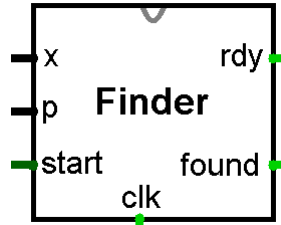


Parte a. (3 puntos) El circuito *Finder* de la figura entrega 1 por la salida *found* si $x \gg i \equiv p$ para algún i . De lo contrario *found* es 0. Los números x y p son entradas de 32 bits. El cuadro de texto muestra cómo se determinaría la respuesta en el lenguaje C. La búsqueda de la respuesta comienza cuando se detecta que *start* es 1 justo en el momento en que la entrada *clk* pasa de 1 a 0. En ese instante la salida *rdy* debe ir a 0 y permanecer en 0 mientras se busca la respuesta. Si *rdy* es 1 cuando el reloj pasa de 1 a 0, entonces la respuesta al problema está saliendo por *found*. Después de eso, *found* y *rdy* deben permanecer constantes hasta que nuevamente se detecte que *start* es 1 justo en el momento en que la entrada *clk* pasa de 1 a 0.



```
int finder(unsigned x,
           unsigned p) {
    while (x!=p) {
        if (x==0)
            return 0;
        x = x>>1;
    }
    return 1;
}
```

Implemente el módulo *finder* del circuito *finder.circ* incluido en los archivos adjuntos de esta tarea. Para entender cómo resolver esta parte y cómo verificar que funciona correctamente, vea [este video](#) en donde explico la solución de un problema similar de un control de arquitectura de computadores. La solución del problema que sale en el video está en el circuito *max1s.circ* de los archivos adjuntos. También le será de mucha utilidad ver los videos de [esta cátedra](#), [esta otra cátedra](#) y la [clase auxiliar](#), todos sobre circuitos. Pruebe que su solución funciona correctamente seleccionando el módulo *test* y simulando el circuito con control-r y control-k. Solo obtendrá el puntaje de esta parte si se enciende la luz verde.

Como resultado de esta pregunta Ud. debe entregar el circuito *finder.circ* en donde completó la implementación del módulo *finder*. Puede regular la velocidad de la simulación en *Simular* → *Seleccionar Frecuencia del reloj*.

Ayuda: El circuito *finder.circ* adjunto ya incluye las componentes que se necesitan para resolver el problema, pero puede no usarlas y usar otras componentes si lo estima necesario. Use el registro RX para almacenar la variable x de la solución en C y el registro RP para almacenar p . Almacene la entrada p en RP cuando *start* es 1. Para ello

use la entrada *enable* de RP. Para la entrada de RX use un multiplexor para seleccionar entre $RX \gg 1$ o la entrada x . Almacene la salida del multiplexor en RX solo cuando *start* es 1 o cuando *found* es 0.

Parte b. (1,5 puntos) La figura muestra un extracto del contenido de un *cache* de 4 KB (2^{12} bytes) de 1 grado de asociatividad con 256 líneas de 16 bytes. Por ejemplo en la línea 0a del *cache* (en hexadecimal) se almacena la línea de memoria que tiene como etiqueta 30a (es decir, la línea que va de la dirección 30a0 a la dirección 30af).

línea cache	etiqueta	contenido
0a	30a	
35	b35	
c1	1c1	

Un programa accede a las siguientes direcciones de memoria (en hexadecimal): b354, 30a5, 4c18, 30ac, 4c10, f0a0, b350. Indique qué accesos a la memoria son aciertos en el *cache*, cuales son desaciertos y rehaga la figura mostrando el contenido final del *cache*.

Parte c. (1,5 puntos) La tabla de la derecha muestra las instrucciones Risc-V ejecutadas por un programa. Haga un diagrama que muestre el ciclo en que se ejecuta cada etapa de las instrucciones, considerando una arquitectura (i) en pipeline con etapas *fetch*, *decode* y *execute*, y (ii) superescalar, con 2 pipelines con las mismas etapas de (i). Suponga que el salto ocurre y se predice exitosamente. Base su diagrama en los ejemplos que aparecen en [esta cátedra](#).

```
A and a5, s4, t0
B addi a4, t1, 4
C sub a4, t2, a5
D ori a4, t3, 4
E bgt a5, t4, L
F
G
...
M sub a2, s0, a0
N ori a1, a1, 4
```

Instrucciones

Baje *t6.zip* de U-cursos y descomprímalo. Contiene el circuito *finder.circ*, que Ud. debe modificar, y el circuito *max1s.circ* con la solución del ejemplo del [video](#).

Entrega

Entregue por medio de U-cursos un archivo .zip con el circuito *finder.circ*, modificado con su solución de la parte a, y las soluciones de las partes b y c en el formato de su elección (por ejemplo foto legible de su solución en papel). La parte a es binaria, se otorga 0 o todo el puntaje, pero en las partes b y c se otorga puntaje de acuerdo a lo logrado. Se descontará medio punto por día de atraso (excluyendo sábados, domingos, festivos o recesos).