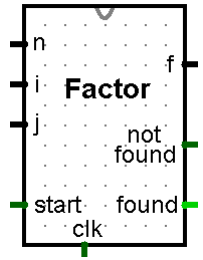
CC3301 Programación de Software de Sistemas – Tarea 6 – Primavera 2023 – Profesor: Luis Mateu

Parte a. (3 puntos) El circuito *Factor* debe buscar un factor impar de un número entero *n* en el intervalo [*i*, *j*] usando el siguiente algoritmo:

```
typedef unsigned int uint;
uint factor(int n, int i, int j) {
  for (uint k= i; k<=j; k+=1) {
    if (n % k == 0) { // calcula el resto de n/k
      return k; // k es un factor de n
  }
  return 0; // no se encontró ningún factor
}
```



La búsqueda 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 las salidas *found* y *not fount* deben ir a 0 y permanecer en 0 mientras se realiza el cálculo. Si se encuentra un factor se entrega un 1 en la salida *found* y el factor se entrega en la salida *f*. Si no hay un factor impar en [*i*,*j*] se entrega un 1 en la salida *not-found* y *f* es irrelevante. Las entradas *n*, *i* y *j* permanecerán constantes hasta que se solicite una nueva búsqueda llevando *start* a 1. Debe mantener las salidas *f, found* y *not fount* constantes hasta que *start* vuelva a 1.

Implemente el circuito *Factor* en el módulo *factor* del circuito *factor.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 estas 2 cátedras y la clase auxiliar, todos sobre circuitos. **Pruebe que su solución funciona** correctamente seleccionando el módulo *test-factor* 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 *factor.circ* en donde completó la implementación del módulo *factor*. Puede regular la velocidad de la simulación en *Simular → Seleccionar Frecuencia del reloj*.

***Ayuda*:** El circuito *factor.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 *RegK* para representar la variable *k* de la solución en C. Observe que en los comentarios del módulo *factor* está la solución del problema en texto. Solo debe traducir ese texto a un circuito de logisim.

Parte b. (1,5 puntos) La figura muestra un extracto del estado actual de un *caché* de 4 KB (2^{12} bytes) de 1 grado de asociatividad con 256 líneas de 16 bytes. Por ejemplo en la línea 2a del caché (en hexadecimal) se almacena la línea de memoria que tiene como etiqueta 92a (es decir, la línea que va de la dirección 92a0 a la dirección 92af).

línea cache	*etiqueta*	*contenido*
e2	4e2	
45	c45	
2a	92a	

Un programa accede a las siguientes direcciones de memoria (en hexadecimal): c450, 92ac, 5e24, 5e20, 92a8, 2450, 4e20, 92a4. Indique qué accesos a la memoria son aciertos en el caché, cuáles son desaciertos y rehaga la figura mostrando el estado final del cache. Por ejemplo el acceso c450 es un acierto.

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 en F ocurre (y no hay ningún tipo de predicción de saltos). Base su diagrama en los ejemplos que aparecen en estas cátedras.

```
A sub  a3,s5,t2
B add  a5,t2,s4
C andi a3,a3,255
D addi a3,a3,1
E ori  a5,a5,15
F bgt  a3,s1,R
G add  ...
H sub  ...
I xor  ...
J andi ...
...
R sub a3,a3,a5
S ori a3,a3,255
```

Instrucciones

Baje *t6.zip* de U-cursos y descomprímalo. Contiene el circuito *factor.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 *factor.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).