



IIC2343 – Arquitectura de Computadores (II/2018)

Tarea 4

Fecha de entrega: lunes 10 de septiembre de 2018 a las 11:59 AM

Parte única de programación

En esta tarea construiremos sobre las anteriores, extendiendo ahora el análisis a la velocidad del *clock* que alimenta a los circuitos. En particular, deben escribir un programa que, dada la descripción de una **máquina secuencial**, calcule la velocidad máxima del *clock* que todavía permita el funcionamiento correcto de la máquina. Para poder calcular esto, la descripción del circuito incluirá, para cada componente, el tiempo de retardo en que incurre una señal al pasar por él.

El programa debe ser escrito en Python 3.5 ó 3.6, debe tener como nombre `max_clock_calculator.py` y debe tomar como argumento por **línea de comandos** la ruta a un archivo JSON que describe la máquina secuencial. La salida del programa debe ser una impresión en pantalla, donde se indique la frecuencia calculada.

Formato

A continuación, se muestra un ejemplo del archivo de entrada JSON que podría recibir su programa:

```
1 {
2   "sequential": [
3     {"from": "input1", "to": "or1"},
4     {"from": "input2", "to": "or2"},
5     {"from": "or1", "to": "not1"},
6     {"from": "not1", "to": "or1"},
7     {"from": "not1", "to": "xor1"},
8     {"from": "xor1", "to": "or2"},
9     {"from": "or2", "to": "xor1"},
10    {"from": "xor1", "to": "out1"}
11  ],
12  "delays": [
13    {"gate": "or1", "delay": 4.1},
14    {"gate": "not1", "delay": 0.1},
15    {"gate": "xor1", "delay": 10.0},
16    {"gate": "or2", "delay": 4.5}
17  ]
18 }
```

Este es correspondiente al circuito de la siguiente figura:

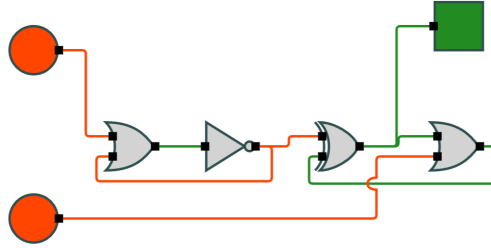


Figura 1: Circuito equivalente a la descripción anterior, donde los círculos representan `input1`, `input2` y el cuadrado `out1`.

Note que si bien este ejemplo no incluye elementos de memoria, **su programa debe ser capaz de soportarlos**. En case de haberlos, solo de incluirán *flip-flops* del tipo D.

El formato de *output* se espera que sea como sigue:

```
C:\User\IIC2343>python max_clock_calculator.py example.json
Maxima frecuencia permitida: 0.123
```

Note que esta frecuencia **no corresponde** a la respuesta del ejemplo.

Convención

La convención de entradas y salidas será “`input{i}`” y “`out{j}`” respectivamente, siendo *i* y *j* sus numeraciones para diferenciar las señales entre sí. Por otra parte, la convención de las compuertas es simplemente el uso de sus nombres con su respectiva numeración: “`and{k}`”, “`or{l}`”, “`xor{m}`”, “`not{n}`”. Finalmente, los *flip-flops* (que son exclusivamente de tipo D) siguen la nomenclatura “`ffd{s}`”, siendo *s* la numeración correspondiente. **Solo los *flip-flops* poseen el atributo “gate” dentro de su hash**, tanto para indicar un ingreso de señal a este como de salida.

A considerar

Además de lo anterior, deben considerar lo siguiente:

- Su circuito **puede** contener *loops*, tal como se evidencia en el ejemplo anterior.
- Dos *flip-flops* **pueden** estar conectados en serie, dado que no supone una complicación para los requisitos de esta tarea.
- Las compuertas a utilizar en el circuito son **solo** las descritas en la sección de **convención**.
- La descripción de los componentes **puede** estar desordenada, *i.e.* la lista que describe las componentes y conexiones del circuito podría no seguir un orden establecido. Por ejemplo, no necesariamente se definirán conexiones de *input* al principio y de *output* al final de la lista.
- El retardo se considera como una unidad adimensional, por lo que debe considerarlo como “unidades de tiempo”.
- Compuertas del mismo tipo **no necesariamente** comparten el mismo retardo, por ello en el atributo “`delays`” del JSON se especifica uno para cada compuerta.
- Los cables **no cuentan** con retardo.

Entrega y evaluación

La tarea se debe realizar de **manera individual** y la entrega se realizará a través de GitHub. El formato de entrega será el *script* en Python correspondiente, de nombre `max_clock_calculator.py` como se estipuló anteriormente. El repositorio, además, debe contar con un `README.md` escrito en *Markdown* que identifique sus datos y consideraciones que el corrector deba tomar en cuenta, tales como la versión en Python utilizada, sistema operativo usado para realizar la tarea, etc.

Archivos que no ejecuten o que no cumplan este formato de entrega implicarán nota **1.0** en la tarea. En caso de atraso, se aplicará un descuento de **1.0** punto por cada 6 horas o fracción.

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros. Si un alumno (grupo) copia un trabajo, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir reprobación del curso y un procedimiento sumario. Por “copia” se entiende incluir en el trabajo presentado como propio partes hechas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.