CC41C Introducción al Hardware

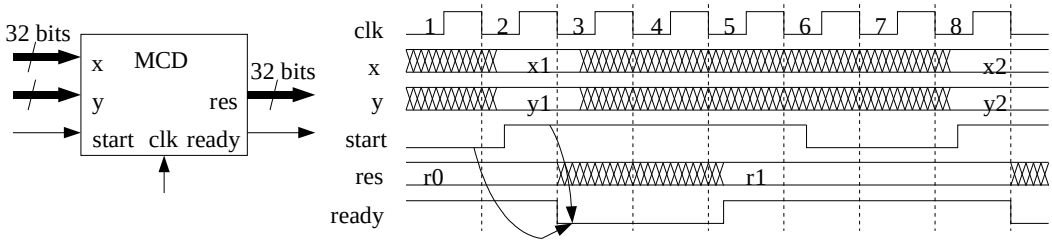
Control 2 - Otoño 2005 Profesor: Luis Mateu

Pregunta 1

El siguiente algoritmo permite calcular el máximo común divisor (MCD) entre dos números x e y:

```
while (x!=y) {
  if (x>y) x= x-y;
  else y= y-x;
}
el MCD es x
```

Use diseño modular para construir un circuito que calcule el MCD. El diagrama de tiempo de la figura muestra el funcionamiento de este circuito. Normalmente la salida *ready* se encuentra en 1 indicando que el circuito está listo para realizar un nuevo cálculo y que la salida *res* contiene el último MCD calculado (r0). El circuito debe comenzar a trabajar cuando detecta una transición de 0 a 1 en la entrada *start* en dos pulsos de bajada del reloj sucesivos (lo que ocurre en los ciclos 2 y 8 en el diagrama). Las cifras *x* e *y* son memorizadas en el pulso de bajada del reloj (x1 e y1). Entonces el circuito trabaja por un número variable de ciclos. Mientras tanto, la salida *ready* debe estar en 0. Cuando el cálculo termina, MCD coloca la salida *ready* en 1 y arroja el resultado por la salida *res* (r1=mcd(x1, y1), en el ciclo 5) y permanece constante hasta que se deba realizar un nuevo cálculo.



Observación: Utilice sumadores, registros (a base de flip/flops data), multiplexores, etc. Use además un circuito secuencial para controlar el funcionamiento de las distintas partes. Para el circuito secuencial especifique solo su comportamiento mediante un diagrama de estados y explique sus entradas y salidas, *no lo implemente*.

Pregunta 2

El bus de datos de un microcontrolador es de 8 bits y su bus de direcciones de 16 bits. Esto permite direccionar hasta un máximo de 64 KB de memoria. Después del encendido, el microcontrolador parte ejecutando la instrucción que se encuentre en la dirección hexadecimal `0xff00` (casi al final de los 64 KB), por lo que se requiere que en esa dirección haya una memoria ROM que se encargue de cargar e inicializar el sistema operativo. Esto es conflictivo porque dado que 64 KB es poca memoria se desea por otra parte que todo corresponda a memoria RAM.

Para resolver el conflicto se usa una técnica que consiste en que justo después del encendido el microcontrolador ve una memoria RAM ubicada en [0, 56 KB[y una memoria ROM de 8 KB en [56, 64 KB[. Un programa de la ROM se encarga de cargar e inicializar el sistema operativo en la memoria RAM, *sin escribir* en la memoria ROM. La interfaz en hardware de la memoria ROM y RAM hace que cuando se escribe la primera vez en alguna dirección asignada a la memoria ROM (i.e. Rango [56, 64 KB[ ), ésta se reemplaza por 8 KB de memoria RAM adicionales (no inicializada), y así la primera escritura se realiza en memoria RAM. De ahí en adelante el microcontrolador ve que los 64 KB son memoria RAM. No hay forma de acceder nuevamente al contenido de la memoria ROM sin apagar el sistema.

Se le pide a Ud. que *diseñe e implemente* este sistema a partir de las componentes de la figura. Ud. dispone además de un Latch, que se garantiza que después del encedido su valor es 0, y el resto de las componentes modulares usuales (ands, ors, nots, decodificadores, multiplexores, reloj, etc.).