

Exámen Febrero 1999

2.3.2. Letra del problema

Diseñar un circuito RTL que invierta el orden de transmisión de los bits en una comunicación serial. El circuito tendrá una entrada de datos serie in y una salida de datos out. La recepción y transmisión se hace con un bit de arranque en cero, 8 bits de datos y al menos un bit de parada en uno.

Se supondrá que la señal in cambia sincronizada con una señal de reloj que se utilizará como reloj del sistema. El circuito deberá tener una entrada de RESET.



Se deberá tener en cuenta que la transmisión de un byte y el siguiente pueden ser contiguas (separados por el bit de parada), con lo cual el sistema podría estar recibiendo un byte y al mismo tiempo transmitiendo el anteriormente recibido.



Nota: se admitirán soluciones en más de un bloque RTL, siempre que se especifiquen sus conexiones e interfaz.

Solución

Secuencia RTL del receptor

MODULE: RECEPTOR

INPUTS: in

MEMORY: DIN_FF[8], CONT_IN[3]

OUTPUTS: DATO[8], start

1. CONT_IN[] <- 0
-> (in, !in) / (1,2)
2. CONT_IN[] <- INC(CONT_IN[])
DIN_FF[7..0] <- DIN_FF[6..0], in
-> (Fin_R, !Fin_R) / (3,2)
3. start = 1
-> (1)

END SEQUENCE

CONTROL RESET (1)

DATO[] = DIN_FF[]

Fin_R = CONT_IN[2] and CONT_IN[1] and CONT_IN[0]

END

Secuencia RTL del transmisor

MODULE: TRANSMISOR

INPUTS: DATO[8], start

MEMORY: DOUT_FF[8], CONT_OUT[3]

OUTPUTS: out

1. out=1
CONT_OUT[] <- 0
DOUT_FF[]*start <- DATO[]
-> (start, !start) / (2,1)
2. out = 0
3. out = DOUT_FF[0]
CONT_OUT[] <- INC(CONT_OUT[])
DOUT_FF[][7..0] <- DOUT_FF[0], DOUT_FF[7..1]
-> (Fin_T, !Fin_T) / (1,3)

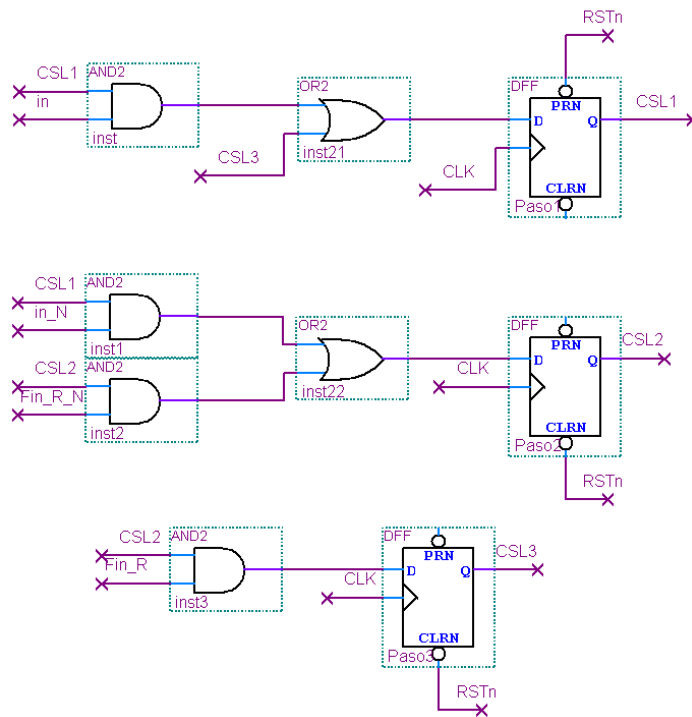
END SEQUENCE

CONTROL RESET (1)

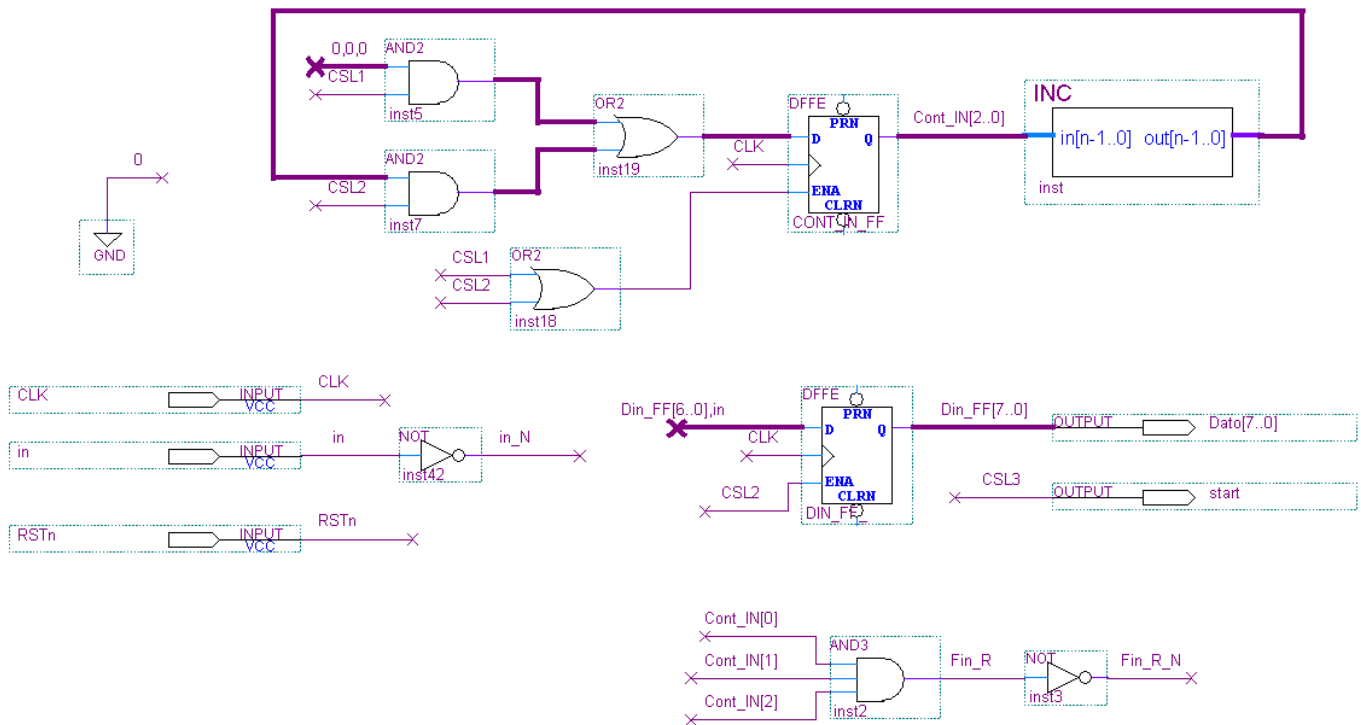
Fin_T = CONT_OUT[2] and CONT_OUT[1] and CONT_OUT[0]

END

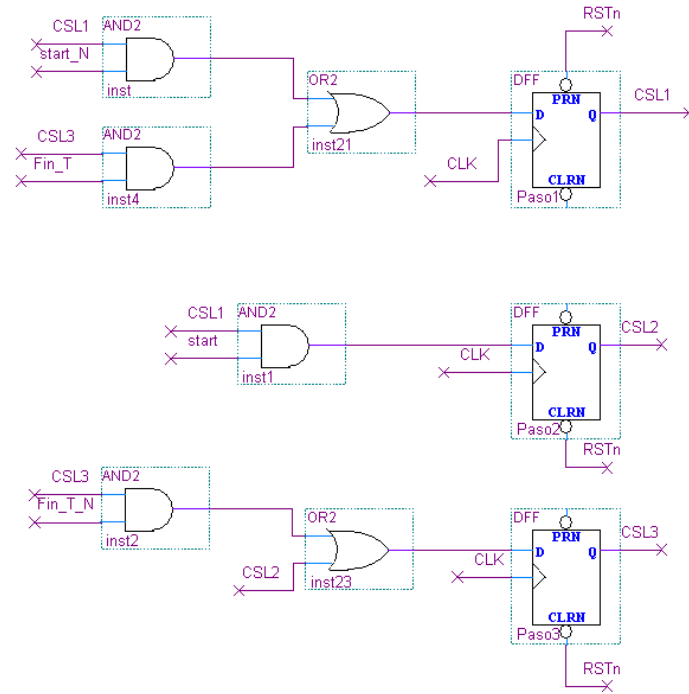
BLOQUE DE CONTROL RECEPTOR



BLOQUE DE DATOS RECEPTOR



BLOQUE DE CONTROL TRANSMISOR



BLOQUE DE DATOS TRANSMISOR

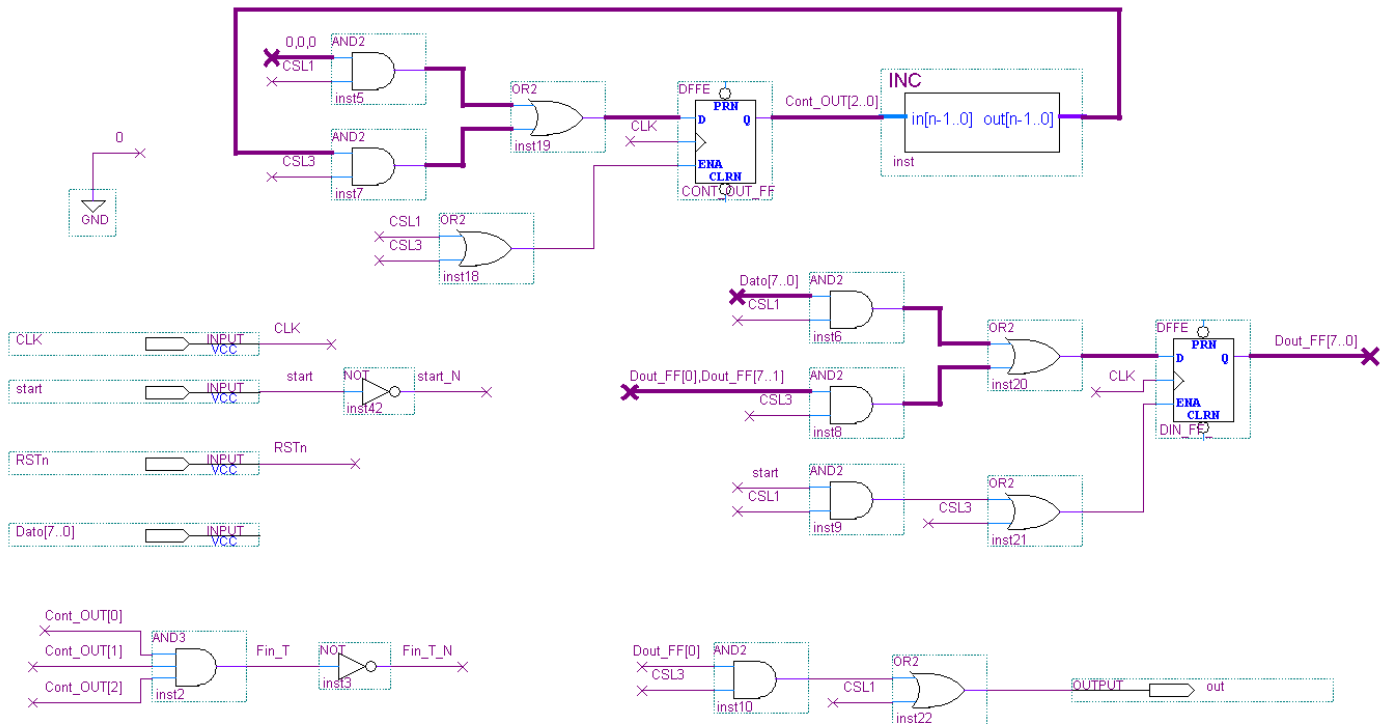


DIAGRAMA DE TIEMPO

