

- Mínimo de aprobación: **un problema y un ejercicio completos**
- Cada hoja debe tener Nombre y Cl.
- Utilice solo un lado de las hojas
- Deben estar numeradas y la primer hoja debe decir el total de hojas
- Incluya un solo problema por hoja
- Sea prolijo

Problema1

Se quiere diseñar un circuito modo reloj que controle el arranque del motor de un disco duro, que deberá encenderse cuando se active la entrada On.

El motor debe arrancarse en forma suave por lo tanto el encendido se realiza en dos etapas. Para ello, el sistema que comanda el motor dispone de dos entradas para indicar la tensión de alimentación del motor: Tmed y Tmax. La entrada Tmed conecta el motor a la mitad de la tensión de alimentación, con lo que el motor empezará a girar suavemente hasta llegar a su velocidad de rotación de 4000 rpm. Esto se indica mediante la señal Vel, que permanece activa siempre que el motor gire por encima de esa velocidad.

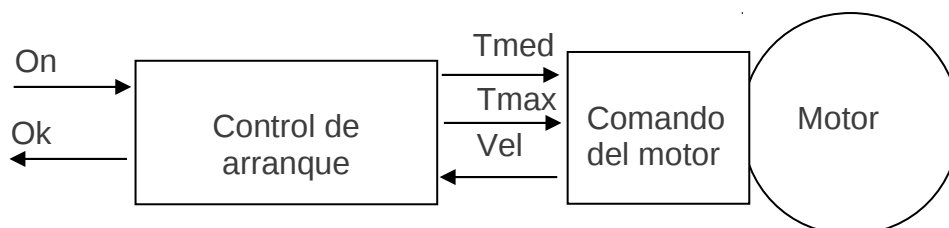
Luego que se active Vel, el circuito de control de arranque debe activar la señal Tmax, manteniendo Tmed activa, para que el motor acelere hasta la velocidad de régimen. Una vez activada Tmax se deberá esperar al menos un período de reloj completo para darle tiempo al motor a que llegue a su velocidad máxima y a partir de ahí se deberá activar la salida Ok para indicar que el motor está disponible para su uso. Para mantener la velocidad de régimen, Tmax y Tmed deben permanecer activas.

Si después de activar Ok el motor llega a trancarse y su velocidad baja de 4000 rpm se desactivará la señal Vel, en este caso se deberá desactivar inmediatamente la señal Ok y parar el motor. Para volver al funcionamiento normal se deberá esperar que se baje la entrada On.

En todo momento cuando se desactive la entrada On se deberá apagar el motor. Se podrá suponer que una vez que se apaga éste permanecerá apagado varios períodos de reloj antes de volver a encenderse.

Todas las señales son activas en nivel alto.

Diseñar completamente el circuito modo reloj que controle el arranque del motor.



Problema 2

Se desea diseñar un taxímetro que aproveche la información de un GPS para determinar el monto a pagar por viaje.

El GPS tiene una interfaz con las siguientes entradas y salidas (ver figura):

Distancia[14..0]: Salida que indica la distancia neta recorrida, en metros, desde la última vez que se reseteo el GPS. El dato es válido si newData =1.

newData: Salida que indica con un nivel alto que hay un dato válido en Distancia. La señal está **sincronizada** y permanece en 1 por 200ns.

ResetGPS: Entrada activa por nivel alto, que lleva a cero el contador de Distancia.

El GPS está diseñado para dar una "Distancia" cada 0.1s, pudiendo ser ésta igual a la anterior.

El taxímetro a diseñar, además de la interfaz con el GPS, deberá contar con una salida **Ficha[6..0]** para mostrar las fichas totales contadas y 3 botones de entrada; **Start** para comenzar a contar las fichas, **Stop** para parar el contador de fichas y **Borrar** para volver a cero la cantidad de fichas a cobrar. Los 3 botones son activos con un pulso a 1 de entre 110 ns y 190 ns.

Los botones serán operados de acuerdo a la secuencia Start >> Stop >> Borrar >> Start Si no se respeta esta secuencia, se debe ignorar el botón oprimido.

El sistema tendrá dos condiciones de cobro, según el estado de movimiento del taxi. En el caso en que el taxi se encuentre detenido, se debe incrementar una ficha cada 5 segundos. Si por el contrario, está en movimiento, se debe incrementar una ficha cada vez que el vehículo recorre 250m, independientemente de que se hayan contabilizado fichas por tiempo en el recorrido.

Notas:

- Si el taxi permanece detenido menos de 5 segundos, se debe ignorar dicho lapso de tiempo.
- Todo viaje dura menos de 128 fichas y por lo tanto es más corto que 32Km.
- El contador de fichas deberá ir mostrando la cantidad de fichas, pero no es necesario que se actualice en forma inmediata.
- Se dispone de un reloj de 10MHz para el funcionamiento del sistema (taxímetro y GPS).

Diseñar completamente un circuito RTL, secuencia, bloque de datos y control, que implemente el taxímetro descrito anteriormente.

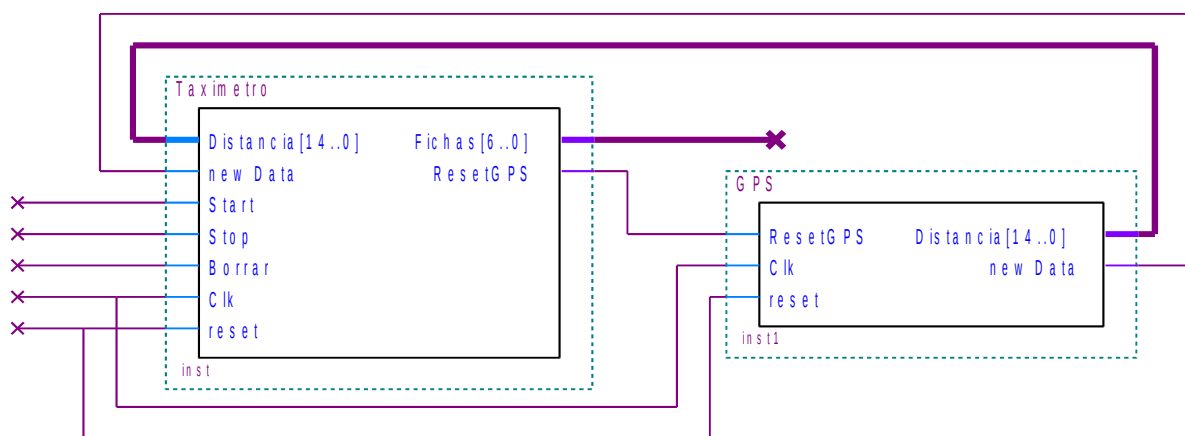


Figura 1: Esquema taxímetro

Ejercicio 1

Dados A=01001010 y B=01100100

- Realizar las operaciones A+B y A-B si A y B son números representados en complemento a dos indicando si hubo o no overflow, justificar.
- Tomando A como parte entera y B como parte fraccionaria de un número en punto fijo convertirlo a punto flotante indicando el error de conversión. Utilizar la representación en punto flotante: 1 bit de signo, 5 bits para el exponente y 10 bits para la mantisa: $N=(-1)^s * 2^{(e-15)} * 1,f$

Ejercicio 2

Dada la siguiente tabla correspondiente a un circuito secuencial modo nivel, se pide:

- asignar variables de estado de modo que no haya carreras usando la menor cantidad de variables de estado posibles
- asignar salidas evitando espurios

	00	01	11	10	00	01	11	10
q0	q1	<u>q0</u>	<u>q0</u>	q2		1	0	
q1	<u>q1</u>	q3	q2	<u>q1</u>	0			0
q2	q3	q0	<u>q2</u>	<u>q2</u>			0	1
q3	<u>q3</u>	<u>q3</u>	q0	q1	1	0		