir las cuatro instrucciones restantes, debemos primero describir la Unidad Aritmética y ver como trabaja. Observemos primero el proceso de adición de dos números binarios. Considere una posición de un bits entre medio de una palabra. Existen dos entradas (a,b) y posiblemente un acarreo (Carry) desde el bit anterior menos significativo (inmediatamente a la derecha). En esta posición, se debe generar un bit SUMA (SUM) y quizás un bit de acarreo local (Carry), el cual irá a la posición de la izquierda. La Fig. 14 muestra un Sumador (ADDER) en forma simbólica. Nótese que hay 3 tipos de compartimentos: α , β y γ .

α es un circuito que corresponde a la posición (bit) del signo, y deberá generar el signo resultante de la suma y una señal de rebosamiento si dos números positivos dan un resultado negativo y viceversa. La Tabla 6 muestra las 'tablas de verdad' de α , β y γ .

β , por razones que se verán después, es llamado usualmente Semi Sumador (Half Adder). Se puede construir según la Fig. 15. Se nota que la salida S es la operación OR Exclusiva (XOR) de las entradas.

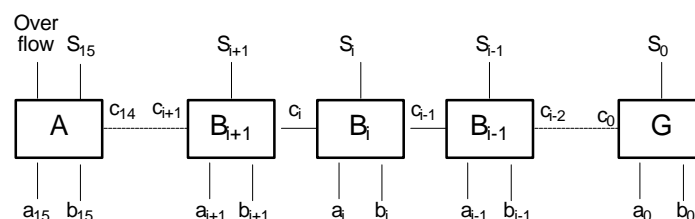


Fig 14 Un esquema de un sumador binario paralelo $A+B=S$ [JY15]

α					β					γ			
a_{15}	b_{15}	c_{14}	S_{15}	OV	a_i	b_i	c_{i-1}	S_i	C_i	a_0	b_0	S_0	c_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	1	1	0	0	1	1	0
0	1	0	1	0	0	1	0	1	0	1	0	1	0
0	1	1	0	0	0	1	1	0	1	1	1	0	0
1	0	0	1	0	1	0	0	1	0	1	1	0	0
1	0	1	0	0	1	0	1	0	1	1	0	1	0
1	1	0	0	1	1	1	0	0	1	1	1	0	0
1	1	1	1	0	1	1	1	1	1	1	1	1	1

Tabla. 6 Tabla de verdad de α, β y γ .

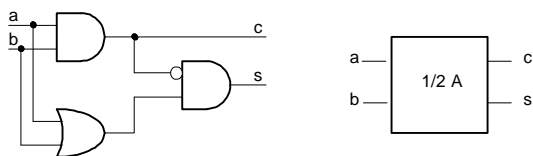


FIG.15 Semi Sumador y su representación simbólica.

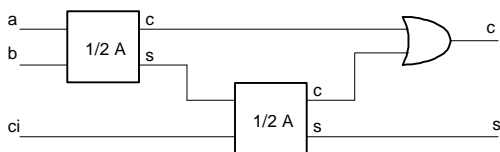


Fig.16: Un Sumador Total construido con 2 Semi Sumadores

Un análisis un poco más profundo nos muestra que se puede implementar la tabla de verdad para β combinando 2 Semi Sumadores y un compuerta 'OR' como se muestra el la Fig. 16

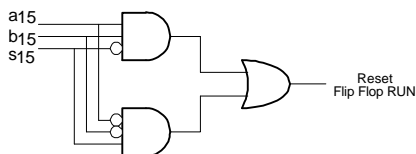


Fig.17: Circuito que genera una señal de rebosamiento.

El circuito para generar S_{15} puede ser exactamente igual que el Sumador Total sin la compuerta OR para generar el acarreo. Para generar el sobrepasamiento (Overflow) se puede usar un circuito como el mostrado en la Fig 17

PREGUNTA 8: Si se implementa α como se sugiere arriba, se necesitan un total de 9 compuertas. Se puede hacer lo mismo con menos compuertas?

Ahora estamos listos para examinar un bit típico de la Unidad. Aritmética, recordando los sumadores de los bits cero y el quinceavo son diferentes del resto. El circuito que se muestra en la Fig.18

Los registros Y y Z juntos forman las entradas a la unidad aritmética. Se pueden cargar desde el 'vínculo común'. Del contenido de Z se obtienen la operación de Desplazamiento a la Izquierda ($2*Z$) y Complemento ($Zbar$) su complemento conectando al bus las salidas respectivas. De la misma manera se obtienen la suma de Y y Z, la Or inclusive y exclusiva y la operación And. Nótese que en particular la función OR exclusiva viene 'gratis'. La salida de la suma del primer sumador por mitades genera la XOR automáticamente, de modo que todo lo que se requiere es una salida al bus común. Esta situación gratuita ocurre

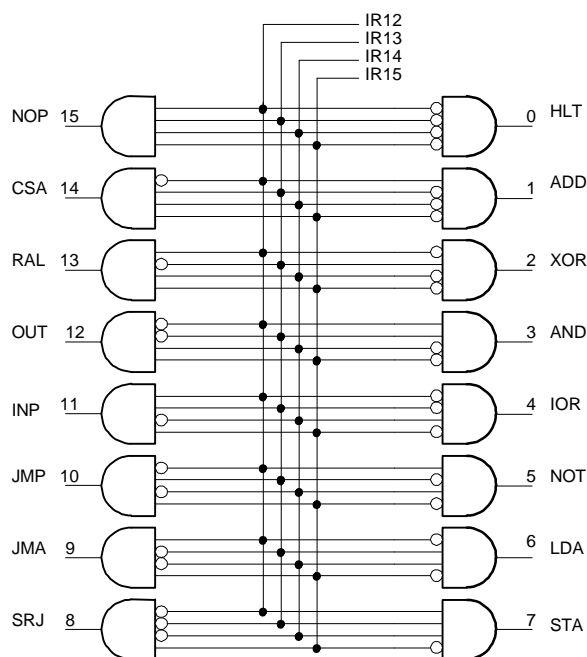


Fig 19: Arbol de decodificación completa.

CIRCUITOS DE LA UNIDAD DE CONTROL.

Aún no se ha indicado como la unidad de control decide que sentencia se debe ejecutar. Los 4 bits más significativos del Registro de Instrucción se cargan en lo que se denomina un **Árbol Completo de Decodificación** (Complete Decoding Tree) en recuerdo en que de una época en que esta operación la realizaba un árbol de relays electromecánicos. La Fig.19 muestra dicho árbol con sus salidas apropiadamente indicadas. Estos 'demultiplexadores 1 a 16 están ahora disponibles en un solo chip. Las figuras 20 al 29 y la Tabla 8 muestran los diagramas de circuitos lógicos de la Unidad de Control. Cuando a un lugar dado entran varias señales, por ejemplo 'cargar A' es activado desde varias lugares diferentes, se entiende que debe haber un elemento OR que separa dichas señales. Los detalles de la memoria ya se han discutido así como la estructura de 'bus común'. De tal modo que la descripción de la BLUE está completa.

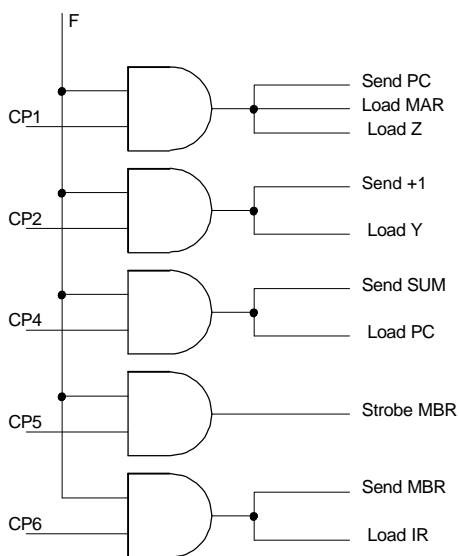


Fig. 20: Compuertas de la primera parte del ciclo de búsqueda

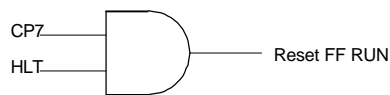


Fig. 21: Implementación de la instrucción Parar (Halt)

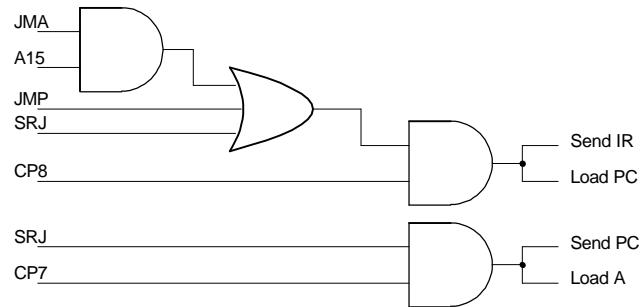


Fig. 22: La tres instrucciones de salto.

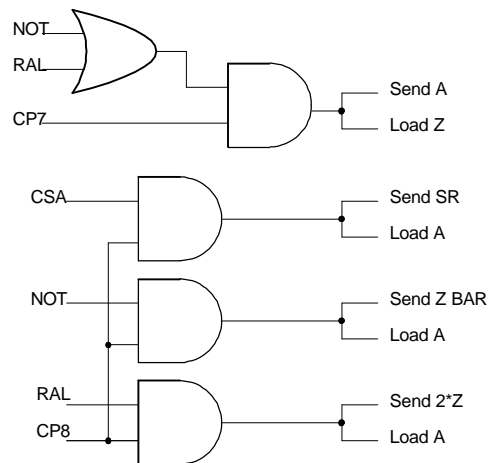


Fig. 23: Implementación de las instrucciones Copiar Llaves al Acumulador (CSA), Negación (NOT) y Rotación del Acumulador a la Izquierda (RAL)

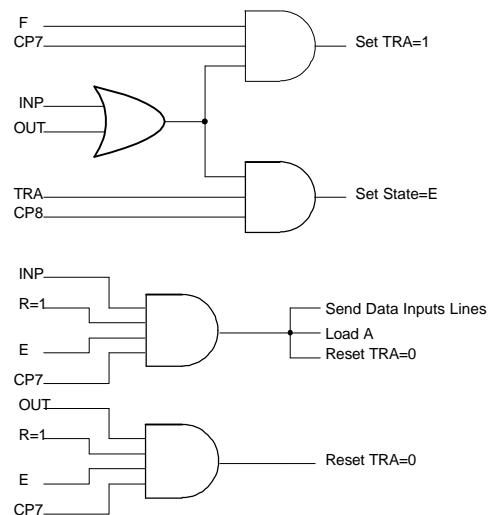


Fig. 24: Las instrucciones de Entrada y Salida.

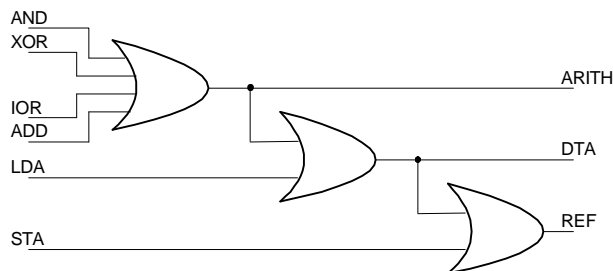


Fig. 25: Generación de las señales ARITH, DTA y REF.

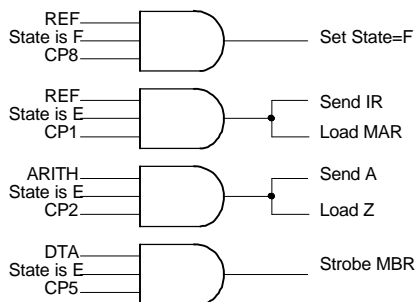


Fig. 26: Primeros pasos de las instrucciones de referencia a memoria



Fig. 27: El resto de la instrucción almacenar el acumulador (STORE)

PREGUNTA 9 : Ya hemos intentado cambiar señales donde quiera que podemos. Vea por ejemplo la señal REF que se usa en varias instrucciones ¿Es posible asignar instrucciones a un código numérico operacional en una manera distinta a la de la BLUE, de modo que se obtengan ahorros aún mayores ?



Fig. 28: El resto de la instrucción cargar el acumulador (LOAD)

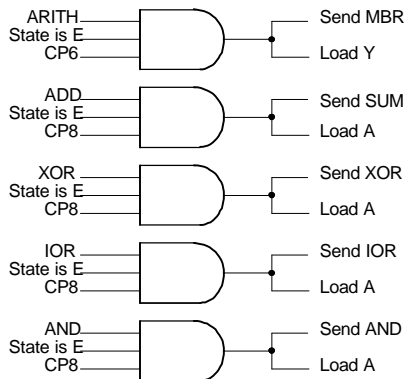


Fig. 29: El resto de las cuatro instrucciones aritméticas

Reloj	Acción		Comentarios
	Examine	Deposit	
1	Send PC Load MAR Load Z Memory Read	Send PC Load MAR Load Z Memory Read	Copia el PC en MAR y Z. Inicia ciclo de lectura de memoria.
2	Send +1 Load Y	Send +1 Load Y	Copia +1 en Y de la ALU.
3	-	-	
4	Send SUM Load PC	Send SUM Load PC	Incrementa PC en 1
5	Send MBR Load A	Send LLaves Load MBR Memory Write	
6	Send MBR Load IR	-	
7	-	-	
8	-	-	-

Tabla 8 Detalle de las operaciones generadas por la llaves Examinar y Depositar del panel frontal. Se asume aquí y en la Fig. 10 que dichas llaves proveen una señal limpia (sin rebotes) de al menos 1 μ s

PREGUNTA 10 : Arme los circuitos necesarios para implementar la tabla 8.

CONCLUSIONES

En el presente trabajo se ha presentado el diseño de una computadora muy sencilla, tanto que probablemente nadie actualmente quiera comprarla, aunque varias computadoras existentes son de la misma clase que la BLUE.

En la BLUE se tienen instrucciones de cargar, almacenar, sumar complementar y bifurcación. Esto es lo que realmente se necesita para cualquier cálculo. Adicionalmente la máquina posee un razonable conjunto de funciones lógicas e instrucciones de entrada salida capaces de direccionar hasta 64 dispositivos periféricos.

No existen ninguno de las “buenas características” presentes en la mayoría de las computadoras actuales, tales como: direccionamiento indirecto, registros índices, interrupciones etc.