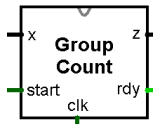


Parte a. (3 puntos) El circuito *Group Count* de la derecha calcula y entrega en su salida *z* la cantidad de grupos de unos en su entrada *x*. Por ejemplo la cantidad de grupos de unos en 0b110100111 es 3, la de 0b10111011101011 es 5. El programa del cuadro de más abajo realiza el cálculo.



```
int group_count(
    unsigned int x) {
    int c = 0;
    while(x!=0){
        if((x&0b11)==0b1)
            c++;
        x=x>>1;
    }
    return c;
}
```

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 la salida *rdy* debe ir a 0 y permanecer en 0 mientras se realiza el cálculo durante varios ciclos del reloj. Una vez terminado el cálculo, el resultado debe aparecer en la salida *z* y *rdy* debe ir a 1. Debe mantener las salidas *z* y *rdy* constantes hasta que *start* vuelva a 1 solicitándose un nuevo cálculo de grupos.

Implemente el circuito *Group Count* en el módulo *group* del circuito *group.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-group* 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 *group.circ* en donde completó la implementación del módulo *group*. No puede modificar los demás módulos. Puede regular la velocidad de la simulación en *Simular* → *Seleccionar Frecuencia del reloj*.

Ayuda: El circuito *group.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 *XReg* para representar la variable *x* de la solución en C y el registro *CReg* para la variable *c*. Observe que en los comentarios del módulo *groups* 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 inicial 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 3b del caché (en hexadecimal) se almacena la línea de memoria que tiene como etiqueta f3b (es decir, la línea que va de la dirección f3b0 a la dirección f3bf).

línea cache	etiqueta	contenido
a4	7a4	
3b	f3b	
95	c95	

Un programa accede a las siguientes direcciones de memoria (en hexadecimal): 7a48, f3b4, d950, f3b0, c950, 5a40, c958. Indique qué accesos a la memoria son aciertos en el caché, cuáles son desaciertos y rehaga la figura mostrando el estado final del cache. Observe que el caché no está vacío inicialmente y por lo tanto el acceso 7a48 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 E ocurre (y no hay ningún tipo de predicción de saltos). Base su diagrama en los ejemplos que aparecen en [estas cátedras](#).

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 *groups.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 *groups.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).