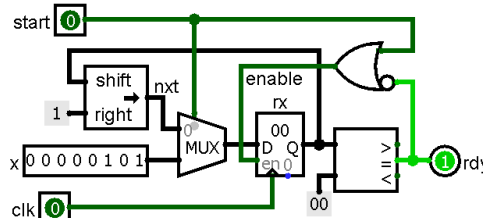


CC3301 Programación de Software de Sistemas – Control 3
Semestre Primavera 2024 – Profs. Mateu/Ibarra/Urrea

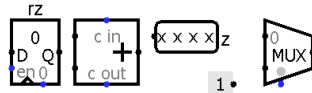
Pregunta 1

I. Considere el circuito de la figura. Complete la tabla de la derecha con los valores que toma cada señal hasta el ciclo 6 del reloj. El valor de la entrada x se mantiene constante en 0b101 (5).



ciclo	start	rx (bin)	nxt (bin)	enable	rdy
1	0	0	0	0	1
2	1	0			1
3	0	101			0
4	0				
5	0				
6	0				

II. Modifique el circuito agregando la salida z . En z Ud. debe entregar el número de ciclos que tarda rx en volver a 0. Por ejemplo, si la entrada x es 0b101 (5), cuando rdy se ponga en 1, la salida z debe ser 0b11 (3). Necesitará agregar y conectar estas componentes:



Pregunta 2 (30%)

A) La figura de la derecha muestra un extracto del estado inicial de un *caché* de 4 KB (2^{12} bytes) de 1 grado de asociativad con 256 líneas de 16 bytes. Por ejemplo en la línea 35 del *caché* (en hexadecimal) se almacena la línea de memoria que tiene como etiqueta **435** (es decir, la línea que va de la dirección **4350** a la dirección **435f**). Un programa accede a las siguientes direcciones de memoria (en hexadecimal):

da90, 27b8, 6a90, 27bc, e350, da98, 6a94.

línea cache	etiqueta	contenido
35	435	
7b	27b	
a9	da9	

Indique qué accesos a la memoria son aciertos en el *caché*, cuáles son desaciertos y rehaga la figura mostrando el estado final del *caché*. Observe que el *caché* no está vacío inicialmente y por lo tanto el acceso **da90** es un acierto.

B) 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 B no ocurre y el salto en E sí ocurre (y no hay ningún tipo de predicción de saltos).

A	addi	s1,t0,15
B	bgt	t0,a3,P
C	ori	a4,t2,1
D	add	s2,s1,a4
E	beq	s2,zero,P
F	...	
G	...	
H	...	
I	...	
J	...	
...		
P	addi	a4,a3,1
Q	sub	t7,a4,t5

Pregunta 3 (30%)

En la función f de abajo, $p(x)$, $g(x)$ y $h(x)$ toman mucho tiempo en calcularse.

<code>int p(double x);</code>	<code>double f(double x) {</code>
<code>double g(double x);</code>	<code> if (p(x))</code>
<code>double h(double x);</code>	<code> return g(x);</code>
	<code> else</code>
	<code> return h(x);</code>
	<code>}</code>

Se sabe que $p(x)$ es verdadero en el 90% de los casos. **Modifique** f de manera que se invoque *fork* para calcular $p(x)$ en paralelo con $g(x)$, especulando que $p(x)$ será verdadero. Si $p(x)$ es verdadero, entregue el resultado de $g(x)$, y si no tome un tiempo adicional para calcular y entregar $h(x)$. No puede crear más de un proceso adicional.