

USO DEL SIMULADOR LOGISIM COMO HERRAMIENTA DE APOYO PARA EL ESTUDIO DE UNA UNIDAD DE CONTROL DE UN PROCESADOR BÁSICO

USE OF THE LOGISIM SIMULATOR AS A SUPPORT TOOL FOR THE STUDY OF A CONTROL UNIT OF A BASIC PROCESSOR

Carlos Federico Hernández Farfán

Tecnológico Nacional de México / ITS de Irapuato, México

carlos.hf@irapuato.tecnm.mx

Recepción: 24/noviembre/2023

Aceptación: 26/diciembre/2023

Resumen

Cuando se realiza el estudio básico de un procesador, surgen algunos problemas tales como, la complejidad del procesador, recursos limitados de infraestructura de laboratorio, la búsqueda de un balance teórico y práctico de los conceptos, y que las señales internas del procesador ocurren en intervalos de tiempo muy cortos. En este trabajo se presenta el uso del simulador Logisim como apoyo para el estudio de un procesador simplificado de 16 bits y su unidad de control. Se muestra el funcionamiento de la unidad de control, al ejecutar el ciclo de instrucción para diferentes instrucciones del procesador. Se muestra qué ocurre en el procesador, en qué momento y en qué condiciones, cuando se ejecuta el ciclo de instrucción. Se implementa el procesador sin tener que requerir una mayor infraestructura de laboratorio, mejorando el balance teórico - práctico de los conceptos y visualizando las señales internas del procesador.

Palabras Clave: Arquitectura de computadoras, Logisim, Unidad de control.

Abstract

When the basic study of a processor is carried out, some problems arise such as, the complexity of the processor, limited resources of laboratory infrastructure, the search for a theoretical and practical balance of the concepts, and that the internal signals of the processor occur in intervals very short time. This paper presents the use of the Logisim simulator as support for the study of a simplified 16-bit processor

and its control unit. The operation of the control unit is shown, when executing the instruction cycle of processor instructions. It shows what happens in the processor, at what time and under what conditions, when the instruction cycle is executed. The processor is implemented without having to require a larger laboratory infrastructure, improving the theoretical-practical balance of the concepts, and visualizing the internal signals of the processor.

Keywords: *Computer architecture, control unit, Logisim.*

1. Introducción

Dentro de la carrera de Ingeniería en Sistemas Computacionales del Tecnológico Nacional de México campus Instituto Tecnológico Superior de Irapuato, se imparte la materia de Arquitectura de Computadoras. En la materia se estudian conceptos relacionados con el funcionamiento de un procesador y su unidad de control, como se muestra en [Tecnológico Nacional de México campus Instituto Tecnológico Superior de Irapuato, 2023]. Cuando se imparte la materia es común que se presenten algunas situaciones problemáticas tales como, la complejidad del procesador, recursos limitados de infraestructura de laboratorio, la búsqueda de un balance de contenido teórico y práctico del curso; y en particular de que las señales que se generan en la unidad de control y los eventos que ocurren internamente en el procesador ocurren en intervalos de tiempo sumamente pequeños lo cual dificulta su estudio y visualización.

En atención a la situación problemática mencionada, en lo que se refiere a la revisión bibliográfica, en el libro de texto de Arquitectura de Computadoras de [Mano, 1994], se describe la arquitectura de un procesador básico de 16 bits, que se puede considerar muy didáctica, y que permite mostrar los componentes internos y funcionamiento del procesador y de la unidad de control, durante la ejecución de un programa ensamblado en memoria. Aunque no presenta la forma de realizar la implementación física o simulación del procesador o de la unidad de control, ni la forma de simular la ejecución de los programas en ensamblador.

En otros libros de texto de Organización y Arquitectura de Computadoras como el de [Tanenbaum, 2000] o el de [Parhami, 2007], también se describen los conceptos

relacionados con el procesador y la unidad de control, aunque enfocados a procesadores más complejos y sin incluir la forma de su implementación física o simulación. En el libro de [Salido, 2023], se presentan algunos conceptos de lógica digital y tecnología de computadores con un enfoque práctico empleando el simulador Logisim, pero no incluye los conceptos relacionados con la implementación de un procesador y su unidad de control.

Existen diferentes artículos en los que se presentan la implementación de procesadores, incluso más complejos, para fines didácticos; que emplean, por ejemplo, el uso del lenguaje VHDL, la técnica de diseño asistido por computadora y la implementación en circuitos FPGA como se muestra en [Gurrola, 2022]. Sin embargo, tienen un enfoque hacia el uso o programación del procesador y no a mostrar lo que ocurre internamente en el procesador o cómo se generan las señales de control y temporización. Además de que se requiere el aprendizaje del lenguaje VHDL, la interpretación mediante gráficas de las señales de control y la adquisición de las tarjetas de circuitos FPGA.

También en trabajos previos como en el de [Hernández, 2017], se muestra la técnica de diseño asistido por computadora y el uso del lenguaje VHDL como apoyo didáctico para el estudio de una unidad de control del tipo microprogramada, pero no se muestra la implementación de la unidad de control con su procesador.

Continuando con la revisión de artículos, existen diferentes trabajos en los que se destaca el uso de simuladores como herramienta de apoyo didáctico para mejorar la comprensión del funcionamiento interno de la computadora, como se muestra en [Prasad, 2015]. En otros artículos, se muestra el uso del simulador Logisim de [Burch, 2011] en la materia de Arquitectura de computadoras, en los que se mencionan las características y versatilidad de su uso con fines didácticos, tales como en [Hanafi, 2017], [Hernández, 2019], [Hanafi, 2020], y [Sadat, 2022]; en los que se muestra la implementación de procesadores de 4 y de 8 bits, así como el grado de aceptación del simulador como herramienta didáctica por parte los estudiantes. También existen simuladores de procesadores tales como Simuproc de [Yepes, 2003], o simuladores en línea como CPUlator de [CPUlator computer system simulator, 2023], que permiten la ejecución de un programa en ensamblador,

mostrando el contenido de los registros del procesador durante el ciclo búsqueda de instrucción, decodificación y ejecución de la instrucción, pero no muestran los detalles internos de la arquitectura del procesador, ni las señales generadas por la unidad de control.

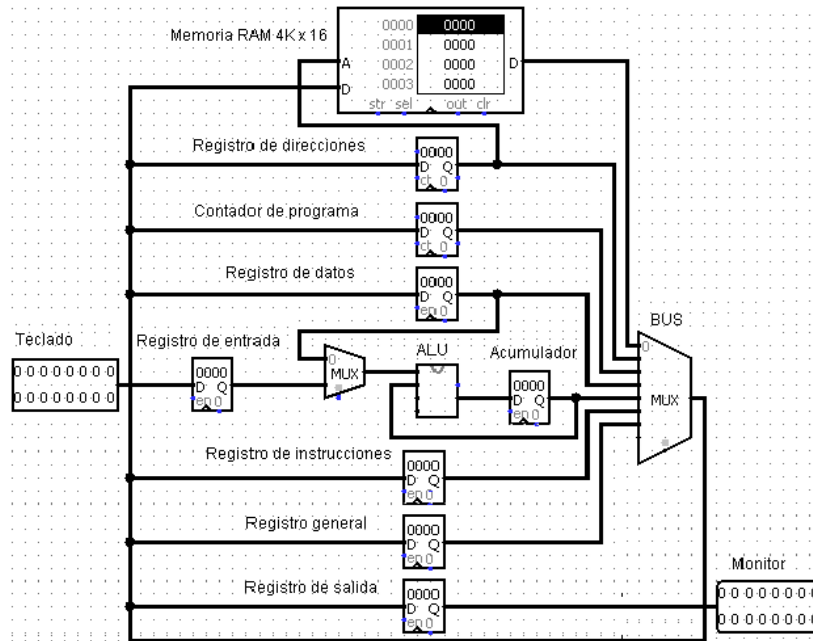
De acuerdo a la situación problemática y la respectiva revisión bibliográfica, este trabajo tiene como intención implementar un procesador básico y su unidad de control mediante el uso del simulador Logisim, como herramienta de apoyo en el estudio de una unidad de control del tipo cableada, que permita visualizar los componentes internos del procesador y la unidad de control; así como visualizar las señales de control, el momento en que se generan y el efecto que producen en los componentes del procesador cuando se ejecuta el ciclo de instrucción o un programa ensamblado en la memoria RAM.

2. Métodos

Para el desarrollo de este trabajo, se empleó el simulador Logisim versión 2.7.1 de [Burch, 2011], debido a que se trata de software libre, bajo licencia pública GNU, que consume muy pocos recursos de cómputo, que cuenta con una interfaz amigable y de fácil manejo; así como por su versatilidad y características para ser empleado como herramienta didáctica en la enseñanza de la materia de Arquitectura de Computadoras. Mediante el uso del simulador, es posible realizar un circuito integrado a partir de componentes digitales más sencillos, permitiendo visualizar el comportamiento del circuito integrado, así como los componentes internos del mismo.

Este trabajo está basado en la arquitectura de un procesador de 16 bits, descrito por [Mano, 1994]. Se implementó el procesador a partir de sus componentes básicos: una memoria RAM de 4K x 16, un registro de direcciones (AR), un registro contador de programa (PC), un registro de datos (DR), una unidad lógico-aritmética (ALU), un registro acumulador (AC), un registro de entrada (INPR), un registro de instrucciones (IR), un registro general, un registro de salida (OUTR); y un sistema de bus para direcciones y datos. La figura 1 muestra el diagrama de bloques e interconexión de los componentes del procesador. Una vez realizada la

interconexión de dichos componentes, se integran en un solo circuito mediante el uso del simulador.

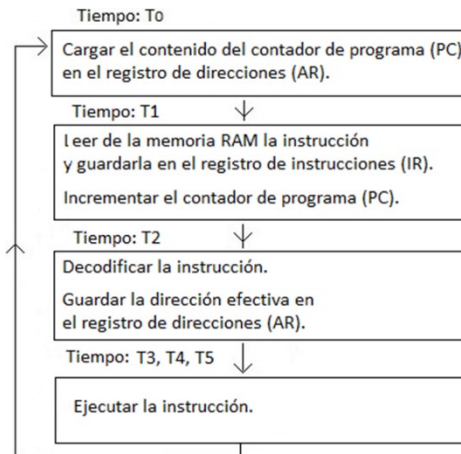


Fuente: [Mano, 1994].

Figura 1 Componentes del procesador, basado en la arquitectura de [Mano, 1994].

La unidad de control es la encargada de generar las señales necesarias para que se realice el ciclo de instrucción (búsqueda, decodificación y ejecución de instrucciones) como se describe en [Tanenbaum, 2000]. La unidad de control empleada es del tipo cableada y está basada en la descripción de para la arquitectura de un procesador básico [Mano, 1994]. La unidad de control está compuesta por un circuito generador de señales de temporización (T0, T1, ..., etc.), para indicar el momento en que se deben realizar ciertas transferencias y microoperaciones del procesador; un decodificador de instrucciones que genera las señales (D0, D1, ..., etc.) que indican la instrucción que se debe ejecutar; y de una serie de circuitos digitales combinacionales, que a partir de las salidas del decodificador de instrucciones y del temporizador, generan las señales de control hacia los diferentes componentes del procesador, tales como escritura o lectura de la RAM, habilitación de un registro para carga o incremento, tipo de operación en la ALU, etc.

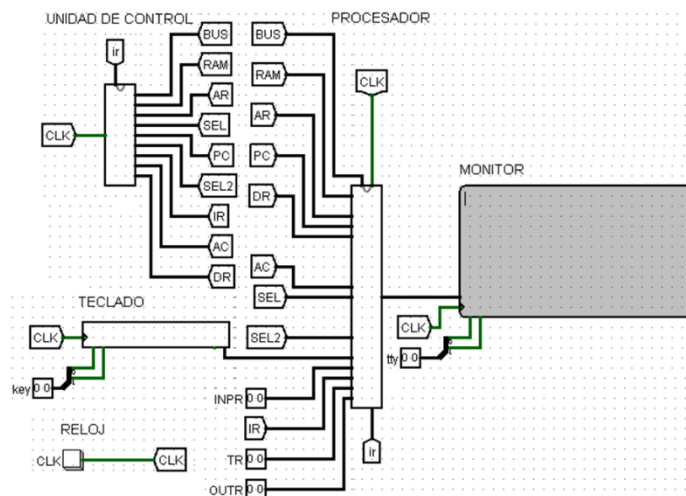
Una vez realizada la interconexión de los componentes de la unidad de control, se integran mediante el simulador en un solo circuito. En la figura 2, se muestra el ciclo de instrucción y se indica el intervalo de tiempo en que ocurren las microoperaciones necesarias para completar cada fase del ciclo de instrucción.



Fuente: elaboración propia.

Figura 2 Ciclo de instrucción.

En la figura 3, se muestra el circuito principal, conformado por el procesador y la unidad de control. Se incluyó como elemento de entrada un teclado y como elemento de salida un monitor. En la parte inferior se encuentra un botón pulsador, que se emplea como pulso de reloj del sistema.



Fuente: elaboración propia.

Figura 3 Circuito principal.

La unidad de control se encuentra conectada al procesador mediante túneles indicados por las etiquetas BUS, RAM, AR, etc. Mediante el simulador, es posible observar los componentes internos del procesador y de la unidad de control, para verificar de esta manera, el valor de las señales de control y el contenido de los registros, memoria y demás componentes del sistema; así como verificar el momento en que se generan las señales de control y el efecto que producen.

El formato de instrucción del procesador, el conjunto de instrucciones y el proceso de ensamblado está basado en la descripción de [Mano, 1994]. En la tabla 1, se muestra el conjunto de instrucciones implementado para el procesador. Se incluyen instrucciones que hacen uso de la memoria RAM, instrucciones que se ejecutan sobre un registro e instrucciones de entrada y salida. También es posible cambiar el conjunto de instrucciones de acuerdo con las operaciones que es capaz de realizar la ALU.

Tabla 1 Conjunto de instrucciones.

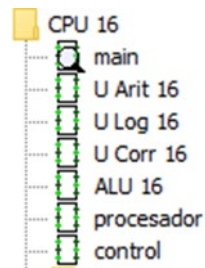
Mnemotécnico	Descripción
AND	Operación lógica AND entre el acumulador y un dato de memoria.
ADD	Operación aritmética de suma entre el acumulador y un dato de memoria.
LOAD	Cargar el acumulador con un dato de memoria.
STORE	Almacenar el contenido del acumulador en la memoria.
JMP	Salto incondicional.
SUB	Operación aritmética de resta entre el acumulador y un dato de memoria.
OR	Operación lógica OR entre el acumulador y un dato de memoria.
CLEAR	Limpiar el registro acumulador.
COM	Complemento a 1 del registro acumulador.
SHR	Corrimiento a la derecha del registro acumulador.
SHL	Corrimiento a la izquierda del registro acumulador.
INC	Incremento en 1 del registro acumulador.
DEC	Decremento en 1 del registro acumulador.
HALT	Detener el procesador.
IN	Cargar el acumulador con un dato de entrada.
OUT	Mostrar el contenido del acumulador a la salida.

Fuente: elaboración propia.

Para verificar las señales generadas por la unidad de control durante el ciclo de instrucción, el momento y las condiciones en que se generan; se almacenó en la memoria RAM manera independiente, cada una de las instrucciones del procesador y se aplicaron los pulsos de reloj al sistema, verificando en el subcircuito correspondiente el valor de las señales de control. Posteriormente, a partir de un

programa ensamblado y almacenado en la memoria RAM, se verificaron las señales básicas generadas por la unidad de control a fin de que se ejecuten las instrucciones del programa.

En la figura 4, se muestran los subcircuitos que además del teclado y el monitor, conforman el circuito principal: el procesador, la unidad de control y la ALU. De cada subcircuito es posible mostrar los detalles de su implementación.



Fuente: elaboración propia.

Figura 4 Subcircuitos que conforman el circuito principal.

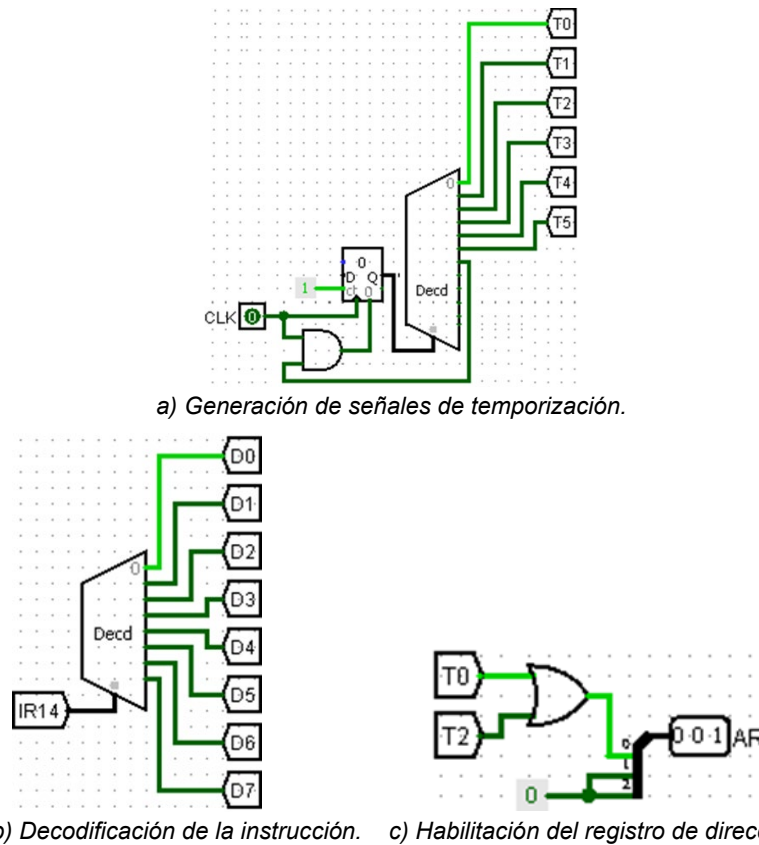
3. Resultados

Mediante el uso del simulador Logisim se realizó la implementación de un procesador básico de 16 bits y su unidad de control del tipo cableada. Se identificó cada etapa del ciclo de instrucción realizada por el procesador, la función de cada etapa, es decir, las transferencias entre los registros y los diferentes componentes del procesador; y el momento y las condiciones en que se ejecuta el ciclo de instrucción.

En la figura 5, se muestra una parte de la unidad de control con las señales correspondientes a la primera etapa del ciclo de instrucción. Se puede observar la señal de temporización T0 (Figura 5a), y la señal D0 generada por el circuito decodificador de instrucciones (Figura 5b). Dentro de la primera fase del ciclo de instrucción, también se genera la señal de habilitación del registro de direcciones (AR) para poder cargar la dirección del contador de programa (figura 5c).

Se probó la ejecución del ciclo de instrucción de cada una de las 16 instrucciones del procesador. En la figura 6, se muestran los componentes internos del procesador relacionados con la ejecución del ciclo de instrucción correspondiente a

la instrucción AND. La instrucción se encuentra ensamblada como 0002 Hx y está almacenada en la localidad 001 Hx de la memoria RAM, el valor del dato es 5 decimal y se encuentra almacenado en la localidad 002 Hx. El contenido del acumulador es 3 decimal.

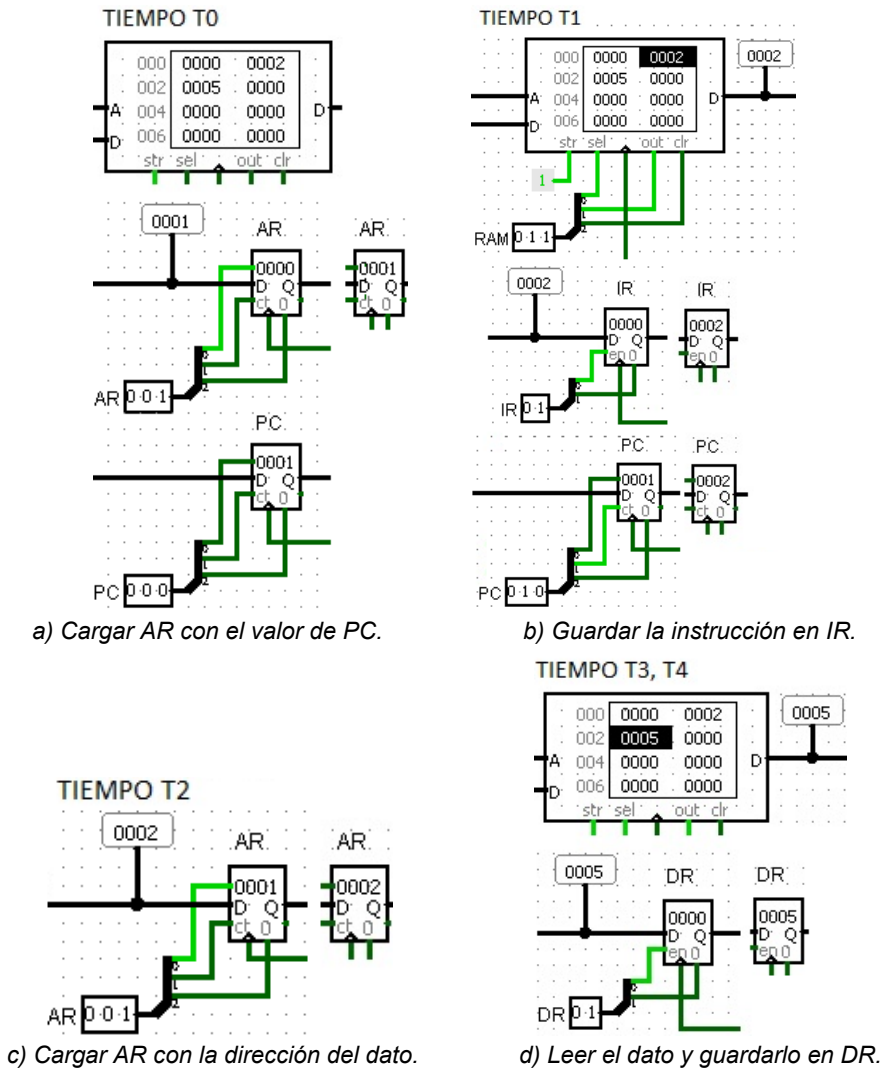


Fuente: elaboración propia.

Figura 5 Generación de las señales de la unidad de control (continuación).

En la figura 6a, se muestra el contenido de los componentes relacionados a la primera fase del ciclo de instrucción correspondiente al periodo de tiempo identificado como T0. Se puede observar el contenido de la memoria RAM, el contenido del contador de programa con la dirección de la instrucción (001 Hx), y el registro de direcciones habilitado para cargar la dirección de la instrucción, a un costado se muestra el contenido del registro de direcciones con la dirección cargada (001 Hx). En la figura 6b, se muestra el contenido de los componentes relacionados a la segunda fase del ciclo de instrucción correspondiente al periodo de tiempo

identificado como T1. Se puede observar que la memoria RAM se encuentra habilitada para la lectura de la instrucción (0002 Hx), también se observa que el registro de instrucciones se encuentra habilitado para almacenar la instrucción, y que el contador de programa se encuentra habilitado para incrementar su valor, indicando la próxima instrucción a ejecutar. A un costado del registro IR y PC se muestra su contenido al completarse esta fase.



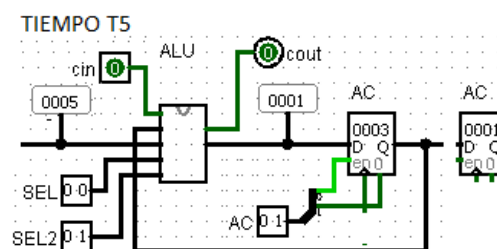
Fuente: elaboración propia.

Figura 6 Ejecución del ciclo de instrucción.

En la figura 6c, se muestra el contenido de los componentes relacionados a la tercera fase del ciclo de instrucción correspondiente al periodo de tiempo

identificado como T2. Se puede observar que el registro de direcciones se encuentra habilitado para obtener la dirección del dato (002 Hx). A un costado del registro AR se muestra su contenido al completarse esta fase. En la figura 6 inciso d, se muestra el contenido de los componentes relacionados a la cuarta fase del ciclo de instrucción correspondientes al periodo de tiempo T3 y T4. En el que se lee el dato de la memoria y se guarda en el registro de datos. Se puede observar que la memoria RAM se encuentra habilitada para la lectura del dato (0005) y que el registro de datos se encuentra habilitado para almacenar el dato. A un costado del registro DR se muestra su contenido al completarse esta fase.

En la figura 7, se muestra el contenido de los componentes relacionados a la etapa final de la ejecución de la instrucción correspondiente al periodo de tiempo identificado como T5. Se pueden observar los operandos de entrada a la ALU, el contenido del registro de datos (0005) y el contenido del registro acumulador (0003), así como el resultado de la operación AND (0001). También se observa que el registro acumulador se encuentra habilitado para almacenar el resultado. A un costado del registro AC se muestra su contenido al completarse esta fase.



Fuente: elaboración propia.

Figura 7 Ejecución de la instrucción AND.

Se realizó la prueba de un par de programas ensamblados y almacenados en la memoria RAM para verificar las señales básicas generadas por la unidad de control a fin de que se ejecuten las instrucciones del programa, así como verificar el resultado final del programa. En la tabla 2, se muestra el programa para la suma de dos números almacenados en memoria, guardando el resultado en la memoria. En la columna de la izquierda, se puede observar la dirección de cada instrucción y la dirección de los datos. En la columna de la derecha se muestra el valor en

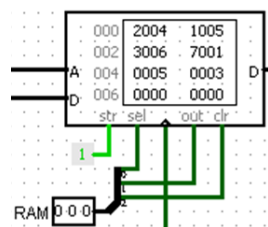
hexadecimal de cada una de las instrucciones ensambladas o el valor en hexadecimal de los datos tal como quedan almacenados en la memoria.

Tabla 2 Programa de la suma de dos números almacenados en memoria.

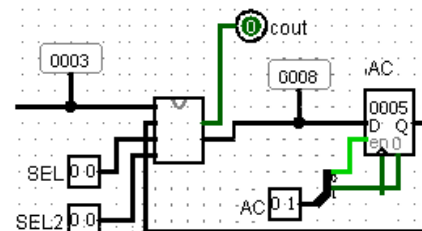
Dirección (Hx)	Contenido de la memoria (instrucción o dato)	Descripción del contenido de la memoria	Instrucción ensamblada o dato (Hx)
0000	LOAD A	Cargar el acumulador con un dato (5) de memoria.	2004
0001	ADD B	Sumar el contenido del acumulador (5) con un dato (3) de memoria.	1005
0002	STORE C	Guardar el contenido del acumulador (8) en la memoria.	3006
0003	HALT	Detener el programa.	7001
0004	0005	Número 5	0005
0005	0003	Número 3	0003
0006	0000	Resultado	0000

Fuente: elaboración propia.

La figura 8 muestra algunos componentes del procesador relacionados con la ejecución del programa. La figura 8a, se muestra el contenido de la memoria RAM, se pueden observar las localidades con su respectiva dirección de memoria, con el programa ensamblado en las direcciones 000 a 003, y los datos en las direcciones de memoria 004 y 005. El resultado de la ejecución del programa se almacena en la dirección 006. Para cada una de las instrucciones, se realiza el ciclo de instrucción de manera semejante al proceso mostrado en la figura 6. En la figura 8b, se muestra el resultado al final de la ejecución de las instrucciones. Se pueden observar los dos operandos de entrada a la ALU (0005 y 0003), así como el resultado de la suma (0008) a la salida de la ALU; también se puede observar la señal de habilitación del registro acumulador para almacenar el resultado en el siguiente pulso de reloj.



a) Programa y datos almacenados en RAM.



b) Ejecución de la operación suma.

Fuente: elaboración propia.

Figura 8 Ejecución del programa de la suma de 2 números.

En la tabla 3, se muestra el programa para la suma de dos números ingresados por teclado, mostrando el resultado en el monitor. En la columna de la izquierda, se puede observar la dirección de memoria para cada instrucción y la dirección de los datos del programa. En este programa los valores a sumar son ingresados por teclado en código ASCII, por lo que, en el programa, se realiza el ajuste de ASCII a decimal mediante la resta del valor 30 hexadecimal a los valores numéricos de entrada y la suma del valor 30 hexadecimal a los datos de salida hacia el monitor.

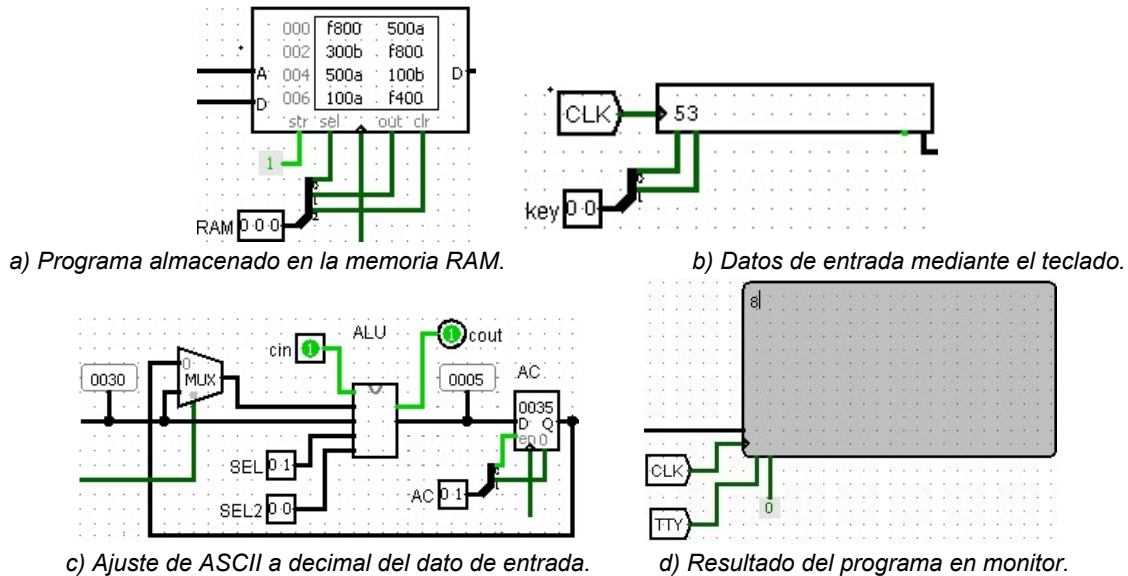
Tabla 3 Programa de la suma de dos números ingresados por teclado.

Dirección (Hx)	Contenido de la memoria (instrucción o dato)	Descripción del contenido de la memoria	Instrucción ensamblada o dato (Hx)
0000	IN	Cargar el acumulador con un dato de entrada en código ASCII.	F800
0001	SUB A	Realizar el ajuste de ASCII a decimal mediante la resta.	500A
0002	STORE B	Guardar el dato en la memoria.	300B
0003	IN	Cargar el acumulador con un dato de entrada en código ASCII.	F800
0004	SUB A	Realizar el ajuste de ASCII a decimal mediante la resta.	500A
0005	ADD B	Sumar datos ingresados.	100B
0006	ADD A	Realizar el ajuste de decimal a ASCII mediante la suma.	100A
0007	OUT	Mostrar el resultado en el monitor.	F400
0008	HALT	Detener el programa.	7001
0009	0000	Localidad vacía.	0000
000A	0030	Valor para realizar el ajuste	0030
000B	0000	Localidad para el dato de entrada.	0000

Fuente: elaboración propia.

En la figura 9, se muestran algunos los componentes del procesador relacionados con la ejecución del programa, tales como, la memoria RAM, la ALU, el teclado, y el monitor de salida. Mediante la instrucción IN, se ingresa el carácter en código ASCII que se ha pulsado en el teclado, se almacena en el registro de entrada INPR, se pasa a través de la ALU sin modificarse para quedar registrado en el registro acumulador. En la figura 9a, se muestra el contenido de la memoria RAM con el programa ensamblado. En la figura 9b, se muestran los datos ingresados mediante el teclado (5 y 3 respectivamente). Para cada una de las instrucciones, se realiza el ciclo de instrucción de manera semejante al mostrado en la figura 6. En la figura 9, se muestra una parte de la ejecución del programa en la se realiza el ajuste ASCII

(0035 Hx) a decimal (0005) mediante una resta en la ALU, del primer dato ingresado por teclado; también se puede observar la señal de habilitación del registro acumulador para almacenar el resultado en el siguiente pulso de reloj. En la figura 9d, se muestra en el monitor el resultado (8) de la ejecución del programa.



Fuente: elaboración propia.

Figura 9 Suma de 2 números ingresados por teclado.

4. Discusión

El trabajo está enfocado en la implementación de un procesador simplificado y su unidad de control, de la cual se pretende verificar mediante la simulación, los conceptos teóricos del funcionamiento básico del procesador.

En los libros de texto, que se considera que cubren los temas del procesador y la unidad de control de manera más didáctica, no se incluye la implementación física o simulación de los conceptos. En otro libro de texto se incluye la simulación componentes digitales, pero no se abarca el tema de procesador. En este trabajo se contribuye a mejorar el balance teórico y práctico de los conceptos del procesador y la unidad de control.

Como se mencionó en la revisión de artículos, existen otros desarrollos didácticos con procesadores más complejos que pueden ejecutar programas y mostrar los valores de los registros del procesador, pero no permiten mostrar componentes

internos del procesador, las señales internas de la unidad de control hacia los componentes del procesador, ni mostrar el momento en que ocurren y el efecto que producen. En este trabajo se muestra que, mediante el uso del simulador, es posible verificar el comportamiento de los componentes internos del procesador y las señales de control.

Para la implementación del procesador y la unidad de control, se consideró un pequeño conjunto de instrucciones. Para incluir una mayor cantidad de instrucciones, se debe modificar en la unidad de control, el circuito lógico que genera las señales de control de los componentes del procesador que se involucran la ejecución de la instrucción. El procesador y la unidad de control implementados en este trabajo tienen un enfoque didáctico, e incluyen las características que permiten mostrar de manera visual los conceptos teóricos básicos tales como el ciclo de instrucción, señales de control y componentes de un procesador. Se puede considerar que es factible incluir características o funciones un poco más complejas al procesador presentado, tales como el ciclo de interrupción, diferentes modos de direccionamiento y una mayor variedad de instrucciones.

5. Conclusiones

Mediante el uso del simulador Logisim es posible implementar un procesador básico de 16 bits y su unidad de control del tipo cableada. En este trabajo se contribuye a mejorar el balance teórico y práctico de los conceptos del procesador y la unidad de control.

Se verificaron mediante el uso del simulador, los conceptos teóricos tales como, componentes y función del procesador y la unidad de control; así como el ciclo de instrucción y las señales de temporización y control. Se mostró de forma didáctica qué ocurre en el internamente en el procesador, y se verificaron las condiciones y el momento en que se generan las diferentes señales de control, requeridas para la ejecución de las instrucciones de un programa ensamblado y almacenado en la memoria RAM.

Se implementó el procesador para un conjunto de 16 instrucciones y es viable ampliar las funciones del procesador y el conjunto de instrucciones.

Para el desarrollo de este trabajo no se requirieron mayores costos en equipo de laboratorio, o el aprendizaje de lenguaje de descripción de circuitos.

6. Bibliografía y Referencias

- [1] Burch, C. (2011). Logisim (version 2.7.1) [software]. Obtenido de Logisim español. Hendrix College. <http://www.cburch.com/logisim/es/index.html>
- [2] CPUlator computer system simulator. (28 de septiembre de 2023). <https://cpulator.01xz.net/>.
- [3] Gurrola N. (2022). Diseño de procesador RISC-V de 32-bits de ciclo único. *Pistas Educativas*, Vol.44 (No. 143), 473-491. ISSN:2448-847X.
- [4] Hanafi M., Kurniawan W. (2017). Design and implementation 8 bit CPU architecture on Logisim for undergraduate learning support. 2017 International Conference on Sustainable Information Engineering and Technology. DOI:101109/SIET.20178304123.
- [5] Hanafi M., Kurniawan W. (2020). CPU implementation using only Logisim simulator to achieve computer architecture learning outcome. *Bulletin of Electrical Engineering and Informatics*. Vol. 9 (No.2), 747-754. ISSN: 2302-9285, DOI:1011591/eei.v9i2.1972.
- [6] Hernández C., Hernández A., Sánchez J. (2017). Apoyo didáctico del lenguaje VHDL en el estudio de una unidad de control microprogramada. *Pistas Educativas*, Vol. 39 (No. 127), 243-254. ISSN:2448-847X.
- [7] Hernández, C. (2019). Uso del simulador Logisim como apoyo didáctico en la material de Arquitectura de Computadoras. *Pistas Educativas*, Vol. 41 (No.133), 55-69. ISSN:2448-847X.
- [8] Pahrami B. (2007). *Arquitectura de Computadoras. De los microprocesadores a las supercomputadoras*. México: Prentice Hall. ISBN: 0-19-515455-X.
- [9] Prasad P., Alsadoon A., Beg A., Chan A. (2015). Using simulators for teaching computer organization and architecture. *Computer Applications in Engineering Education*. DOI: 10.1002/cae21699.

- [10] Mano, M. (1994). *Arquitectura de Computadoras* (3a Ed.). México: Pearson Educación. ISBN: 968-880-361-8.
- [11] Sadad N., Afrin A., Mandal M. (2022). SP-1 Design and simulation of 4-bit simple CPU on Logisim for Computer Architecture Education. 2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE). DOI: 10.1109/ICECTE57896.2022.10114523.
- [12] Salido J. (2023). *Lógica digital y tecnología de computadores. Un enfoque práctico mediante simulación con Logisim*. España: Ediciones de la Universidad de Castilla – La Mancha. ISBN: 978-84-9044-621-8.
- [13] Tanenbaum A. (2000). *Organización de Computadoras. Un enfoque estructurado* (4a Ed.). México: Prentice Hall. ISBN: 970-17-0399-5.
- [14] Tecnológico Nacional de México campus Instituto Tecnológico Superior de Irapuato. (2023). *Ingeniería en Sistemas Computacionales*. Obtenido de Oferta Educativa https://irapuato.tecnm.mx/moferta/sistcomputacionales/pdf/plan_estudios/5%20Arq%20de%20Comp.pdf.
- [15] Yepes V. (2003). Simuproc (version 1.4.2.0) [software]. <http://simuproc.cbj.net>.