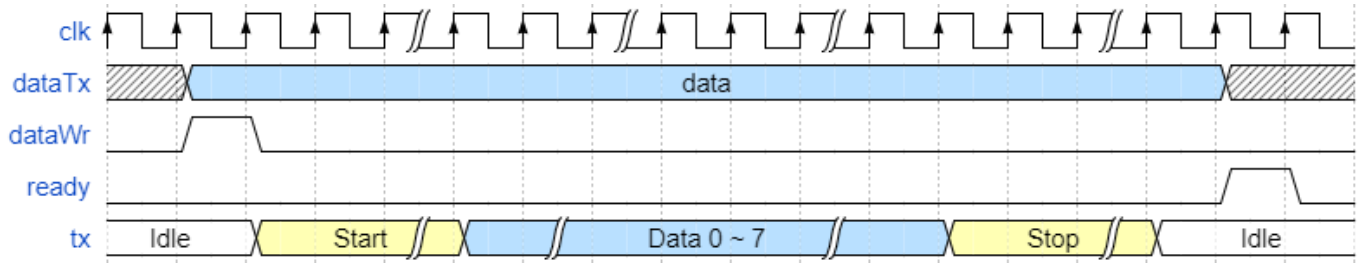
1. Implemente un transmisor de la UART (9600 8N1).

```
entity myUartTx
   Generic (baudRate : integer := 9600;
            sysClk     : integer := 50000000;
            dataSize   :  integer := 8);
   Port ( clk          : in  std_logic;
          rst          : in  std_logic;
          dataWr       : in  std_logic;
          dataTx       : in std_logic_vector (dataSize - 1 downto 0);
          ready        : out std_logic;
          tx           : out std_logic);
end myUartTx;
```

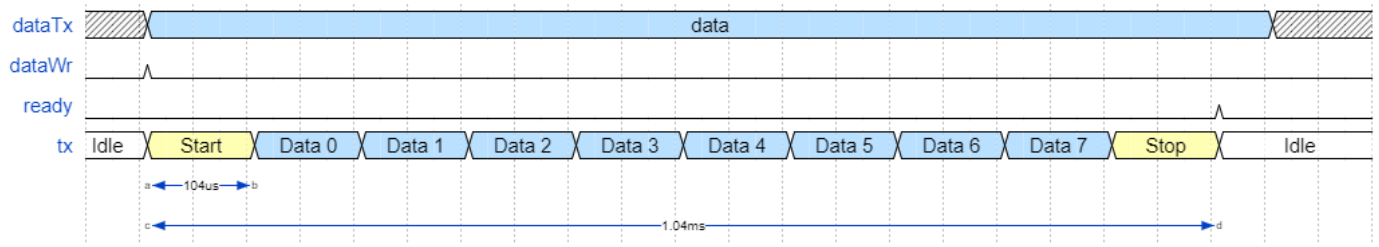
Donde:

- dataTx: Es el dato a Tx.
- dataWr: Es una señal que indica el inicio de la transmisión de los datos. Dura un pulso de clk
- ready: Indica cuando el tx esta nuevamente disponible para Tx. Dura un pulso de clk
- tx: La salida de transmisión.
- sysClk: Es la frecuencia de clock del sistema
    - Para el kit Atlys es de 100MHz
    - Para el kit SP3 es de 50MHz

Ejemplo de Tx de datos con el clock como referencia.



Ejemplo de Tx de datos para observar el tiempo de transmisión de un bit y el byte entero. Las señales dataTx y ready se ponen como referencia, tenga en cuenta al observar el diagrama que duran un ciclo de clock del sistema.



El pin tx toma el valor cero en el bit start, y uno en el bit stop y en el estado idle.
El tiempo por bit es de 104us mientras que el tiempo de tx total es de 1.04ms
Verifique el funcionamiento recibiendo con una simulación.

2. Utilizando myUartTx como componente, diseñe una máquina de estados que transmita el estado de los 4 interruptores del kits en en el nibble más significativo y en el nibble menos significativo transmita 0x5 cada vez que los interruptores cambien de estado. Verifique lo transmitido recibiendo los datos en una terminal de la PC.

```
entity myUartTxTest
   Port ( clk              : in  std_logic;
          rst              : in  std_logic;
          sw               : in std_logic_vector (3 downto 0)
          tx               : out std_logic);
end myUartTxTest;
```

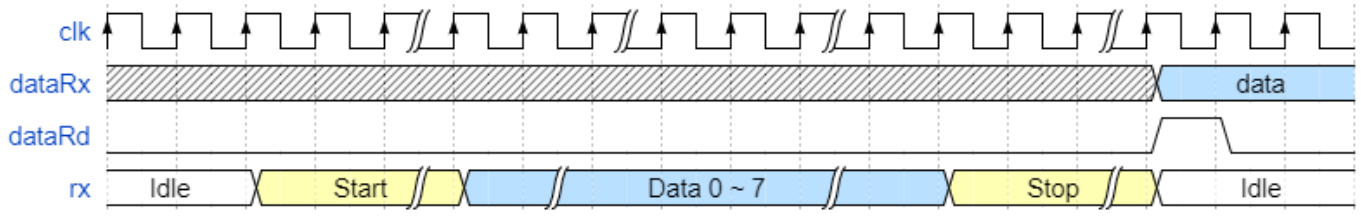
3. Implemente un receptor de la UART (9600 8N1). Verifique el funcionamiento con una simulación

```
entity myUartRx
   Generic (baudRate : integer := 9600;
            sysClk     : integer := 50000000;
            dataSize  :  integer := 8);
   Port ( clk          : in  std_logic;
          rst          : in  std_logic;
          dataRd       : out  std_logic;
          dataRx       : out std_logic_vector (dataSize - 1 downto 0);
          rx           : in std_logic);
end myUartRx;
```

Donde:

- dataRx: Es el dato recibido.
- dataRd: Indica cuando se recibio un dato.Dura un pulso de clk.
- rx:Entrada de recepción.

El siguiente gráfico muestra la recepción de datos con el clk de referencia.



4. Utilizando myUartRx como componente, verifique el funcionamiento colocando el dato recibido en los leds del kit

```
entity myUartRxTest
   Port ( clk          : in  std_logic;
          rst          : in  std_logic;
          led          : out std_logic_vector (7 downto 0);
          rx           : in std_logic);
end myUartRxTest;
```

5. Usando el módulo transmisor y receptor como componente arme una UART.

```
entity myUart
   Generic (baudRate : integer := 9600;
            sysClk     : integer := 50000000;
            dataSize  :  integer := 8);
   Port ( clk          : in  std_logic;
          rst          : in  std_logic;
          dataWr       : in  std_logic;
          dataTx       : in std_logic_vector (dataSize - 1 downto 0);
          ready        : out std_logic;
          tx           : out std_logic;
          dataRd       : out  std_logic;
          dataRx       : out std_logic_vector (dataSize - 1 downto 0);
          rx           : in std_logic);
end myUart;
```

6. Instancie la UART. Verifique el funcionamiento colocando en los leds el dato recibido y el estado de los 4 interruptores del kits en el nibble más significativo y en el nibble menos significativo transmita 0x5.

```
entity myUartTest
   Port ( clk                 : in  std_logic;
          rst                 : in  std_logic;
          sw                  : in std_logic_vector (3 downto 0);
          tx                  : out std_logic;
          led                 : out std_logic_vector (7 downto 0);
          rx                  : in std_logic);
end myUartTest;
```