

Sistema automático de recolección y clasificación de información basado en circuitos lógicos.

Pablo Dellacqua, Santiago Enrique Sánchez y Francisco Tomás Yeroncich

Facultad Regional Santa Fe, Universidad Tecnológica Nacional,
Lavaisse 610, S3004EWB Santa Fe, Argentina
{pablodellacqua, santiagosanchezutn,
francisco.yeron12}@gmail.com

Resumen. En la materia Arquitectura de Computadoras de primer año de Ingeniería en Sistemas de Información en la Facultad Regional de Santa Fe de la Universidad Tecnológica Nacional, se pide a los estudiantes realizar un trabajo práctico para fortalecer sus habilidades en el diseño de circuitos lógicos combinacionales, secuenciales y mecanismos de sincronización. El problema planteado consistió en elaborar una propuesta que automatice la recolección y clasificación de información en un caso de estudio específico. Se debía manejar una estructura matricial para leer y almacenar información, junto a un mecanismo para desplazar dicha información a la cabecera de cada fila de la matriz para que sea retirada por un recolector. El recolector simula un robot que recupera toda la información de la matriz y la coloca en una estructura similar a una cinta transportadora para ser entregada en forma paralela a un clasificador que la procesa en forma secuencial. Como solución se definieron e implementaron los circuitos necesarios utilizando el software Logisim Evolution. Además, se diseñó una interfaz gráfica que ofrece una abstracción de la implementación de los circuitos, lo que permite visualizar el recorrido de la información en el sistema durante la simulación con el acompañamiento de un panel informativo y la posibilidad de navegación detallada del circuito.

Keywords: Circuitos lógicos combinacionales. Circuitos lógicos secuenciales. Mecanismos de sincronización. Logisim Evolution.

1 Introducción

El presente trabajo se desarrolló como práctico de la cátedra Arquitectura de Computadoras de la carrera Ingeniería en Sistemas de Información, en el cual se solicitó el desarrollo de un modelo basado en circuitos lógicos combinacionales y secuenciales que permita simular la recolección y clasificación de huevos en una granja de producción agroecológica. Se espera que el modelo ayude a realizar una propuesta de automatización de la tarea de recolección y clasificación que actualmente se realiza en la granja de forma manual.

Para dar solución al problema planteado elaboramos un circuito que inicialmente simula la acción de una gallina poniendo un huevo en un nido, lo cual define la entrada de nuestro modelo. Tal como fue solicitado en la consigna del problema, los

nidos se organizaron usando una estructura matricial y la recolección de los huevos se realizó desde la cabecera de cada fila. Todos los huevos recolectados se trasladan en una cinta transportadora, que se implementó para que reciba todos los huevos recolectados de los nidos y los entregue en forma paralela a un circuito que se encarga de contarlos y clasificarlos en forma secuencial.

En el circuito que desarrollamos incorporamos mecanismos de sincronización que nos permitieron modelar la puesta de los huevos, recolección, traslado y clasificación. Además, se agregó un tablero en el cual se puede ir viendo la información actualizada del sistema a medida que los huevos van recorriendo el circuito.

2 Elementos de trabajo y metodología

2.1 Organización

La consigna del trabajo tenía el requisito de confeccionar un cronograma o planificación de tareas necesarias para concretar la resolución del mismo (ver Fig.1).

	Nombre de tarea	Cantidad de personas	Tiempo máximo en días	Fecha de inicio	Depende de las tareas	Integrante 1	Integrante 2	Integrante 3
1	Reunión presencial para definición de pautas y criterios de trabajo.	3	1	24/10/2022	-	Planteo de pautas, necesidades y cualidades personales para la realización del trabajo.	Planteo de pautas, necesidades y cualidades personales para la realización del trabajo.	Planteo de pautas, necesidades y cualidades personales para la realización del trabajo.
2	Crear una planificación de proyecto en Excel.	3	1	29/10/2022	T1	Planificación y definición de tareas.	Planificación y definición de tareas.	Planificación y definición de tareas.
3	Creacion de archivos a Utilizar.	3	1	29/10/2022	T2	Creación y planteamiento del Excel grupal.	Creación y planteamiento del informe (word) y del Excel grupal.	Creación y planteamiento del archivo de Logisim y Excel grupal.
4	Reunión virtual para planteamiento del problema e identificación de elementos que necesitamos para la resolución del mismo.	3	1	29/10/2022	-	Lluvia de ideas/planteamiento lógico.	Lluvia de ideas/planteamiento lógico.	Lluvia de ideas/planteamiento lógico.
5	Realización del NP en el proyecto.circ	1	2	30/10/2022	T4	-	-	Creación de matriz, con manejo de información interna (controles de digitación simultánea y lógica de escritura).
6	Realización del CH en el proyecto.circ	2	1	31/10/2022	T4	Toma de información desde NP hacia CH y habilitaciones (paralelo).	Conversión de información a serie con comunicación a RR. Reseteo de NP's.	-
7	Consulta Martes 1/11	1	1	1/11/2022	T5 y T6	-	-	Ir con las dudas y problemas que no pudimos resolver.
8	Realización del RR en el proyecto.circ	2	3	3/11/2022	T4	Creación de los registros que va a utilizar el robot para el almacenamiento de datos y la preparación para la recepción de	-	Administración y manipulación de la información acumulada en los registros para el intercambio con la(s)

Fig. 1. Planificación de Tareas: vista parcial del conjunto de tareas planificadas inicialmente.

Además, cumplimos con el requisito de mantener actualizada una bitácora de avances de trabajo que nos ayudó a ver claramente el estado de las tareas planificadas y, así, pudimos coordinar y realizar nuestro trabajo como equipo. Este nivel de detalle nos permitió entender la necesidad de replanificar al detectar un desvío significativo en el desarrollo de alguna de las tareas.

2.2 Diseño de la solución

La consigna del trabajo agrupaba funcionalidades identificando los niveles A, B y C. A continuación, indicamos qué contempla el diseño de cada uno de los niveles y damos detalles del comportamiento de los circuitos que intervienen.

Nivel A

Incluye el modelado de los nidos de las ponedoras (NP) y la recolección de los huevos (ver Fig.2). Los NP se implementaron usando una estructura de matriz de dimensión 4x4, que nos permitió trabajar con 16 nidos. Cada nido puede tener 1 huevo ó 0 huevo.

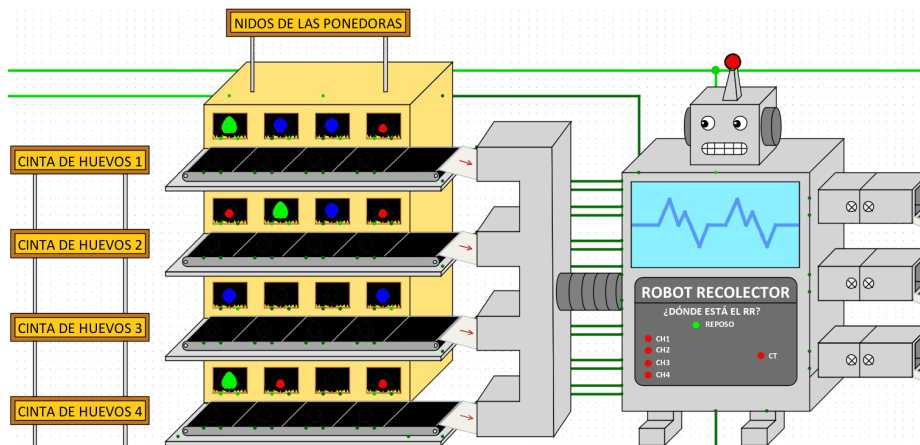


Fig. 2. Vista del Nivel A.

Modelamos visualmente a los huevos usando puntos (leds). Los distintos puntos permiten clasificar a los huevos según su tamaño: punto rojo = huevo chico, punto azul = huevo mediano y punto verde = huevo grande.

Bajo cada fila de la matriz, se encuentra una Cinta de Huevos (CH) que transporta los huevos que bajan de los nidos. Cada CH es la encargada de transportar los huevos hacia el Robot Recolector (RR). Las CH no reciben huevos hasta que el RR da la señal de que se encuentra en posición.

El RR comienza la recolección de la cabecera de la fila superior y, una vez que recolectó todos los huevos posibles, pasa a la fila de abajo y así sucesivamente hasta haber recolectado todos los huevos de la matriz de nidos.

Nivel B

Incluye la entrega de los huevos que realiza el RR a la Cinta Transportadora (CT).

La consigna tenía como requisito que la CT entregue en forma paralela 3 huevos a la vez al circuito Contador y Clasificador (CC) y es por ello que a la CT la implementamos utilizando 3 líneas para transportar a los huevos (ver Fig.3).

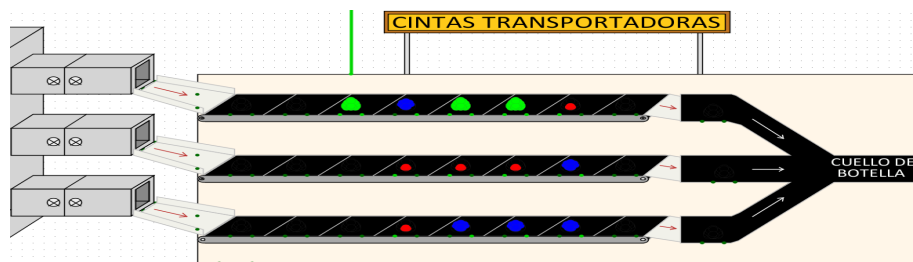


Fig. 3. Vista del Nivel B.

Al final de la CT se implementó un mecanismo de conversión de paralelo a serie que permitió entregar al CC los huevos en forma secuencial.

Nivel C

Incluye la clasificación realizada por el CC que consiste en llevar un total por tipo de huevos y un total general. Esta información se muestra en un tablero (ver Fig.4).

Desde este panel informativo el usuario puede interactuar con el sistema pulsando los botones de INICIAR, PAUSAR/REANUDAR o APAGAR.

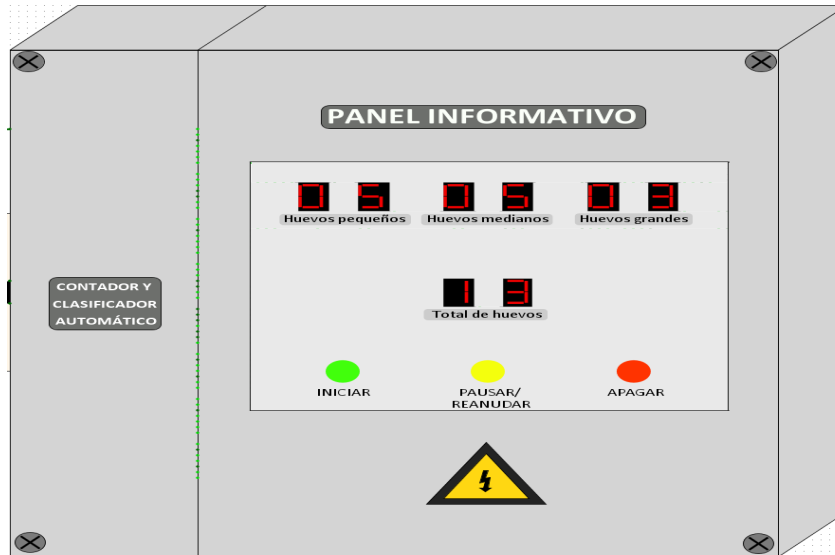


Fig. 4. Panel informativo.

2.3 Diseño de la solución

Para la realización del circuito seleccionamos el simulador con el cual trabajamos en las clases prácticas de la materia Logisim Evolution (versión 3.8.0)[1]. Logisim Evolution es una herramienta que permite construir y simular circuitos lógicos digitales mediante una interfaz gráfica de usuario. Una vez profundizado el conocimiento sobre el simulador, definimos cómo administrar las versiones de cada circuito construido y comenzamos con su implementación optando por diseñar todos los circuitos a nivel de compuertas y biestables.

3 Desarrollo detallado

En este apartado mostramos detalladamente los circuitos intervinientes en la solución de cada uno de los niveles. Es acá donde se muestra en forma explícita todo lo construido en cuanto a circuitos lógicos combinacionales y secuenciales [2][3][4], como así también los mecanismos de sincronización y manejo de información. La integración de todos los niveles en un sistema unificado la trataremos más adelante.

3.1 Nivel A

Creamos el circuito que modela a los NP. En este circuito en los nidos 'Nidos' (ver Fig.5); cargamos la información de asociada a los huevos utilizando un contador que recorre los nidos del 1 al 16 y dos *randomizadores* que envían bits al azar y cargan la matriz de nidos, disponiendo de todos los tipos de huevos.

Un control de digitación simultánea no permite que existan errores a la hora de hacer ingresos de huevos a los nidos y así cumplimos con el requerimiento de que

cada nido tenga a lo sumo un huevo hasta ser recolectado. Además, se le adosó un convertor de decimal a binario para los números de nido.

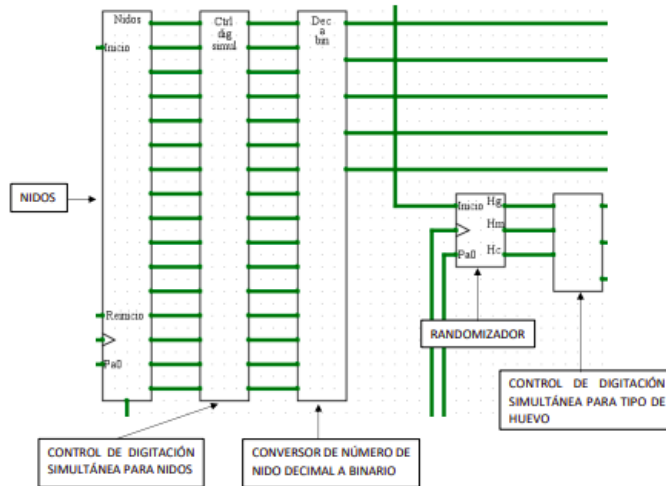


Fig. 5. Carga de huevos en cada nido.

Incorporamos un control de digitación simultánea para las entradas referidas a los tipos de huevos tomando la siguiente convención de bits: 00 = no huevo, 01 = huevo chico, 10 = huevo mediano, 11 = huevo grande.

Cargados los huevos, en el circuito “LógicaNP” (ver Fig.6 en Anexo), redireccionamos las entradas a biestables que cumplen la función de almacén para los huevos. Además, este módulo alimenta a los leds que representan a los huevos ubicados en los nidos, convierte el tipo de huevo a binario y carga los huevos de los nidos a cada CH.

Cada CH (ver Fig.7 en Anexo) recibe la información en paralelo y tiene una capacidad máxima de transportar 4 huevos en forma secuencial al RR. Utilizamos un registro de desplazamiento para implementar cada CH.

El RR (ver Fig.8 en Anexo) recibe en serie los huevos de las distintas CHs y los almacena, cabe resaltar que el robot recibirá los huevos de las CHs a medida que éstas se van vaciando, proceso que logramos con la creación de un ‘Marcapaso’, el cual habilita la siguiente posición cuando la actual está vacía.

3.2 Nivel B

Creamos el módulo de la CT (ver Fig.9), el cual consta de 3 registros de desplazamiento (FilaCT) en paralelo que reciben información del RR y desembocan en un circuito llamado ‘Conversor3a1’ (ver Fig.10) que se encarga de pasar la información de 3 líneas en paralelo a una en serie.

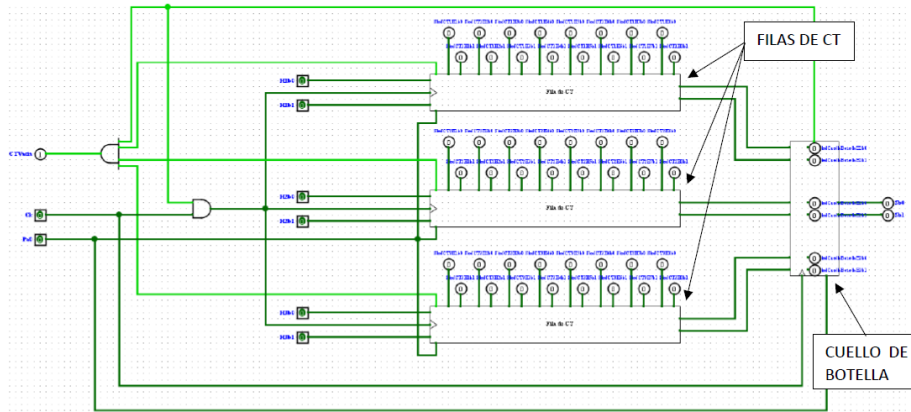


Fig. 9. Circuito de la CT.

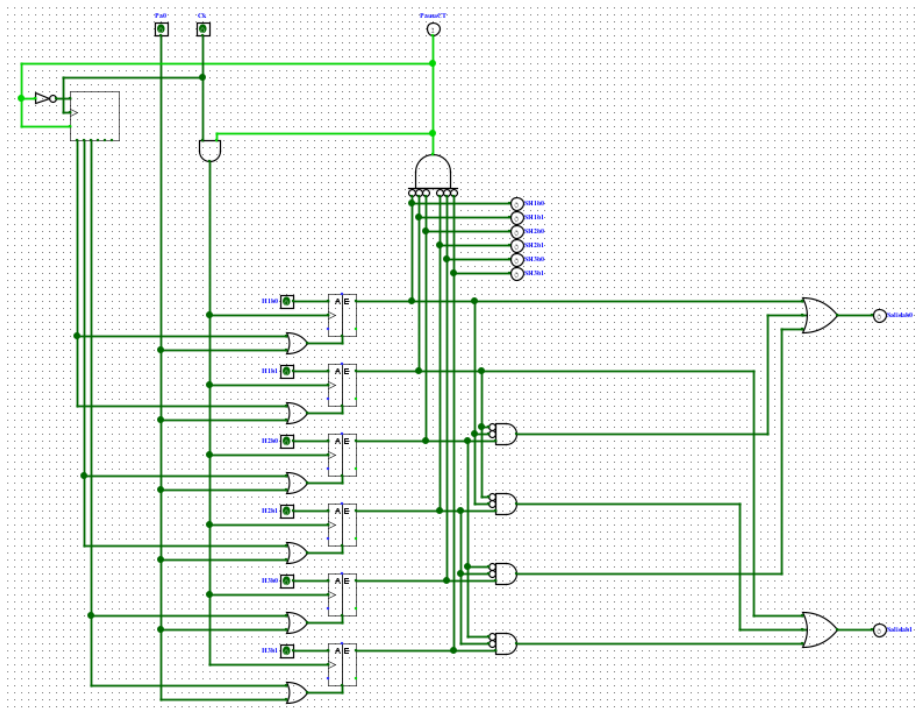


Fig. 10. Circuito Conversor3a1.

3.3 Nivel C

Creamos el circuito del CC (ver Fig.11) que recibe los huevos provenientes de la CT en serie y según el tamaño los va contabilizando.

El CC consta de 4 módulos de contadores binarios naturales, uno por cada tipo de huevo y uno para la cantidad total, que contabilizan cada tipo dependiendo de la restricción/habilitación en su entrada. Cada módulo contador puede contar hasta 99 huevos y el valor del conteo se muestra en el panel informativo con displays de 7 segmentos (ver Fig.12).

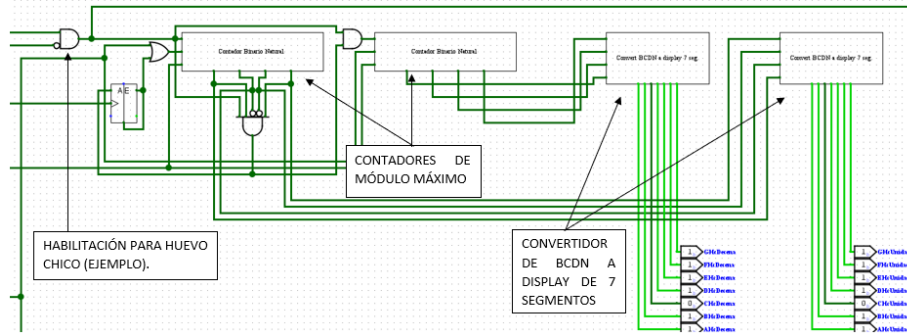


Fig. 11. Circuito del CC.

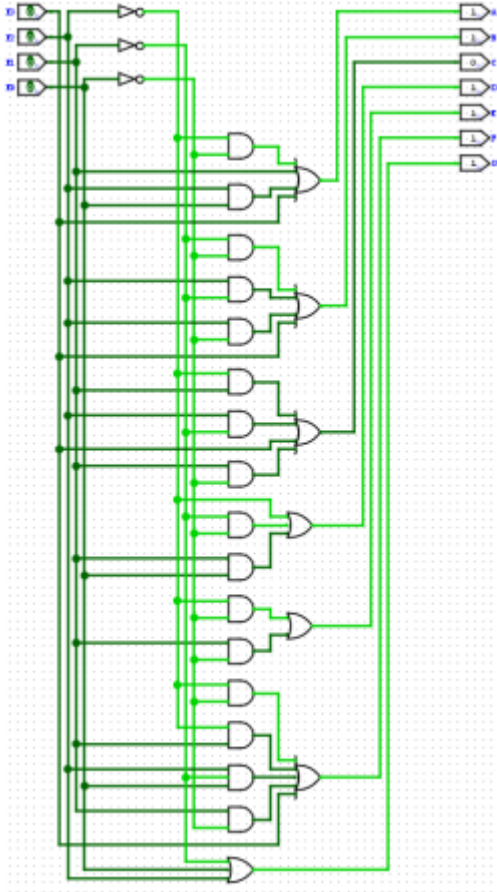


Fig. 12. Circuito conversor BCDN a 7-segmento.

4 Integración de la solución por niveles

La integración de los niveles (ver Fig.13) se vio facilitada dado que desde el inicio planeamos las interfaces para conectar los distintos niveles. En todo momento fuimos muy organizados en torno a los nombres de entradas y salidas de los circuitos, para poder mantener un orden y convención general a utilizar durante todo el trabajo. Por ejemplo, todos los cables que manejan información sobre los huevos tienen en su nombre la letra 'H', seguida del número correspondiente al huevo manejado y de la sigla 'bX' donde la X toma el valor del bit de huevo trabajado (bit 1 o bit 0). Entonces, una posible entrada de huevo siguiendo este formato podría ser 'H1b0', haciendo referencia al bit 0 del huevo 1 (en la primera fila de la matriz, por ejemplo). Esta última parte de integración se realizó en el circuito principal, donde, además, se incorporaron los diseños de la representación visual de las partes y niveles. A fin de poder mostrar una secuencia de ejecución grabamos y editamos un video[5].

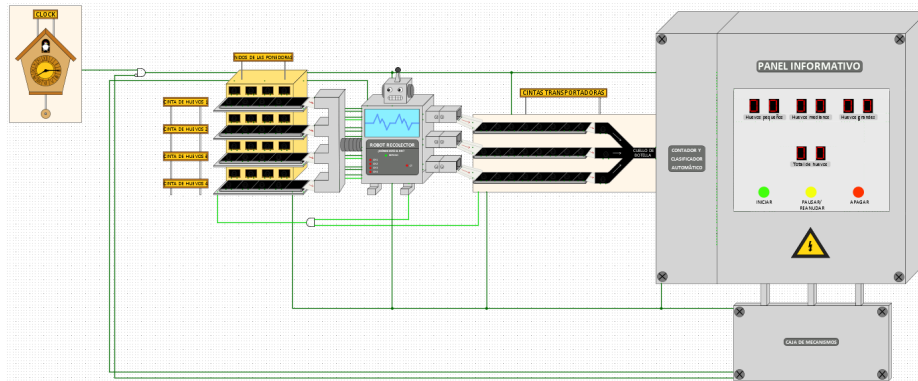


Fig. 13. Circuito principal.

5 Resultados

Desde un comienzo y a lo largo de todo el trabajo, realizamos pruebas para poder verificar y validar el correcto funcionamiento del sistema. Estas pruebas, que consistían en correr los distintos circuitos analizando entradas y salidas, nos permitieron eliminar errores e imperfecciones (como el borrado accidental de datos, la errónea generación de bits en momentos incorrectos entre otros) hasta poder llegar al resultado deseado.

Las pruebas finales realizadas las enfocamos en comparar la información aleatoria cargada en la matriz inicial y los correspondientes resultados mostrados en el panel, y mostraron que el sistema trabaja correctamente.

6 Discusión

Posiblemente podemos considerar que fue un obstáculo para la realización de un sistema más eficiente el hecho de que lo desarrollamos mientras aprendíamos los conceptos de la materia, lo cual dio lugar a que realicemos circuitos y acciones innecesarias o redundantes para el funcionamiento correcto del sistema. Una de ellas fue por ejemplo, que en el circuito general NP realizamos una conversión de 5 bits a 16 bits, para luego de pasar por otros circuitos y volver a los 5 bits con el fin de transmitir la información. Si bien ahora entendemos que esto es innecesario, nosotros lo hicimos en su momento porque pensamos que sería bueno para poder verificar que no haya situaciones donde se cargue la información de 2 nidos a la vez u otras posibles cuestiones que den lugar a fallas.

En términos de extensión del trabajo, una idea pendiente es la implementación y aplicación de un contador de pasadas para lograr que el circuito se reinicie en forma automática actualizando información en el panel existente en el sistema.

7 Conclusión

El trabajo que desarrollamos cumple con lo solicitado por la cátedra de Arquitectura de Computadoras y, como además pudimos profundizar nuestro conocimiento sobre la herramienta, nos permitió incorporar diseño de una interfaz, que al encapsular funcionalidades facilitan el seguimiento visual de la simulación.

El trabajo resultó una experiencia positiva para nosotros dado que nos permitió afianzar muchos conocimientos de la materia, y al ser un trabajo en grupo nos vimos en la necesidad de mantener una buena comunicación entre todos, realizando múltiples reuniones, lo cual nos impulsó y animó a mantener una buena cursada de nuestro primer año universitario.

References

1. Logisim Evolution, <https://github.com/logisim-evolution/logisim-evolution/releases>.
2. Null, L. Lobur J.:The essentials of Computer Organization and Architecture.5ta. ed., Jones and Bartlett Publishers. Burlington, USA(2019).
3. Quiroga P.: Arquitectura de Computadoras.1a. ed., Alfaomega.Buenos Aires (2010).
4. Murdocca, M. Heuring, V.:Principios de arquitecturas de computadoras.1era.ed., Pearson Education. Buenos Aires (2002).
5. Enlace al video que muestra la ejecución del circuito con un caso de prueba, <https://youtu.be/EgIYZAoJAx8>.

Anexo

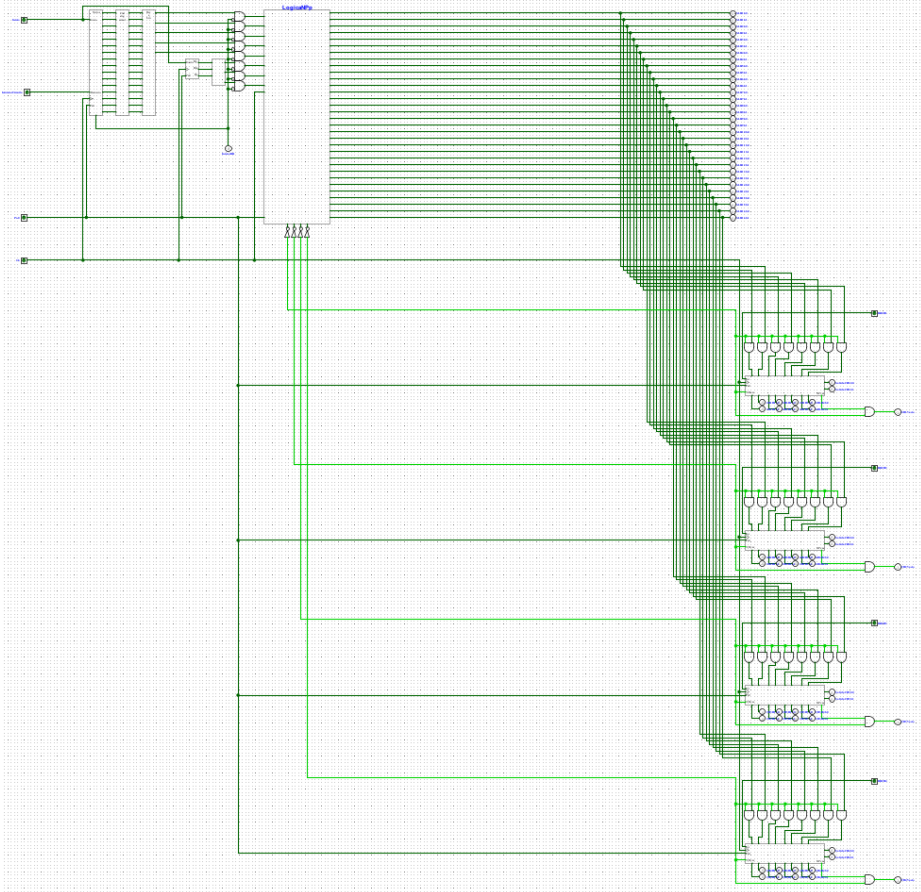


Fig.6. Circuito NP.

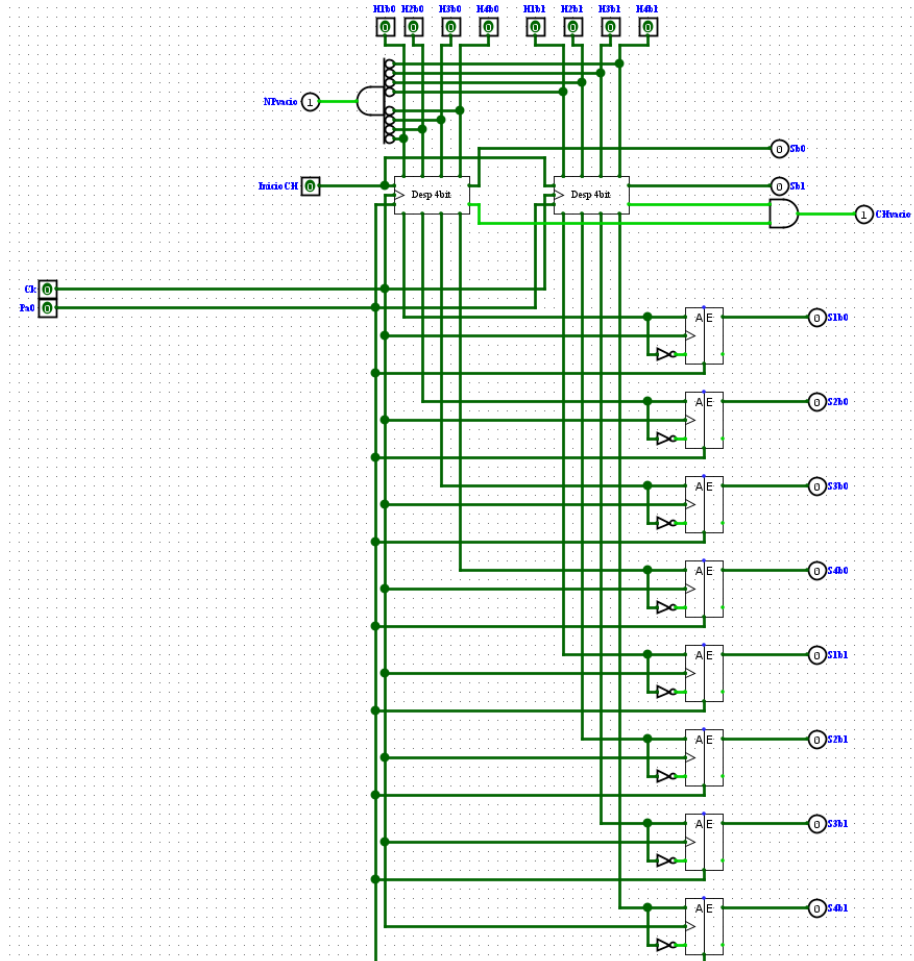


Fig.7. Circuito CH.

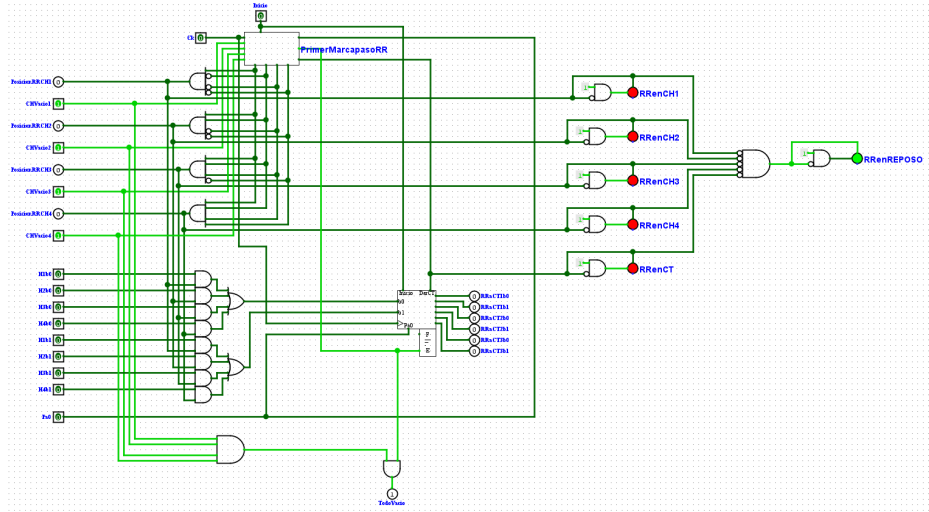


Fig.8. Circuito RR.