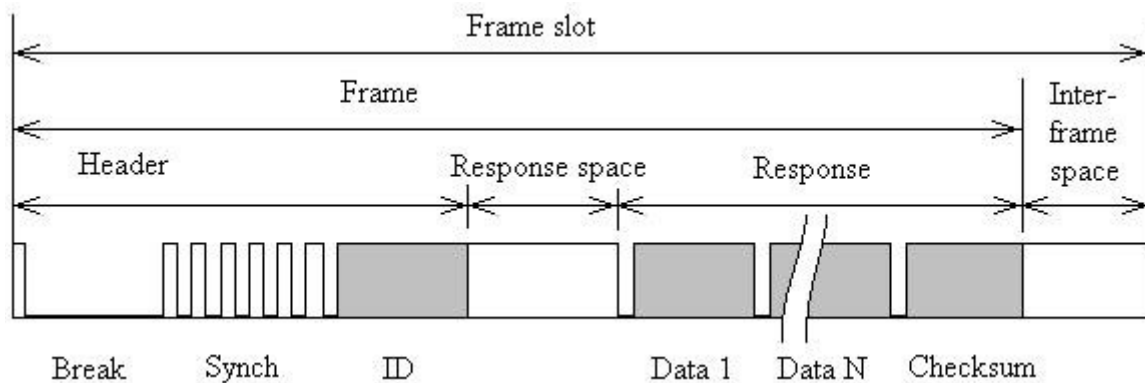


Actividad 4.1

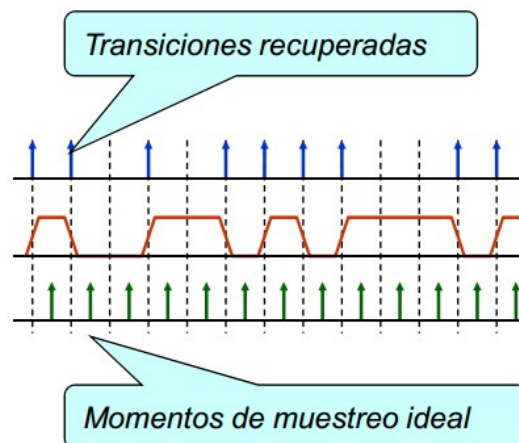
1) En el protocolo LIN el aprendizaje en los Slaves del Baud Rate se hace con el Header de la trama que envía el Master. El dato de sincronización es el hexa 0x55. En la siguiente imagen se puede observar el orden de los datos



En donde la trama Sync en realidad es



2) Para pensar este inciso se partió de la premisa donde el sistema a sincronizarse se asemeja al siguiente esquema



Para esto se planteó un esquema similar al realizado en la Actividad 3.2, solo que esta vez se debe tener en cuenta que Rel2_048 no siempre cuenta con flancos. Para esto se diseñó un seguimiento de fase que corrige con más peso cuando se obtienen flancos inducidos por la transición de los datos. La única diferencia notable fue que el divisor %4/%5 se pensó de manera más dinámica y a mano, ya que el módulo que simula el MC4044 no sigue un orden favorable de estados en caso de pérdida de flancos.

```

process(Clk72MHz)
begin
  if rising_edge(Clk72MHz) then
    sRel2_048 <= Rel2_048;
    sR2048X_Ant <= sR2048X;
  end if;
end process;

Rel2048edge <= (not sRel2_048) and Rel2_048;
R2048Xedge <= (not sR2048X_Ant) and sR2048X;

NUM4_5_Next <= NUM4_5 + PASO when Count_Slow > to_unsigned(to_integer(Count_Fast+Count_Slow)*(100-to_integer(DUTY))/100,Count_Slow'length) and NUM4_5 < MAX_DEN else
  NUM4_5 - PASO when Count_Fast > to_unsigned(to_integer(Count_Fast+Count_Slow)*to_integer(DUTY)/100,Count_Fast'length) and NUM4_5 > to_unsigned(0,NUM4_5'length) else
  NUM4_5;

process(Clk72MHz, Reset)
begin
  if Reset = '1' or Rel2048edge = '1' then
    Count_Slow <= to_unsigned(0,Count_Slow'length);
    Count_Fast <= to_unsigned(0,Count_Fast'length);
    NUM4_5 <= NUM4_5_Next;
    BeforeEdge <= '0';
  elsif rising_edge(Clk72MHz) then
    if R2048Xedge = '1' then
      BeforeEdge <= '1'; -- Con el Duty de esta señal se puede controlar el punto de muestreo de los datos de entrada
    end if; -- En este caso está en 50%, pero con la comparaci

    if BeforeEdge = '0' then
      Count_Fast <= Count_Fast + to_unsigned(1,Count_Fast'length);
    end if;

    if BeforeEdge = '1' then
      Count_Slow <= Count_Slow + to_unsigned(1,Count_Slow'length);
    end if;
  end if;
end process;

process(sR2048X)
begin
  if rising_edge(sR2048X) then
    Dat2_048_Syc <= Dat2_048;
  end if;
end process;

```

Donde la signal BeforeEdge es la señal de control, y el Duty de la misma (configurable) define el instante deseado de adquisición del dato de entrada.

3) Una propuesta sería que los '0's incorporados a las cadenas de '1's sean ceros válidos, es decir, ceros que de todas maneras iban a ser enviados en un futuro, solamente que se adelantan para cumplir el lugar de los bits de stuffing.

4) Una propuesta, la cual también vale para el inciso anterior, es la de generar un cambio de polaridad, es decir, que cada 6 '1's se cambie la polaridad de toda la transmisión, de esta manera si sigue habiendo '1's para enviar se van a enviar '0's. En este caso, si con la polaridad invertida se tienen que mandar muchos '0's, la política será que con 6 '0's consecutivos se vuelva a cambiar la polaridad volviendo a la condición inicial.