



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

# Informe Tarea 1

## Circuito Combinacional

### **Grupo 22**

Martín Arancibia Alvarado 201973517-9  
Miguel Soto Delgado 201973623-K

October 11, 2022

## Contents

1	Desarrollo de la tarea	1
2	Movimiento Básico	1
3	Estado de Carga	2
4	Resultados y análisis	3
5	Conclusiones	7

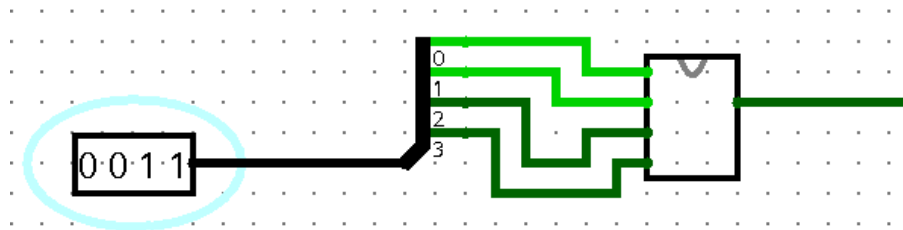
# 1 Desarrollo de la tarea

## 2 Movimiento Básico

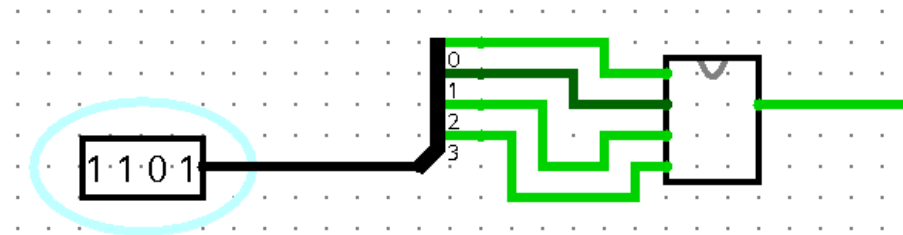
Primero modelamos el movimiento del robot a partir de lo indicado en el enunciado, en donde tomamos las siguientes variables; si el robot tiene más de 3 cargas (B), si hay enemigos a su izquierda, frente o derecha (Ei, Ef, Ed) y finalmente si hay estaciones de carga a su izquierda, frente y derecha (Ci, Cf, Cd) en exactamente ese orden. Todo esto produciendo las distintas salidas; ir a la izquierda, ir al frente e ir a la derecha (L, F, R). Debido a que no existía otras especificaciones a lo largo de la tarea, se asumirán los siguientes supuestos:

- En caso de que no hayan unidades de carga ni enemigos, el ExaBot se moverá según la prioridad dada por su estado de carga, es decir, la prioridad que tengan los movimientos vendrán dados por B (Ej: no existe nada en ninguno de los 3 caminos y el robot tiene 2 cargas, por lo que este se moverá hacia adelante).
- El Exabot siempre intentará moverse, en el caso que haya un camino disponible, esto incluye los casos donde este tiene una carga menor a 3.
- Como 0 es un número menor que 3, el ExaBot podrá moverse con una carga de 0, incluso en el caso que no existan unidades de carga alrededor. Este caso solo hará que se comporte de igual manera que para el 1, 2 y 3.

Para empezar, definiremos los eventos de manera que sean entendibles dentro de los parámetros que nos ofrece la herramienta; la batería (B) tendrá un valor de 0 siempre que su carga sea igual o menor a 3, y un valor de 1 en cualquier otro caso. Esto se logró mediante el siguiente circuito que está insertado a la derecha del input de 4 bits (Input el cual corresponde a la carga actual).



En este caso, la batería tiene un nivel de carga igual a 3, por ende su salida es 0.



En este otro caso, el nivel de la batería es 13, teniendo una salida igual a 1.

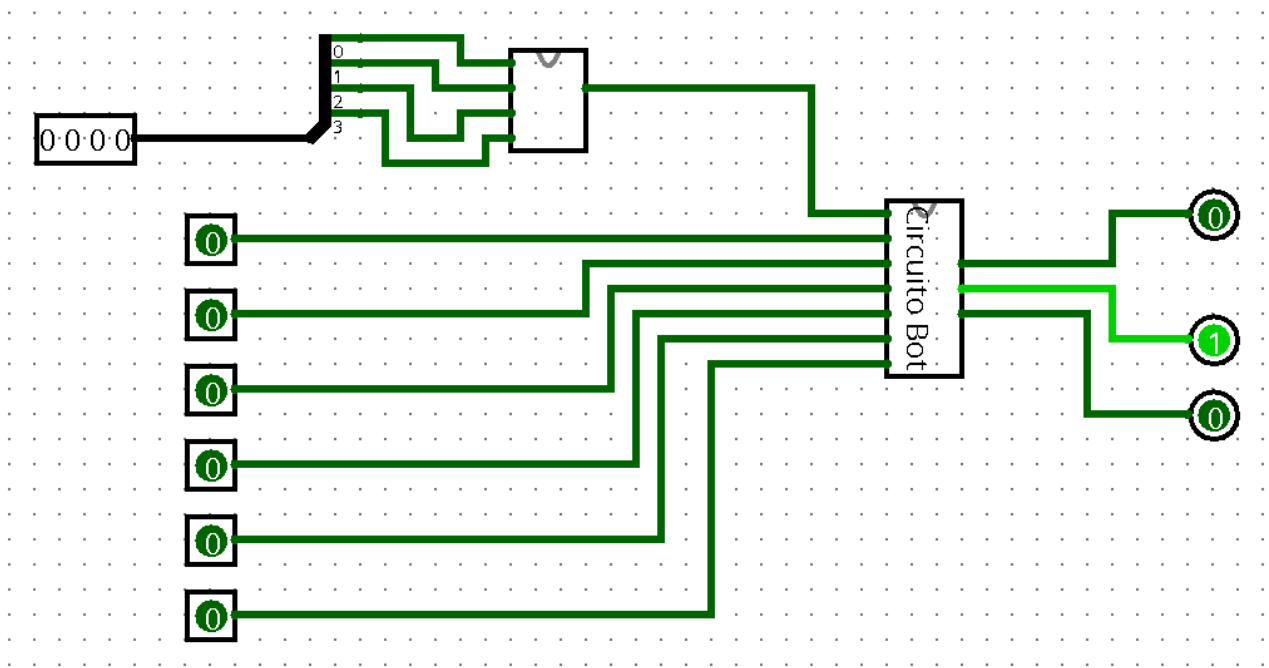
Una vez tenemos el estado actual de la batería, tomaremos los 6 pines que se encuentran a la izquierda y haremos inputs válidos (Reconoceremos como input válido cualquier input el cual no ponga en la misma posición una estación de carga y un enemigo) cambiando los valores de estos entre 0 y 1, 0 indicando que no existe un enemigo/estación de carga en esa posición y 1 indicando que existe un enemigo/estación de carga en esa posición. En caso de poner un input inválido la salida será siempre 0 0 0.

Ya con todos los datos ingresados al circuito, este revisará, por medio del subcircuito llamado "Circuito Bot", hacia que posición debe moverse nuestro bot dada las condiciones iniciales, y en ese momento saldrá, por medio de alguno de los pines de salida, la dirección tomada por el bot. El pin el cual quede con un 1 nos demarca en que dirección el bot se ha movido.

Haciendo el análisis de este circuito, tendremos que la expresión matemática para el circuito, y se define como:

$$\overline{B}E\overline{I}E\overline{f}C\overline{I}C\overline{f}C\overline{r} + \overline{B}E\overline{I}E\overline{r}C\overline{I}C\overline{f}C\overline{r} + \overline{E}\overline{I}E\overline{f}E\overline{r}C\overline{I}C\overline{f}C\overline{r} + \overline{B}E\overline{I}E\overline{f}C\overline{I}C\overline{f} + \overline{B}E\overline{I}E\overline{f}E\overline{r}C\overline{I}C\overline{f}$$

Y el circuito completo quedará de la siguiente manera:



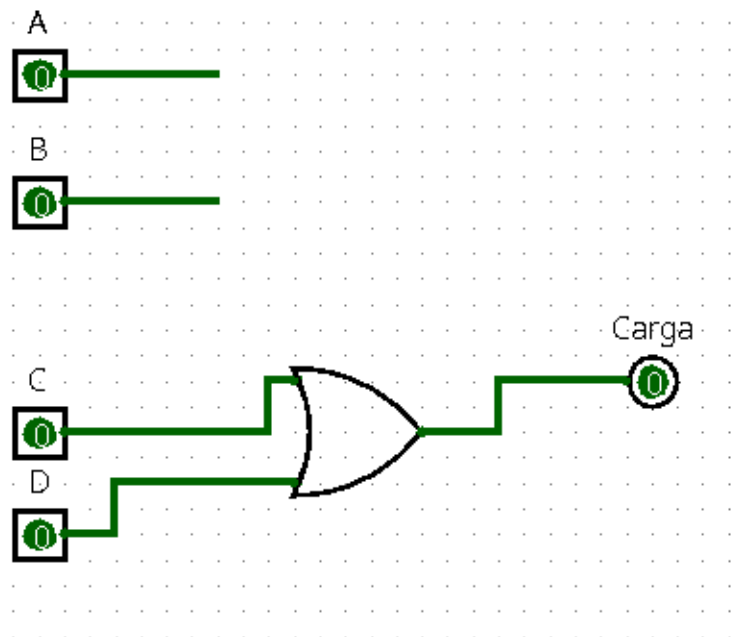
### 3 Estado de Carga

Para lograr lo anterior, se necesita saber en qué estado de carga está el bot, esto se logró haciendo un circuito para evaluar si se tiene la cantidad de energía necesaria (en este caso más de 3 cargas) o no cumple (tiene 3 o menos cargas). Cómo la batería se mide con una entrada de 4 bits, entonces basta con definir 4 entradas: A, B, C, y D, cada una representando un dígito binario, permitiéndonos modelar los 16 estados posibles de batería del bot. Finalmente, esta dispondrá de una única salida de 1 bit.

La ecuación del estado de la batería se define como:

$$C + D \quad (1)$$

Ya que solo nos interesa que el número de carga sea 4 o superior, así que en binario se toma el tercer o cuarto elemento de los cuatro bits.



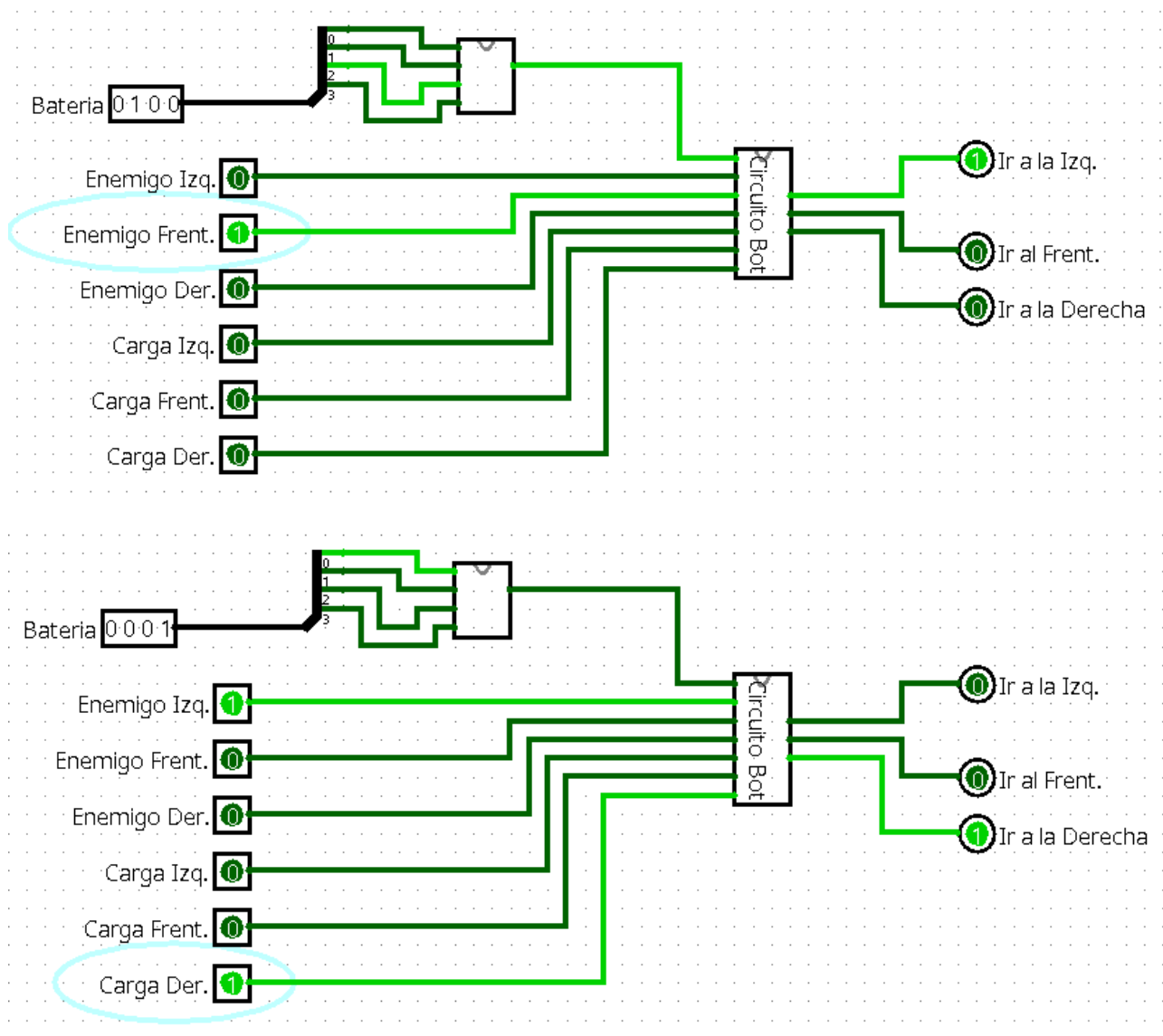
## 4 Resultados y análisis

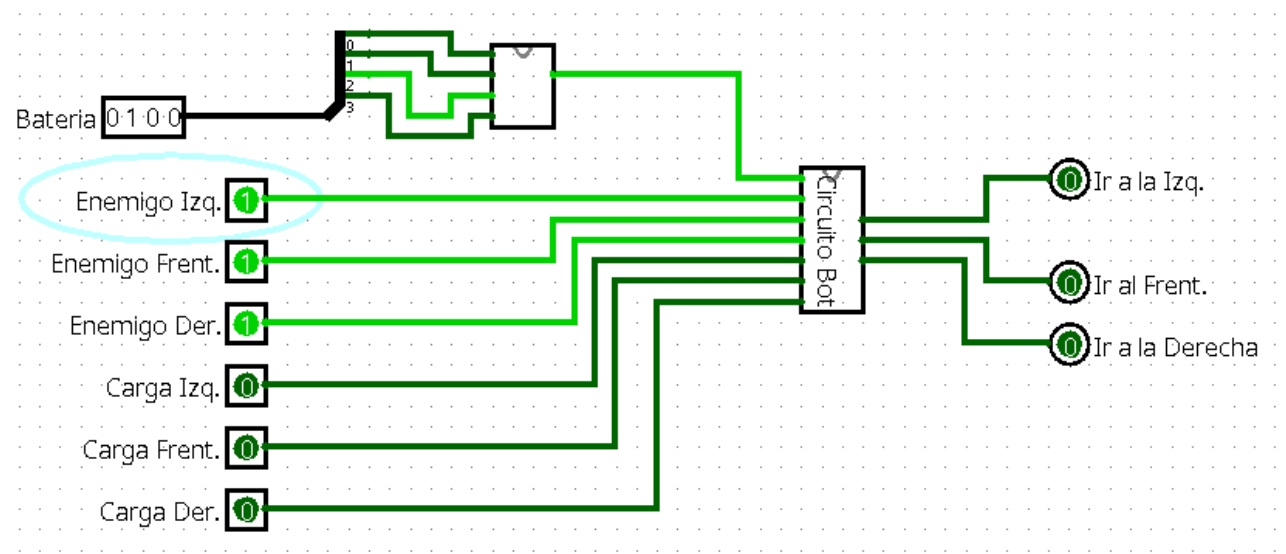
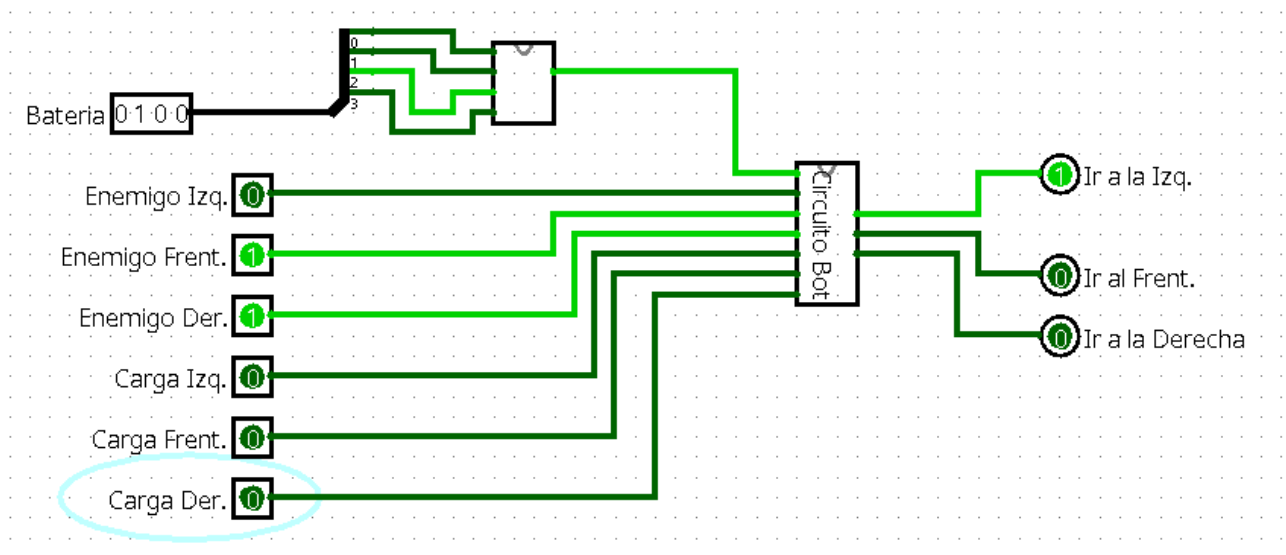
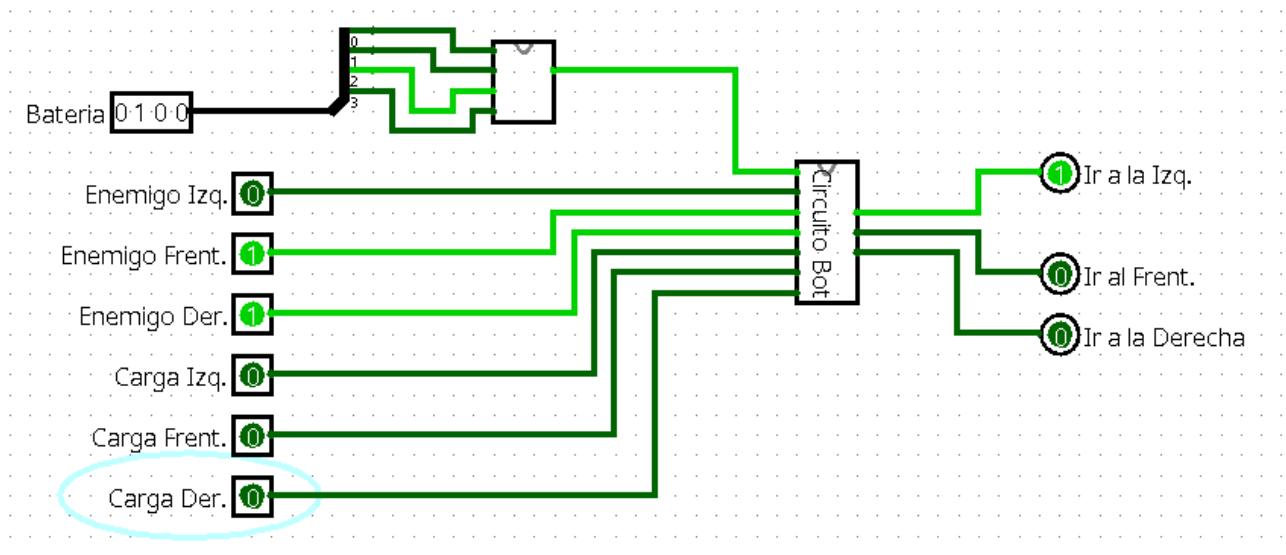
Para corroborar que el circuito del robot funciona acorde a la tarea, usamos los ejemplos presentes en el enunciado de la tarea, los cuales son los siguientes:

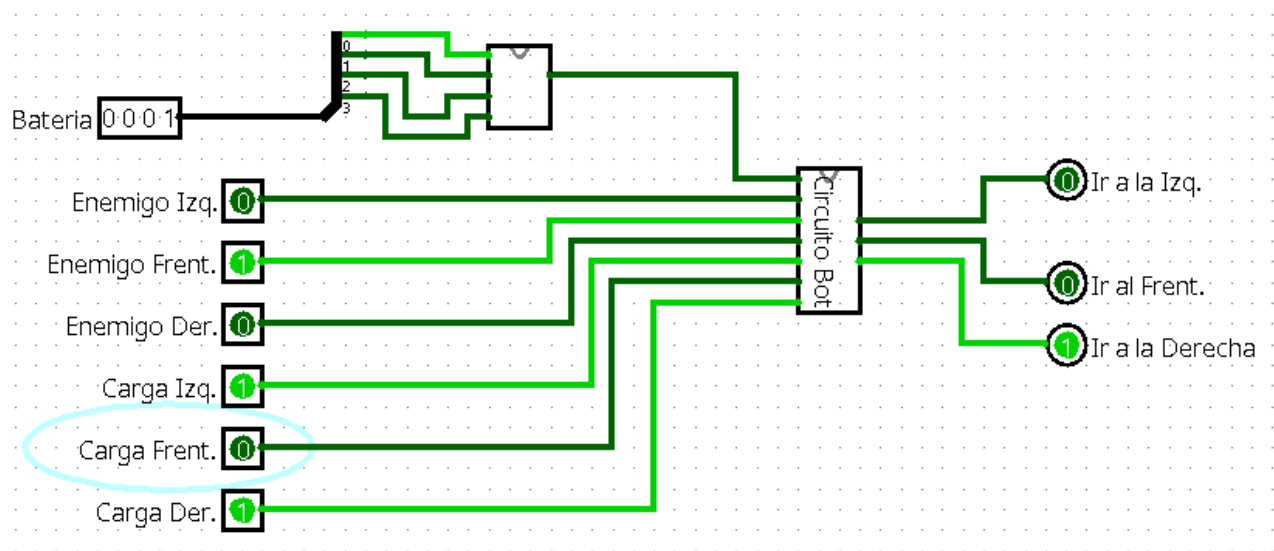
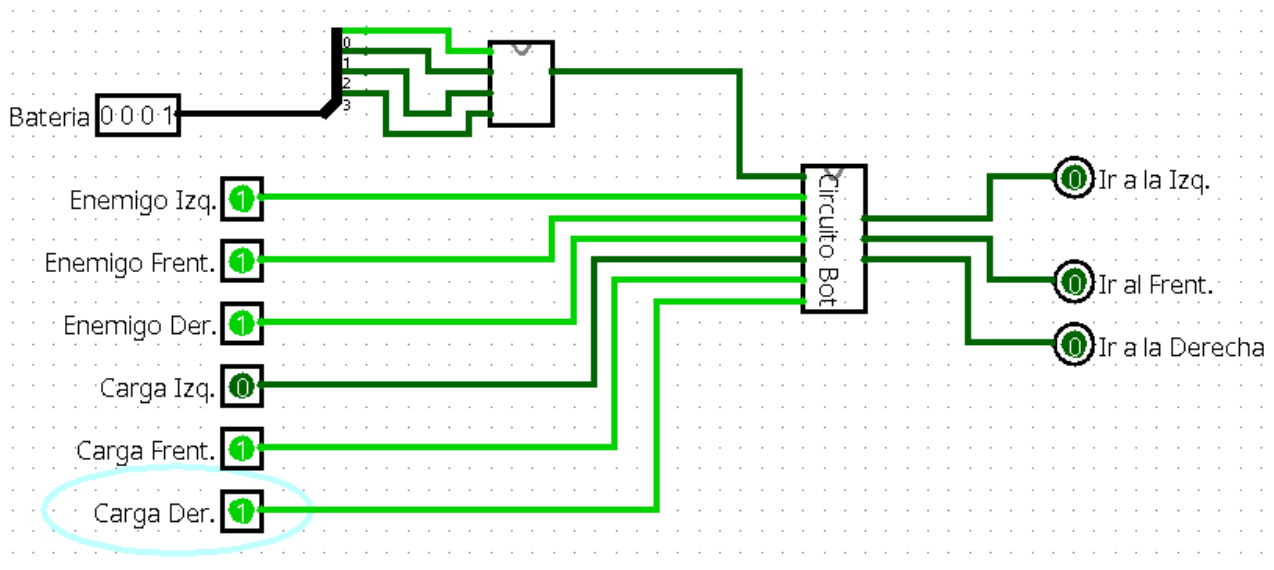
### Ejemplos

Entrada			Salida		
Batería (B)	Oponentes (E)	Recargas (C)	Izquierda (L)	Adelante (F)	Derecha (R)
0100	010	000	1	0	0
0001	100	001	0	0	1
0100	011	000	1	0	0
0100	111	000	0	0	0
0001	111	011	0	0	0
0001	010	101	1	0	1

En dichos casos, nuestros resultados fueron los siguientes:





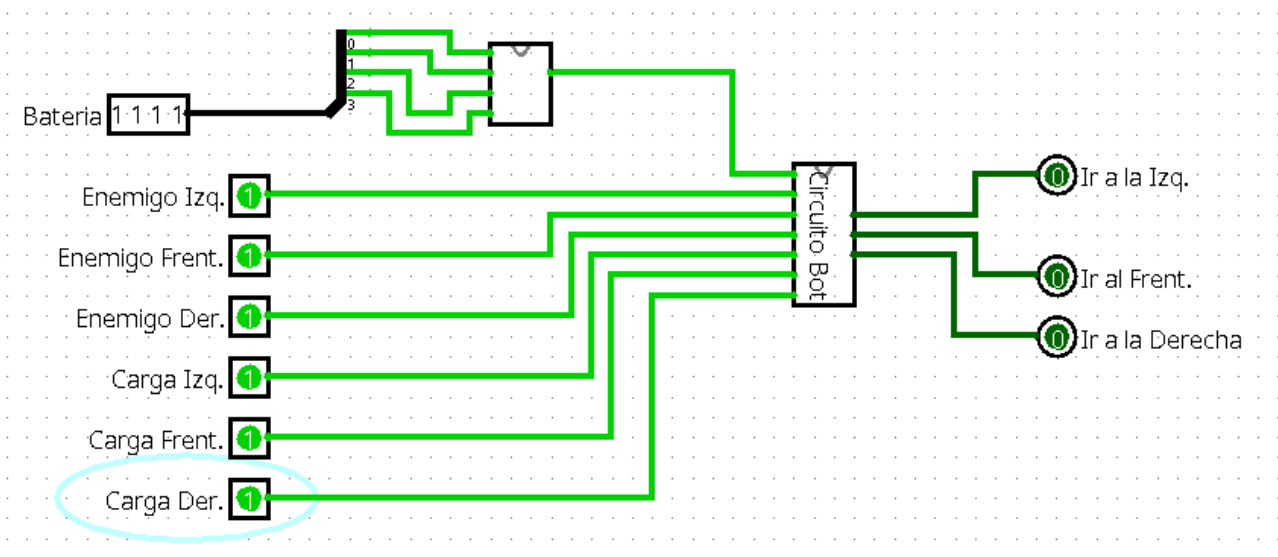
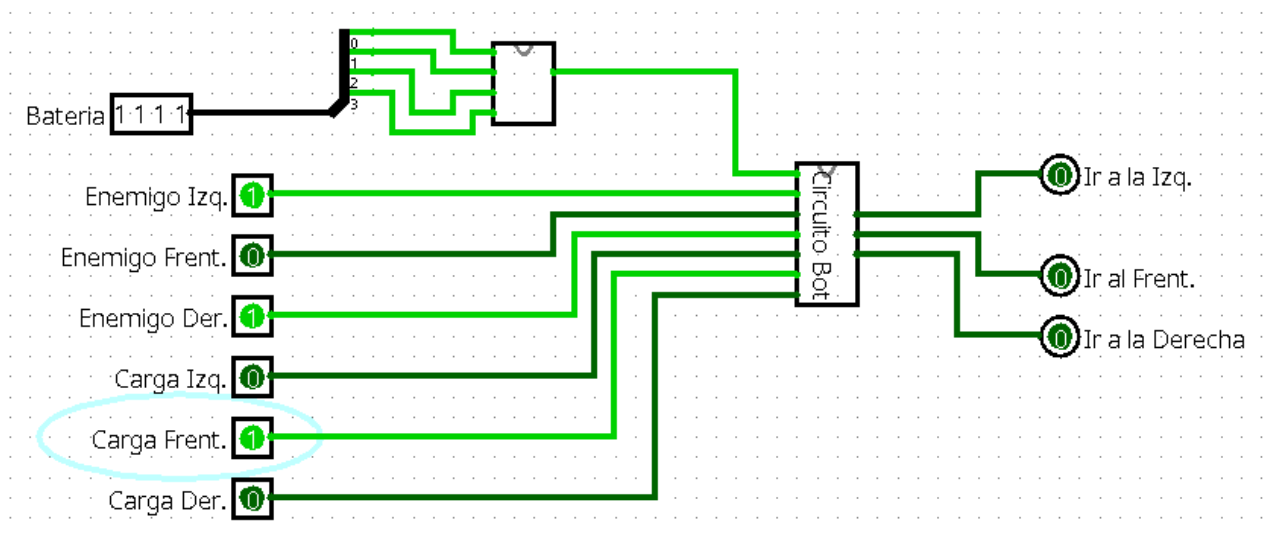


Al ver los casos en comparación al enunciado, notamos que solo el sexto caso difiere del resultado presentado. Esto se justifica, ya que el enunciado tiene un error, ya que el resultado debiese ser solo una de las direcciones presentadas. En nuestro caso el bot opta por ir a la derecha, porque dicha dirección tiene prioridad por sobre la izquierda al estar con el nivel de carga inferior o igual a 3.

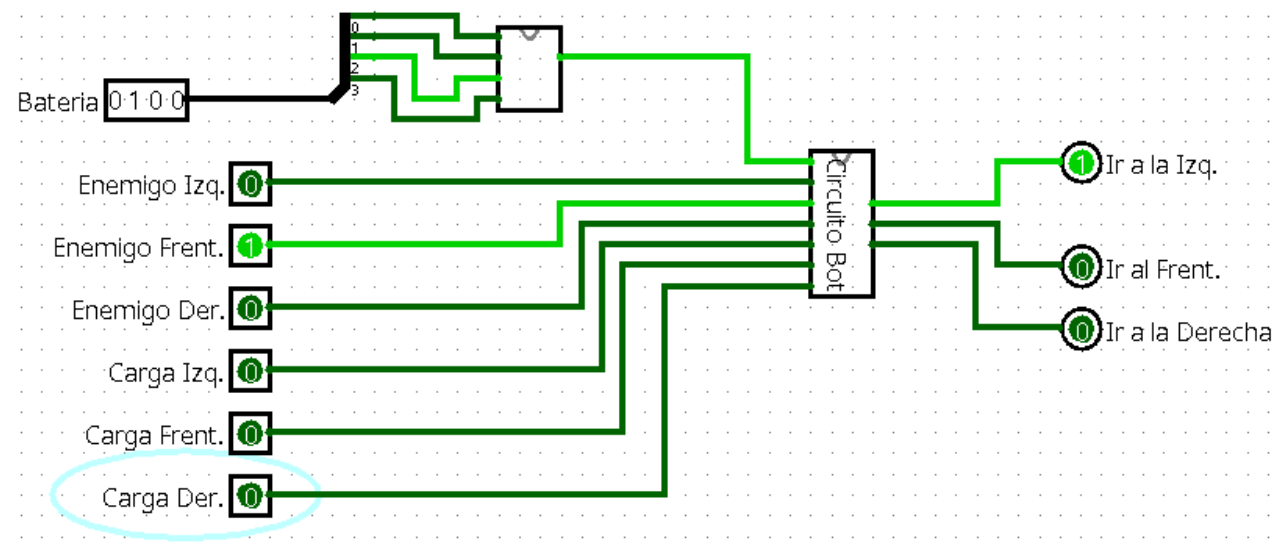
También a la hora de realizar el análisis lógico nos percatamos de que el circuito que representa el movimiento del bot era bastante complejo, ya que al poseer 7 entradas diferentes la cantidad de casos posibles subió exponencialmente (siendo la cantidad de casos igual a  $2^7 = 128$ ), dado esto nosotros decidimos no incluir dicha tabla dentro del desarrollo de la tarea y optamos solo por mostrar la operación lógica que describe dicho movimiento.

Otro incidente notable es que existían ciertos casos que (al menos bajo nuestra interpretación del problema) eran imposibles, en estricto rigor, aquellos casos en donde había más de un obstáculo en una misma posición. Para estos casos, optamos por asumir que el robot no se movería, debido a que no calza con ninguna de las condiciones mencionadas.

Más allá de esto, decidimos incluir otros casos más que sean consistentes con la lógica del movimiento del robot. En los siguientes casos se espera que el bot no sea mueva, ya sea porque son imposibles o porque se rigen por la lógica de prioridad:



Finalmente, uno de los casos representados en la figura del enunciado:





## 5 Conclusiones

Dando conclusión a este informe, podemos dejar constancia de que se cumplió el objetivo de aprender circuitos combinacionales a través del caso ejemplo proporcionado adjunto a la herramienta "Logisim". A la hora de realizar los análisis pedidos, el software nos facilitó de manera importante la labor, ya que contiene múltiples funcionalidades que hacen más ameno el desarrollo de un circuito que cumpla con las necesidades y parámetros lógicos anticipados.

Como grupo, podemos concluir que uno de los mayores problemas que enfrentamos fue el manejo de la tabla porque contenía una cantidad bastante considerable de datos, más allá de eso no tuvimos mayor inconveniente. La experiencia consideramos que fue bastante agradable, esto gracias a que la herramienta nos proporcionó varias facilidades y su naturaleza gráfica hace que una tarea que podría considerarse engorrosa se transforme en un ejercicio entretenido y amigable.