

Laboratorio 2: Arquitectura y Organización de Computadores

Profesor: Viktor Tapia

Ayudante de cátedra: Mauricio Cortés y Muryel Constanzo

Ayudante de Tarea: Vicente Alvear y Luciano Yevenes

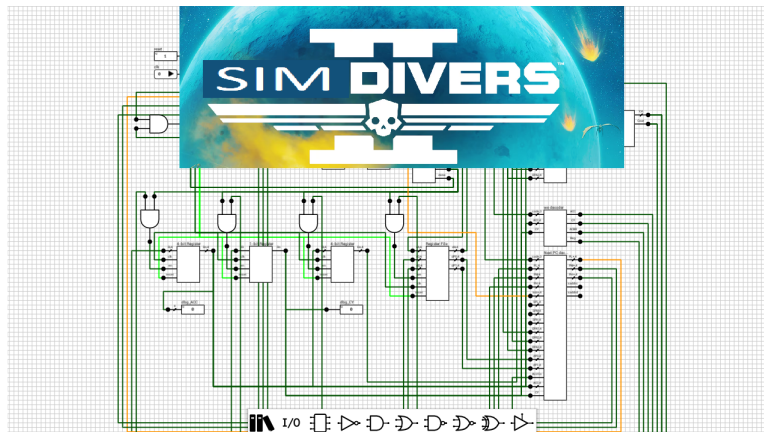
21 de Abril 2024

1 Reglas Generales

Para la siguiente tarea se debe utilizar la plataforma Logisim¹ para diseñar un circuito que cumpla con los requerimientos de la sección 2. Se exigirá que el formato de los circuitos se presente de la forma más limpia y ordenada posible. Deberá incluir un README con la identificación de los estudiantes que desarrollaron la tarea, además de cualquier supuesto utilizado.

2 Programa a implementar

2.1 SIMDIVERS 2



Saludos, ciudadano! Has sido involuntariamente voluntariado a servir a la Super Tierra en su distribución de democracia supervisada al resto de las estrellas. El ministerio de turbo arquitectura y circuitos lógicos de Inviktus Tapia lo ha encomendado a pautear la ruta hacia el próximo objetivo de invasión, *Malevelon Creek*, un planeta enemigo que debe ser arrasado para instalar un centro recreativo para hurones.

Sin embargo el camino hasta este planeta es peligroso y consta de varias rutas, por lo que se le ha pedido modelar las rutas posibles haciendo uso de la plataforma *logisim* para poder analizar cual es el mejor curso de acción.

¹Disponible en <https://sourceforge.net/projects/circuit/>

A continuación están los elementos del circuito a modelar. Cada planeta va a representar un estado, y el estado siguiente debe ser el que se acerque en dirección a *Malevelon Creek*. La entrada corresponde al planeta desde donde se quiere recorrer y posteriormente el estado se debe actualizar automáticamente usando lógica secuencial para llegar al sistema de destino.

2.1.1 El mapa

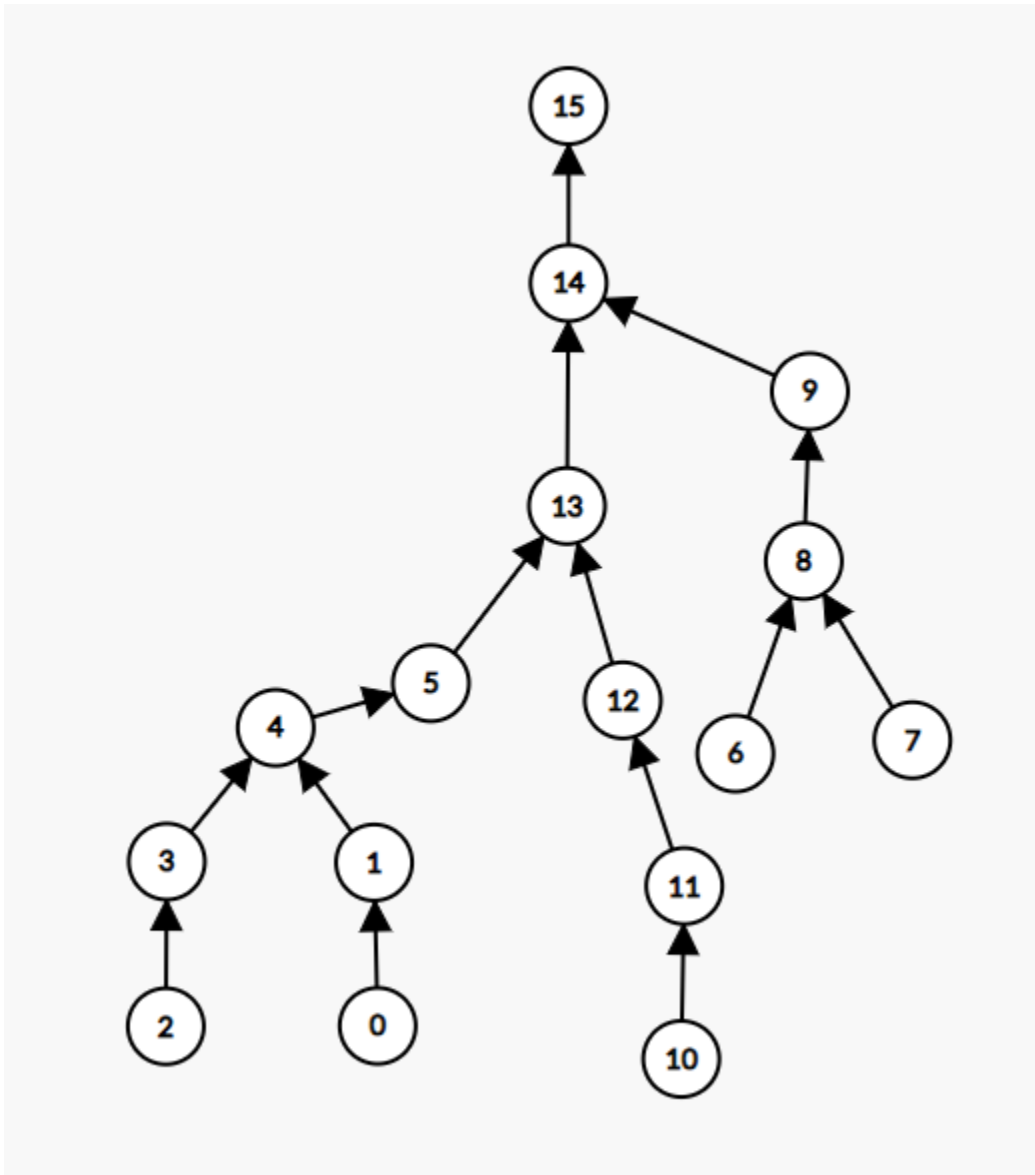


Figure 1: Mapa de ofensiva

Cada planeta tiene una dificultad asociada del 1 al 9, que se debe mostrar en un display de 7 bits. El nombre y la dificultad de cada nodo a continuación.

2.1.2 Los planetas

Numero	Nombre	Dificultad
1	Caladan	2
2	Arrakis	4
3	Terminus	1
4	Trantor	5
5	Etheria	5
6	Sanghelios	7
7	Ásgard	4
8	Namek	4
9	Gensokyo	6
10	Rend	3
11	Experimentation	8
12	Cadia	7
13	Lothric	9
14	Petricor V	7
15	Planeta Gol	8
16	Malevelon Creek	9

Entonces, su misión es generar un circuito que reciba una coordenada inicial que corresponda a uno de estos planetas, que pueda recorrer su ruta hasta *Malevelon Creek* actualizando su estado, y mostrando la dificultad en el display de salida de cada uno.

2.2 Formato Entrada

La lógica combinacional de salida consiste en 9 símbolos del 1 al 9 correspondientes al 1 al 9 en el display de 7 bits. **PRO TIP:** Se puede ahorrar trabajo buscando los valores de los bits de salida ya que el display de 7 bits está en el apunte.

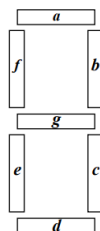


Figure 2: *display* de 7 bits.

Respecto a la lógica secuencial, esta debe encargarse de recibir 4 bits para representar un estado inicial y luego se debe seguir la trayectoria explicitada en el mapa para actualizar el estado siguiente hasta llegar al destino final.

Adjunto un salto de ejemplo:

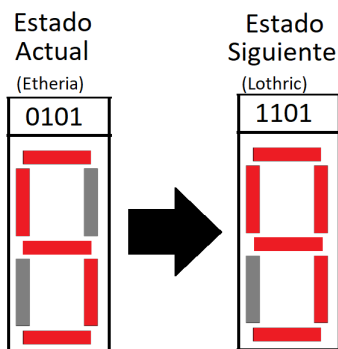


Figure 3: Cambio de estado desde Etheria a Lothric. Los 4 bits representan el estado y el display lo que debe mostrar la salida.

2.3 Requerimientos

El programa deberá poder:

- Recorrer el grafo descrito en 2.1.1 desde cualquiera de los estados de forma de que llegue al estado final.
- Mostrar en el display de salida la dificultad asociada al estado actual.
- Se deben elaborar las tablas de verdad y mapas de karnaugh para cada bit de salida y estos presentarse en un documento pdf llamado **informe.pdf**
- **SE DEBE USAR LA SALIDA DISPLAY DE 7 SEGMENTOS PROVEÍDA POR LOGISIM PARA LA SALIDA**

2.4 Formato de los circuitos

El circuito debe dividirse en varios subcircuitos, creados usando el botón en el menú de la izquierda de Logisim. El subcircuito principal debe llamarse main, y el resto de los subcircuitos deben tener nombres descriptivos. Toda lógica debe hacerse en subcircuitos separados para mantener el orden, y el subcircuito principal solo debe contener esos subcircuitos, similar a como se muestra en la figura 6.

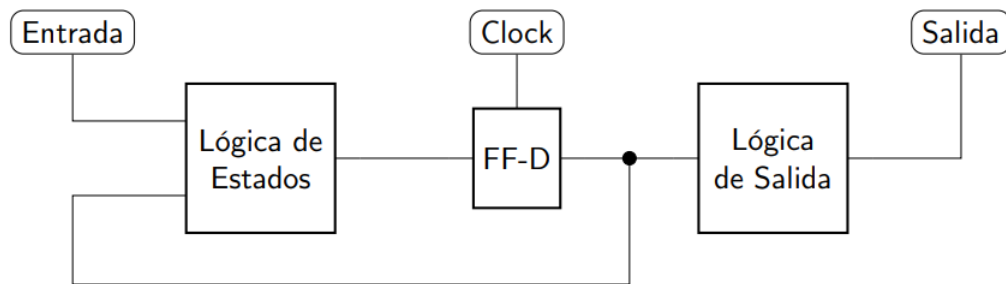


Figure 4: Esquema para *main*

3 README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los algoritmos y desarrollo realizado.
- Supuestos utilizados

4 Consideraciones

- Se deberá trabajar de a pares. Se deberá entregar en Aula a mas tardar el día 5 de Mayo de 2024 a las 23:59 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe realizarse en Logisim. Se recomienda que se familiarice rápidamente con la plataforma, y ante cualquier duda consulte con sus compañeros o directamente con los ayudantes lo antes posible. El único responsable si no acude a alguien para resolver sus dudas a tiempo es usted.
- Puede utilizar una cantidad arbitraria de subcircuitos, siempre y cuando cada uno tenga un nombre descriptivo e incluya como mínimo el subcircuito main.
- La entrega considera dos archivos, mapa.circ y informe.pdf, junto con el README. Los archivos deberán ser comprimidos y enviados juntos en un archivo .zip de nombre LAB2_ROL1_ROL2.
- El circuito puede o bien detenerse en el último nodo , o volver a un nodo inicial existente (como 0). La implementación en este sentido es libre.
- Si no se entrega README, o si su programa no funciona, la nota es 0 hasta la corrección.
- Una vez entregadas las notas de la tarea existirá un plazo de 5 días para apelar. Transcurrido este plazo las notas no podrán ser modificadas.