



```

graph LR
    btn --> Buf[ ]
    Buf -- reset --> echoUnit
    mclk --> diviseurClk
    diviseurClk -- clk 155 kHz --> echoUnit
    diviseurClk -- clk 155 kHz --> UART
    echoUnit -- cs --> UART
    echoUnit -- rd --> UART
    echoUnit -- wr --> UART
    UART -- intR --> echoUnit
    UART -- intT --> echoUnit
    echoUnit -- d1 --> UART
    UART -- d2 --> echoUnit
    UART -- TxD --> TxD_out[TxD]
    RxD_in[RxD] --> UART
    
```

UART\_FPGA\_N4

1

## 2 Interfaces

### 2.1 Nexys4 (architecture à écrire)

```
entity UART_FPGA_N4 is
  port (
    mclk : in std_logic;
    btn  : in std_logic;
    rxd  : in std_logic;
    txd  : out std_logic);
end UART_FPGA_N4;
```

### 2.2 diviseurClk

```
entity diviseurClk is
  generic(facteur : natural)
  port (
    clk, reset : in std_logic;
    nclk       : out std_logic);
end diviseurClk;
```

### 2.3 echoUnit (fourni)

```
entity echoUnit is
  port (
    clk, reset : in std_logic;
    cs, rd, wr : out std_logic;
    IntR       : in std_logic;
    IntT       : in std_logic;
    addr       : out std_logic_vector(1 downto 0);
    data_in    : in std_logic_vector(7 downto 0);
    data_out   : out std_logic_vector(7 downto 0));
end echoUnit;
```