

Fecha de asignación: 15 de marzo de 2021

Fecha de entrega: 23:59 horas del 11 de abril de 2021

1. Objetivo

- Emplear los conocimientos teóricos adquiridos en el curso en el proceso de diseño de sistemas digitales combinacionales.
- Emplear herramientas de software para el diseño y la simulación de sistemas digitales.

2. Descripción

En esta práctica, cada equipo de trabajo debe diseñar y simular un sistema digital combinacional para realizar el control de movimiento de un robot primitivo, de manera que pueda seguir una trayectoria que lo lleve de un punto de partida a otro de destino, avanzando celda por celda, en una cuadrícula de 7×7 celdas. El robot está en capacidad de realizar cuatro acciones:

- Girar sobre su eje vertical 90° a la derecha y avanzar una celda (*Right + Go*).
- Girar sobre su eje vertical 90° a la izquierda y avanzar una celda (*Left + Go*).
- Avanzar una celda (*Go*).
- Detenerse (*Stop*).

En el proceso se debe emplear un estilo de diseño estructural y jerárquico, en el que se diseñan bloques básicos que luego son instanciados para crear otros más complejos y de mayor nivel en la jerarquía de diseño. La herramienta por emplear en este laboratorio será [Logisim Evolution v.3.4.1](#).

En la Figura 1 se muestra la arquitectura general del sistema a desarrollar. El diseño se debe abordar siguiendo los bloques funcionales recomendados en la arquitectura. Observe que en cada momento es posible apreciar la posición actual y la posición siguiente del robot en la cuadrícula, así como la acción a ejecutar para alcanzar esta última. Las posiciones actual y siguiente del robot se muestran en *displays* de siete segmentos. Por su parte, la

acción a ejecutar por el robot se visualiza en un conjunto de LEDs, uno por cada acción posible.

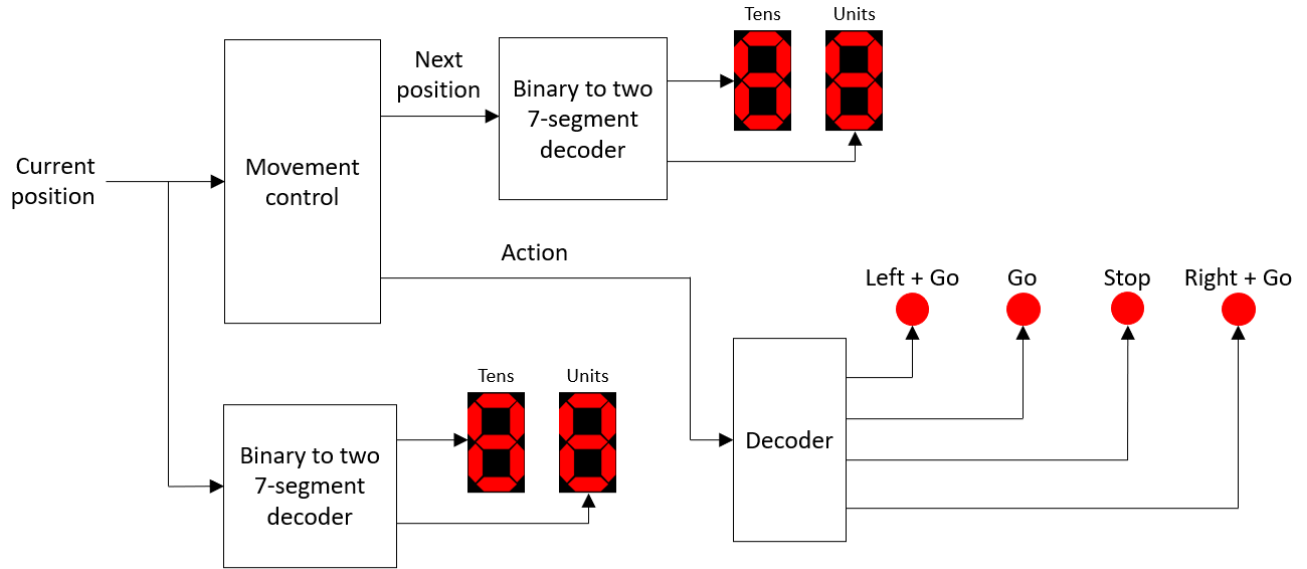


Figura 1. Arquitectura del sistema combinacional

La representación de los números en los *displays* de siete segmentos se debe ajustar a la convención presentada en la Figura 2.

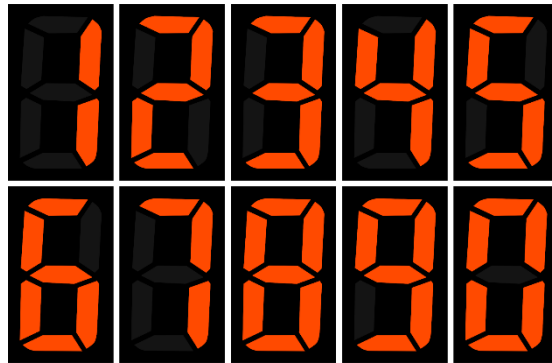


Figura 2. Convención para ilustrar los dígitos decimales en un display de siete segmentos

Cada equipo de trabajo tendrá un problema asignado, que consiste en una enumeración preestablecida para las celdas en la cuadrícula, códigos para las acciones del robot, un punto de partida y otro de destino, correspondiéndole determinar una trayectoria que lleve al robot del primero al segundo, tal como se explica en la sección 5 de esta guía.

El diseño de los bloques funcionales del sistema combinacional demandará partir de tablas de verdad. De ellas se deben obtener, empleando mapas de Karnaugh, las expresiones

Booleanas minimizadas que representan la funcionalidad a implementar mediante compuertas lógicas básicas **and**, **or** y **not**. Los bloques funcionales deben interconectarse usando buses para hacer manejable el diseño y la simulación.

El sistema combinacional tiene que ser simulado exhaustivamente con la misma herramienta para verificar su correcto funcionamiento.

Durante el proceso de diseño e implementación del sistema digital combinacional pueden presentarse diversas situaciones ante las cuales cada equipo debe tomar decisiones. En todos los casos, las decisiones de diseño deben estar ampliamente explicadas y justificadas.

3. Informe

Cada equipo de trabajo debe realizar un informe escrito que incluya una descripción completa del proceso de diseño: decisiones de diseño, tablas de verdad, mapas de simplificación, lista de implicants primos e implicants primos esenciales de cada mapa, funciones lógicas minimizadas, esquemáticos de los circuitos combinacionales, simulación exhaustiva del sistema, observaciones, análisis de resultados y conclusiones. Este documento debe estar en formato **PDF** y ser subido a la plataforma, acompañado de los archivos de diseño de **Logisim Evolution**, antes del cierre del plazo de entrega. El informe tiene un peso del 30% en la calificación global de la práctica.

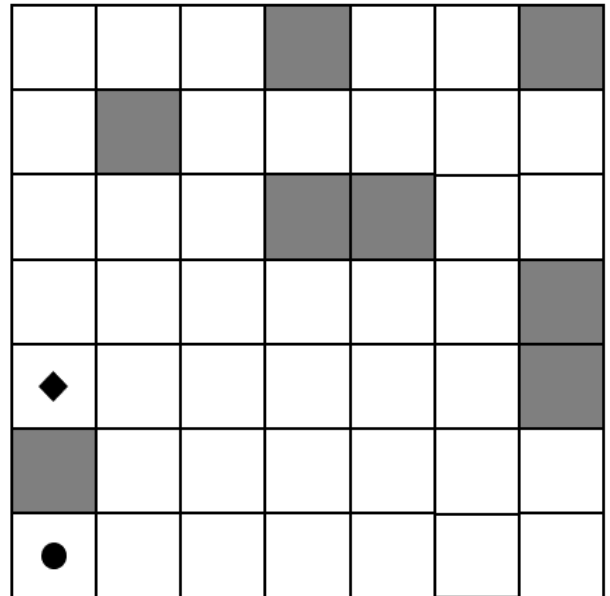
4. Sustentación

Ambos miembros del equipo de trabajo deben demostrar un dominio completo del desarrollo de la práctica. El profesor o el monitor verificará la funcionalidad del diseño y su ajuste a las restricciones impuestas, y planteará preguntas para evaluar los conocimientos adquiridos. La sustentación tiene un peso del 70% en la calificación global de la práctica.

5. Problemas

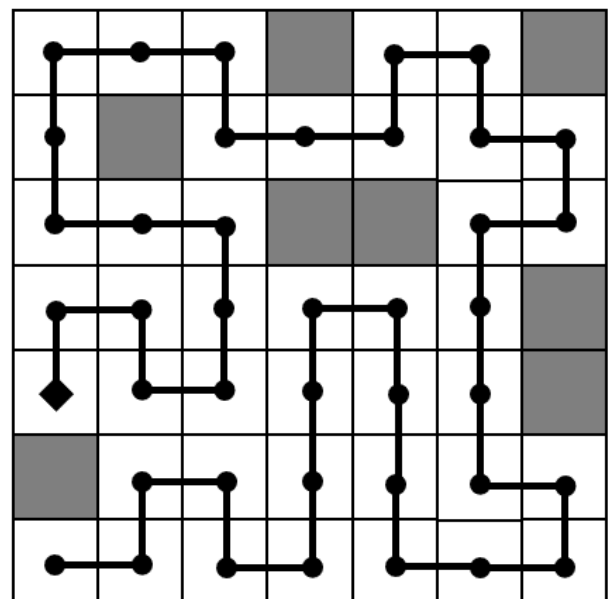
Considere el siguiente ejemplo para comprender el problema a resolver. En la Figura 3, a la izquierda, se muestra la cuadrícula de 7×7 celdas en la que cada celda tiene un número asignado. Este número será usado para identificar tanto la posición actual como la siguiente del robot. En la misma Figura 3, a la derecha, se representa la misma cuadrícula de 7×7 celdas, solo que ahora en ella se destacan algunas celdas sombreadas, una celda con un círculo negro y otra con un diamante negro en sus centros. Las celdas sombreadas indican obstáculos que deben ser sorteados por el robot (posiciones que no pueden ser ocupadas

43	7	24	19	61	28	32
52	0	12	23	17	42	41
8	29	9	48	33	27	4
3	63	47	2	37	13	54
60	49	44	62	6	25	20
35	1	30	31	50	59	57
22	45	38	51	53	40	21



El problema consiste en definir una trayectoria para llevar al robot del punto de partida al punto de destino. La trayectoria debe visitar todas las celdas que no correspondan a obstáculos, sin pasar dos veces por la misma celda. Observe que para la cuadrícula presentada en la Figura 3, la trayectoria representada en la Figura 4 es una solución que satisface los requerimientos.

43	7	24	19	61	28	32
52	0	12	23	17	42	41
8	29	9	48	33	27	4
3	63	47	2	37	13	54
60	49	44	62	6	25	20
35	1	30	31	50	59	57
22	45	38	51	53	40	21



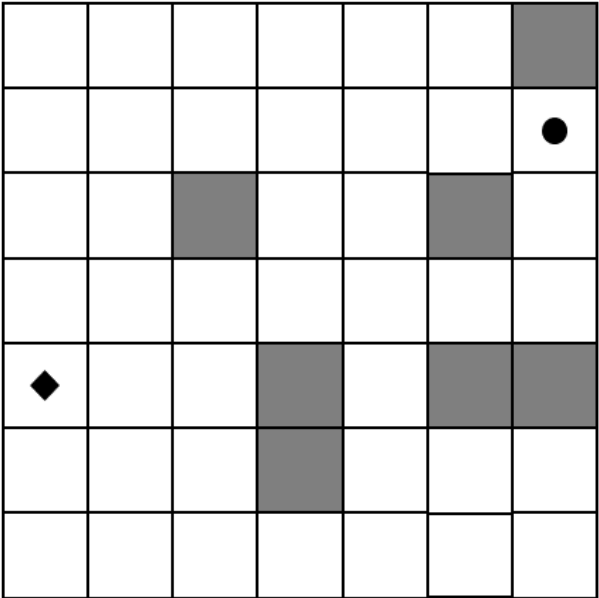
La trayectoria definida para el robot en la Figura 4 se puede describir en términos de las celdas que recorre: 22, 45, 1, 30, 38, 51, 31, 62, 2, 37, 6, ..., 47, 44, 49, 63, 3, 60. La trayectoria descrita de esta manera servirá construir la tabla de verdad del componente digital combinacional que guiará al robot en su recorrido. En el punto de partida, la primera acción ejercida por el robot será “Avanzar una celda” (*Go*), para ir al frente. De acuerdo con la solución planteada en la Figura 4, cuando el robot está ubicado en el punto de partida, su posición actual es “22”, su posición siguiente será “45”, y la acción a ejecutar será “Avanzar una celda” (*Go*). Cada uno de los problemas presentados más adelante tendrá una asignación de códigos para las cuatro acciones que puede realizar el robot.

Para conocer el número del problema que le corresponde desarrollar a cada equipo de trabajo, sume los dos últimos dígitos de la cédula de los miembros del equipo y calcule el módulo 10 del resultado.

Problema #0

Acción	Código
<i>Right + Go</i>	01
<i>Left + Go</i>	00
<i>Go</i>	10
<i>Stop</i>	11

43	7	24	19	61	28	32
52	0	12	23	17	42	41
8	29	9	48	33	27	4
3	63	47	2	37	13	54
60	49	44	62	6	25	20
35	1	30	31	50	59	57
22	45	38	51	53	40	21



Problema #1

Acción	Código
<i>Right + Go</i>	10
<i>Left + Go</i>	00
<i>Go</i>	11
<i>Stop</i>	01

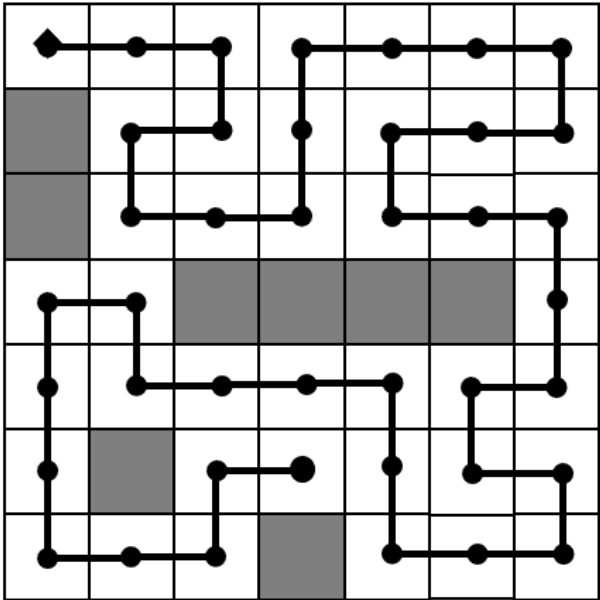
15	17	16	39	21	14	38
4	33	25	13	57	52	61
31	18	62	22	43	30	46
7	36	24	1	55	2	37
40	20	44	50	41	35	6
11	9	42	8	0	3	26
5	19	10	58	56	47	29

■			◆			
					■	
		■	■	■		
■						
		■	■			●

Problema #2

Acción	Código
<i>Right + Go</i>	00
<i>Left + Go</i>	10
<i>Go</i>	01
<i>Stop</i>	11

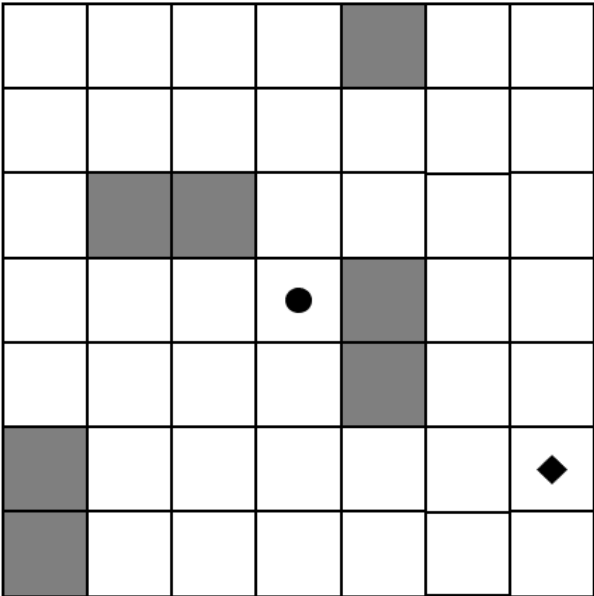
11	0	9	55	52	23	25
63	38	7	34	41	36	16
2	20	37	24	19	61	8
3	42	30	62	53	46	29
33	28	4	49	17	40	6
32	51	60	18	22	43	31
12	21	56	5	27	54	13



Problema #3

Acción	Código
<i>Right + Go</i>	11
<i>Left + Go</i>	10
<i>Go</i>	01
<i>Stop</i>	00

41	13	48	23	52	8	20
30	43	33	9	34	19	35
45	6	3	31	29	51	2
40	14	11	53	1	62	28
57	26	10	25	4	18	42
56	59	32	7	16	38	50
24	54	0	27	22	37	61



Problema #4

Acción	Código
<i>Right + Go</i>	10
<i>Left + Go</i>	11
<i>Go</i>	00
<i>Stop</i>	01

20	32	4	9	14	38	35
25	22	17	5	21	28	50
33	6	63	40	42	61	41
44	53	43	2	13	59	56
3	36	15	57	39	23	0
58	47	34	16	12	24	7
55	19	29	30	11	51	37

Problema #5

Acción	Código
<i>Right + Go</i>	11
<i>Left + Go</i>	01
<i>Go</i>	10
<i>Stop</i>	00

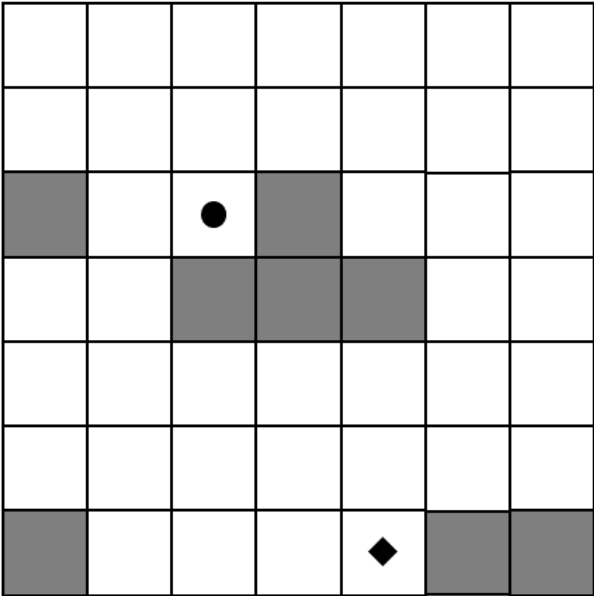
2	54	26	9	51	0	60
10	8	55	28	19	52	62
42	46	45	21	49	24	41
30	37	33	1	5	44	56
31	15	14	57	11	7	35
58	50	13	12	48	47	3
63	20	16	6	34	18	4

	●				◆	

Problema #6

Acción	Código
<i>Right + Go</i>	01
<i>Left + Go</i>	00
<i>Go</i>	10
<i>Stop</i>	11

57	40	6	20	43	19	47
56	33	42	12	50	39	54
51	32	27	5	53	14	18
48	13	55	52	7	45	22
38	1	9	24	41	36	49
0	58	2	30	60	61	63
25	59	26	4	3	16	62



Problema #7

Acción	Código
<i>Right + Go</i>	10
<i>Left + Go</i>	01
<i>Go</i>	11
<i>Stop</i>	00

20	63	44	32	60	57	47
28	38	11	35	7	18	22
58	41	12	33	19	15	53
59	51	5	24	61	39	4
23	1	46	16	55	34	2
40	6	62	13	42	49	56
10	52	29	37	17	8	21

						●
◆						

Problema #8

Acción	Código
<i>Right + Go</i>	00
<i>Left + Go</i>	11
<i>Go</i>	01
<i>Stop</i>	10

59	7	34	9	35	40	2
54	39	19	14	4	15	31
36	5	32	20	8	22	38
13	6	52	55	1	62	48
21	46	24	33	58	61	57
45	3	56	51	10	11	23
0	12	27	37	29	41	18

			◆			
						●

Problema #9

Acción	Código
<i>Right + Go</i>	10
<i>Left + Go</i>	11
<i>Go</i>	01
<i>Stop</i>	00

60	55	13	15	27	31	14
20	37	17	61	3	9	28
1	5	54	0	10	2	51
11	7	26	12	49	59	18
43	56	48	36	44	58	32
63	53	25	33	39	30	34
45	35	62	50	38	57	16
