

Práctico 9 RTL.

Ejercicio 1. (H&P 84 Prob. 10.2) En la figura se muestra la descripción RTL de un módulo sencillo.

a) Siga los pasos en este módulo, suponiendo $X = 0101$ en el momento del paso 1, $X = 1001$ en el paso 3, y $y = 0$ siempre. Hacerlo hasta el paso 7.

b) Repetir a), suponiendo $X = 1001$ en el paso 1, $X = 0101$ en el paso 3, y $y = 1$ siempre.

```

MODULE : EJ_1
MEMORY : A[4]; B[4]; C[4].
INPUTS : X[4]; y.
OUTPUTS : Z[4].
1 A ← X.
2 B ← A\.
3 A ← X.
4 B ← B + A; → ( A[3] ) / ( 7 ).
5 A * y ← A\.
6 A ← A . B.
7 C ← A + B; Z = A; → ( 1 ).
ENDSEQUENCE
CONTROL RESET ( 1 ).
END.
    
```

Ejercicio 2. (H&P 84. Prob. 10.3)

Dado el sistema descrito por la secuencia RTL de la figura, dibujar diagramas lógicos detallados de la sección de datos y de la sección de control. Diseñar un circuito que reciba como entrada la señal de RESET (que se supondrá asíncrona respecto al reloj del sistema, y que se asegura que dura más de un período de reloj) y genera las señales a conectar a las entradas asíncronas set y reset de los F-F de la sección de control, de modo de asegurar que el primer nivel de control CSL dure exactamente 1 período de reloj, luego que la señal de RESET vuelve al nivel 1 inactivo.

```

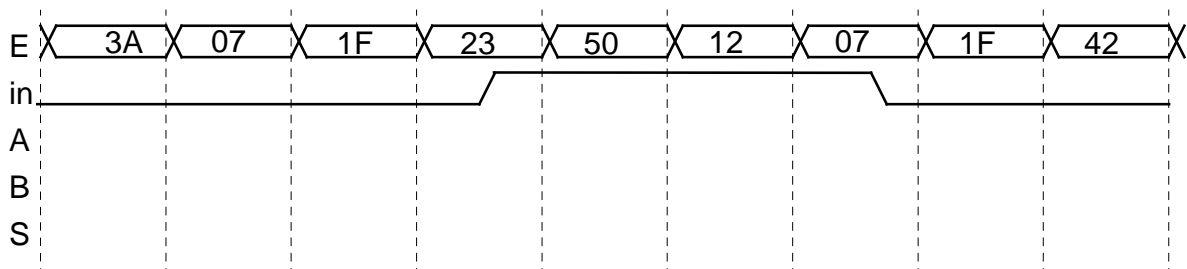
MODULE : Ej_2
MEMORY : R[3].
INPUTS : X[3]; a; b.
OUTPUTS : Z[3]; ready.
1 R ← X;
  → ((a\.b\), a, (a\.b)) / ( 1, 2, 4 )
2 R ← R[0], R[2:1];
  → ( b\ ) / ( 4 ).
3 R ← R[0], R[2:1].
4 ready = 1; Z = R; → ( 1 ).
ENDSEQUENCE
CONTROL RESET ( 1 ).
END.
    
```

Ejercicio 3. (ex. Marzo 1996)

Dada la siguiente secuencia RTL construir el bloque de datos, el bloque de control y completar el siguiente diagrama de tiempos, suponiendo que en el primer período se está en el paso 1:

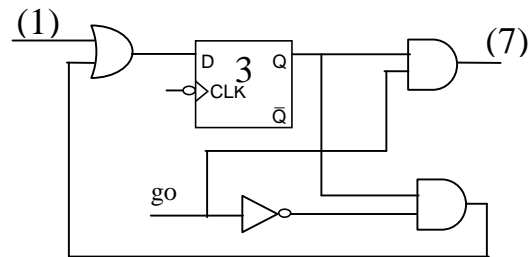
```

MODULE Ej_3
MEMORY:      A[8], B[8]
INPUTS:      E[8], in
OUTPUTS:     S[8]
1. A ← E
2. A ← A+1, S=A, B ← E
3. A ← A+1, S = A+B
4. A*in ← A+1, S=A, → 1
ENDSEQUENCE
CONTROLRESET ( 1 )
END
    
```



Ejercicio 4. (H&P 84. Prob. 10.5)

En cierta secuencia RTL, el paso 3 es una bifurcación solamente, sin transferencia. La figura muestra la sección de control para este paso. Determinar el contenido del paso 3.



Ejercicio 5. (H&P 87. Prob. 4.9)

Construya un diagrama lógico detallado de las secciones de datos y control especificadas por la siguiente descripción RTL. Dibujar diagramas de tiempo para los niveles y pulsos de control suponiendo que la entrada **x** vale 0 todo el tiempo y que **b** inicialmente vale 0. Discutir dos casos diferentes según todos los F-F sean sensibles al flanco de subida o al flanco de bajada del reloj.

```

MODULE Ej_5
MEMORY:      a, b
INPUTS:      x
OUTPUTS:     z
1. a ← x + b
2. b ← x;
   → (a, a\)/(1, 3).
3. z = 1; b ← x xor b
   → ( 1 )
ENDSEQUENCE
CONTROLRESET (1)
END
    
```

Ejercicio 6. (H&P 84. Prob. 10.11)

Un sistema digital tiene líneas de entrada de datos de 8 bits, **X[8]**, y una línea de entrada **ready**. Tiene dos líneas de salida, **same** y **ask**. Cada vez que el sistema pone un 1 en **ask** durante un período de reloj, un sistema externo pondrá nuevos datos en las líneas **X**, siendo indicada la presencia de los datos por **ready** en valor 1 durante un período. Los datos en **X** quedarán estables hasta que se emita una nueva señal **ask**. Cada vez que los datos de entrada son exactamente los mismos que los datos recibidos en cualquiera de las dos veces anteriores, la línea **same** deberá ir a 1 durante cuatro períodos antes de emitir una nueva señal **ask**. Si los datos no son los mismos, la señal **same** deberá permanecer en 0 y se emitirá una nueva señal **ask** en el siguiente período.

Se pide:

- a) Escribir una descripción RTL.
- b) Bloque de datos y de control completos.

Ejercicio 7. (ex. Agosto 97)

En la secuencia RTL del recuadro, la entrada **X** está sincronizada con el reloj del sistema y la salida del bloque combinatorio **comp** vale 1 si **X** es igual a 0Fh y 0 en caso contrario.

Se pide:

- a) Dibujar el circuito completo.
- b) Hallar la frecuencia máxima de funcionamiento. Justificarlo con un diagrama de tiempos.

```

MODULE:      Ej_7
INPUT:      X[8]
OUTPUT:     s
MEMORY:
1. →(comp, !comp)/(2, 1)
2. s = 1; → (1)
ENDSEQUENCE
CONTROLRESET (1)
END
    
```

Datos:

- $0 < t_x < t_x^{max}$: retardo desde el flanco de reloj a **X** estable
- $0 < t_d < t_d^{max}$: retardo de las compuertas
- $0 < t_c < t_c^{max}$: retardo del bloque comp
- $0 < t_p < t_p^{max}$: propagación de los FF
- t_{su} : setup requerido para los FF

Ejercicio 8. (H&P 84. Prob. 10.12)

Un sistema digital tiene dos líneas de entrada, **control** y **data**, 8 líneas de salida **Z[8]**, y una línea de salida **ready**. Los datos representando palabras de 8 bits llegarán serialmente (un bit por vez) sobre la línea **data**. Será convertido a formato paralelo para sacarlo por las líneas **Z**. El primer bit en llegar de los datos serie puede ser el más significativo o el menos significativo. En cualquier caso la salida paralelo debe ser de la forma normal, con el más significativo a la izquierda.

Se provee sincronización por la interacción entre la entrada **control** y la salida **ready**. Cuando el sistema está pronto para recibir una nueva palabra, subirá la línea **ready** y empezará a monitorear **control**. Dos unos consecutivos (igual a uno durante dos períodos de reloj) en **control** indicará que el siguiente bit en control indicará la dirección de los datos serie, 0 para el MSB primero, 1 para el LSB primero. Los 8 bits de datos aparecerán a continuación en **data** durante los siguientes 8 períodos de reloj. **Ready** deberá bajar cuando los dos unos consecutivos en **control** se detecten y deberá volver a 1 cuando los 8 bits hayan sido recibidos. La palabra paralelo deberá presentarse en las líneas **Z** siempre que **ready** esté alta.

Realizar la descripción RTL del sistema descrito y dibujar los bloques de datos y control.

Ejercicio 9. (ex. Marzo 97 modificado)

Se quiere realizar el control de semáforos de tiempo variable (figura 1) para ser colocado en la intersección de una avenida con una calle. El semáforo tiene 4 salidas posibles que son: **RA** (rojo para la avenida-amarillo para la calle), **VR** (verde-rojo), **AR** (amarillo-rojo) y **RV** (rojo-verde). Los tiempos de cada salida serán múltiplos de **T** ($T \gg T_{CLK}$), donde **T** corresponde a la duración de la luz amarilla (salidas **RA** y **AR**).

Las salidas deben activarse (por nivel alto) según la secuencia **RA, VR, AR, RV, RA,**

El semáforo dispone además de 2 entradas, **VRT[4]** y **RVT[4]**, provenientes de sensores y contadores de autos, en las cuales se lee el tiempo que deberán estar activas las salidas **VR** y **RV**, respectivamente. El circuito realizará una lectura de las entradas solamente mientras se está dando la salida **RA**.

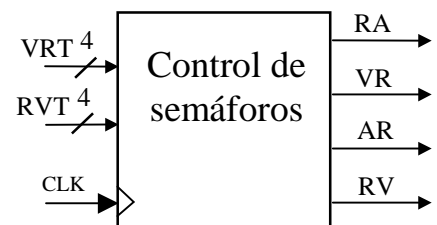


Figura 1

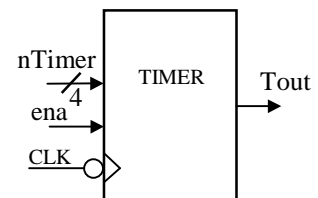


Figura 2

Para medir tiempo se dispone de un "TIMER" (figura 2). Inicialmente o en estado de reposo, **Tout = 0**. El TIMER leerá el valor en su entrada **nTimer** en el flanco de bajada de reloj en que **ena = 1**. A partir de este flanco, **Tout** pasará a 1 por un tiempo $N \times T$. Luego de este tiempo, el TIMER volverá a quedar en estado de reposo hasta recibir un nuevo **ena = 1**. Si **ena = 1** antes de finalizar la cuenta, se reiniciará el TIMER con el valor en **nTimer**.

Para realizar el control de semáforos, diseñar un circuito RTL que utilice el TIMER.

Se pide:

- Diagrama de conexionado entre el circuito RTL a diseñar y el TIMER.
- Secuencia RTL, bloque de control y de datos del circuito a diseñar
- Dibujar el diagrama de tiempos durante los 2 ciclos de reloj alrededor de las transiciones de **RA** a **VR** y **AR** a **RV**, para **VRT = 3** y **RVT = 2**. Indicar el paso, valor de las entradas, salidas y registros.