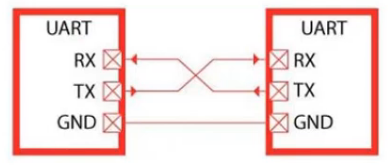
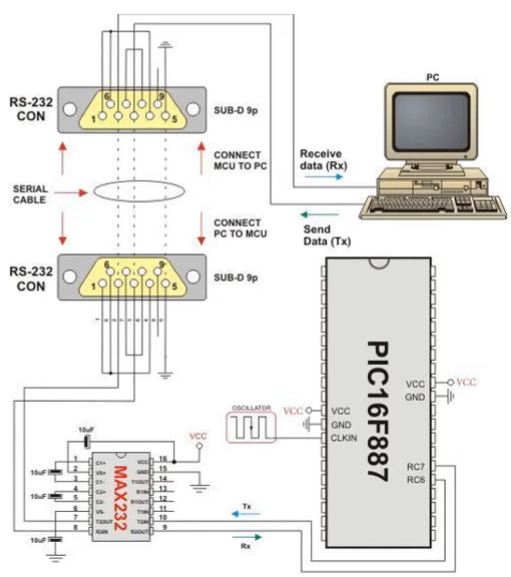
1. GIỚI THIỆU VỀ GIAO THỨC TRUYỀN THÔNG UART

Bộ truyền và nhận không đồng bộ UART để gửi dữ liệu qua một đường truyền nối tiếp



1. Giao thức truyền thông UART

Hai bên đều có TX và RX hai cặp đường này nối chéo với nhau và một đường đất nối chung nhau (GND). Chuẩn UART thường được sử dụng chung với chuẩn giao tiếp RS232, chuẩn giao tiếp xác định đặc tính điện, cơ, hàm và thủ tục để hai thiết bị có thể giao tiếp với nhau. Kết hợp này để truyền dữ liệu đi xa.



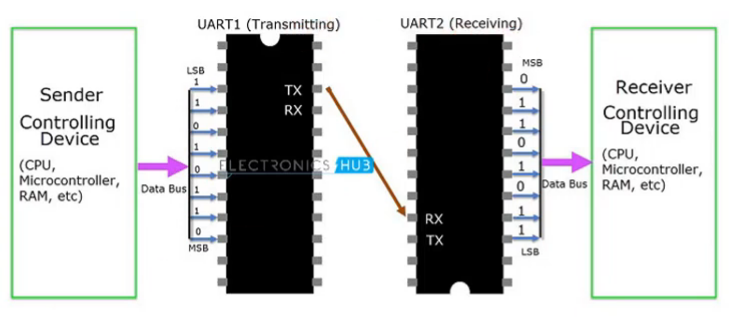
1. Kết hợp UART với RS232

UART sử dụng điện áp chuẩn TTL (mức cao ‘1’ ~ 5V; mức thấp ‘0’ ~ 0V).

Các máy tính ngày nay đã không còn sử dụng cổng COM RS232 vì chiếm nhiều không gian và hạn chế khả năng ứng dụng thay thế bằng cổng phổ biến hơn là cổng USB.

* 1. Khảo sát chi tiết khối UART

Bao gồm bộ phát (Transmitting) và bộ nhận (Receiving).

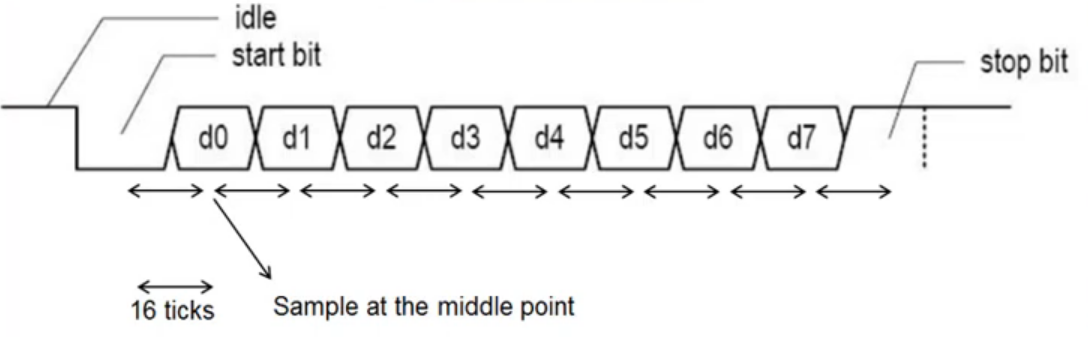


1. Mô tả cấu trúc UART

Bộ phát cần có một thanh ghi dịch đặc biệt để nạp dữ liệu song song và sau đó dịch từng bit ra với một tốc độ xác định ở chân (TX).

Bộ nhận cũng có một thanh ghi đặc biệt (vào nối tiếp - ra song song) đầu vào (RX) này được nối với đầu ra (TX) của bộ phát. Bộ nhận sẽ dịch từng bit dữ liệu vào và sau đó tổng hợp lại thành 1 gói dữ liệu giống như gói dữ liệu mà bộ phát truyền đến.

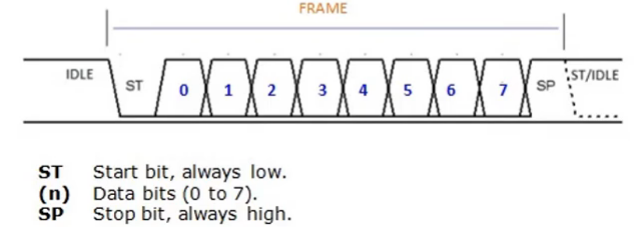
* 1. Khảo sát đường truyền dữ liệu của hệ thống phát và hệ thống nhận



1. Khung truyền của giao thức UART

Ở trạng thái rỗi (iDLE) không thực hiện quá trình truyền nhận thì đường truyền ở mức logic ‘1’. Việc truyền dữ liệu bắt đầu bằng bit START(mức ‘0’). Tiếp theo đó, là những bit dữ liệu. Số lượng bit dữ liệu có thể là 6, 7 hoặc 8. Sau khi truyền xong các bit dữ liệu nếu có sử dụng kiểm tra parity thì tiếp theo là bit PARITY. Cuối cùng kết thúc bằng bit STOP (mức ‘1’). Số lượng bit Stop có thể là 1, 1.5 hoặc 2. Kết thúc quá trình truyền dữ liệu.

Bit PARITY là bit kiểm tra phát hiện lỗi trong quá trình truyền nhận. Đối với kiểm tra lẻ thì bit PARITY bằng ‘0’ khi tổng số bit ‘1’ trong gói dữ liệu là lẻ. Đối với parity chẵn thì bit PARITY bằng ‘0’ khi tổng số bit ‘1’ là chẵn.



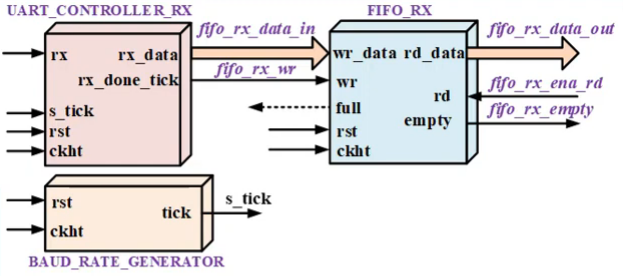
1. Trạng thái của quá trình nhận dữ liệu

Trong hệ thống, LSB của gói dữ liệu sẽ được truyền đầu tiên. Trong hình trình bày việc truyền dữ liệu 8bit dữ liệu, không có bit PARITY, 1bit STOP.

Trong hệ thống truyến dữ liệu UART do không có tín hiệu xung clock được truyền qua đường nối tiếp kết nối giữa hai hệ thống thì trước khi bắt đầu truyền, nhận bộ phát và bộ thu phải thiết lập thông số giống nhau: tốc độ baud, số bit dữ liệu, số bit stop và bit parity có sử dụng hay không. Thông thường, tốc độ baud là 2400, 4800, 9600, 19200, …

1. THIẾT KẾ BỘ NHẬN UART

Bộ nhận có sơ đồ khối 3 thành phần:

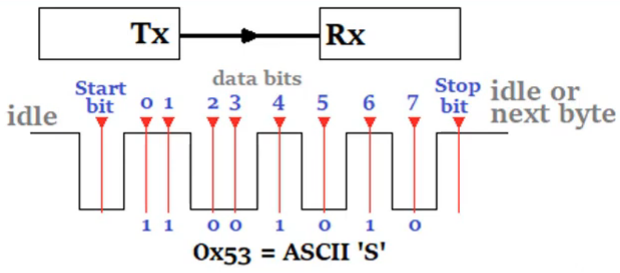


1. Sơ đồ khối nhận UART

* UART Controller Receiver (UCR): Mạch nhận dữ liệu từ chân RX thông qua lấy mẫu.
* Baud Rate Generator (BRG): Tạo ra xung lấy mẫu.
* Interface Circuit (IC): Mạch giao tiếp.
  1. Thủ Tục Lấy Mẫu

Do không sử dụng xung clock khi truyền, bộ nhận sẽ nhận lại dữ liệu bằng cách sử dụng các tham số đã được xác định ban đầu. Bộ nhận sẽ dùng mạch lấy mẫu để tìm điểm giữa của những bit dữ liệu và sau đó tiến hành lưu giữ liệu.

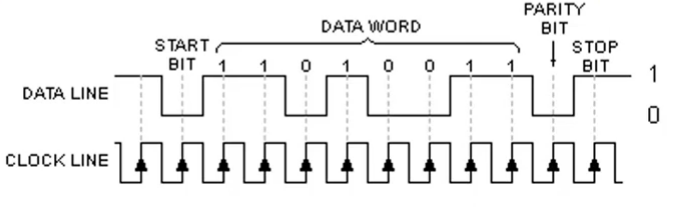
Hầu hết các mạch lấy mẫu sẽ có tốc độ bằng 16 lần tốc độ baud, nghĩa là mỗi bit nối tiếp sẽ được lấy mẫu 16 lần.



1. Vị trí lấy dữ liệu trên mỗi bit

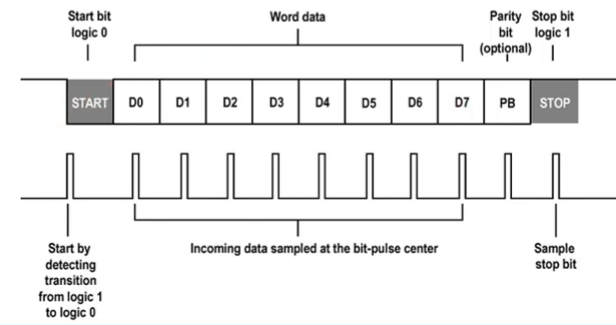
Giả sử ta quy định truyền N bit dữ liệu và M bit stop, khi đó mạch lấy mẫu được thực hiện theo trình tự như sau:

* Bước 1: Bộ nhận đợi cho đến khi xuất hiện tín hiệu start chuyển xuống mức logic ‘0’, sau đó bộ nhận tiến hành khởi động bộ đếm lấy mẫu.
* Bước 2: Khi bộ đếm đạt giá trị bằng 7 tương ứng với 8 xung (đếm từ 0), thì tín hiệu vào đang ở điểm giữa của bit START. Lúc này bộ nhận xóa giá trị của bộ đếm lấy mẫu về 0 và bắt đầu khởi động lại để đếm bit dữ liệu đầu tiên D0.
* Bước 3: Khi bộ đếm đạt đến giá trị đếm là 15 tương ứng với 16 xung (đếm từ 0), khi đó tín hiệu vào đã đã đạt đến điểm giữa của bit giữ liệu đầu tiên D0. Bộ nhận tiến hành ghi nhận giá trị này vào thanh ghi để lưu trữ và khởi động lại bộ đếm tiến hành lưu trữ các dữ liệu tiếp theo.



1. Sườn xung tại vị trí nhận giá trị

* Bước 4: Khối thu lặp lại bước 3 với N-1 lần đẻ nhận những bit dữ liệu còn lại.
* Bước 5: Nếu có bit PARITY được sử dụng thì lập lại bước 3 thêm một lần đẻ lấy giá trị bit PARITY.
* Bước 6: Lặp lại bước 3 M lần để lấy bit STOP từ khối phát. Hoàn thành nhận 1 khung dữ liệu.

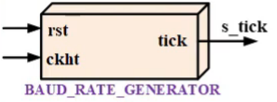


1. Xung tick tại vị trí nhận giá trị

Có thể xem như mạch lấy mẫu thực hiện chức năng của tín hiệu xung clock, thay vì sử dụng cạnh lên để xác định tín hiệu vào tồn tại, ta sử dụng xung lấy mẫu để ước lượng điểm giữa của mỗi bit.

* 1. Bộ Tạo Tốc Độ Baud

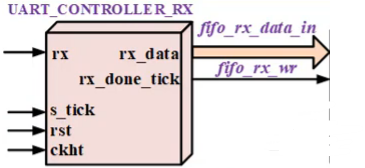
Mạch này sẽ phát ra tidn hiệu xung để lấu mẫu với tần số bằng 16 lần tần tốc độ baud đã chọn. Giả sử xung clock hệ thống là 50Mhz thì mạch tạo tốc độ baud chính là mạch đếm mod 163. Hay chính là mạch chia tần 50Mhz xuống 153600hz.



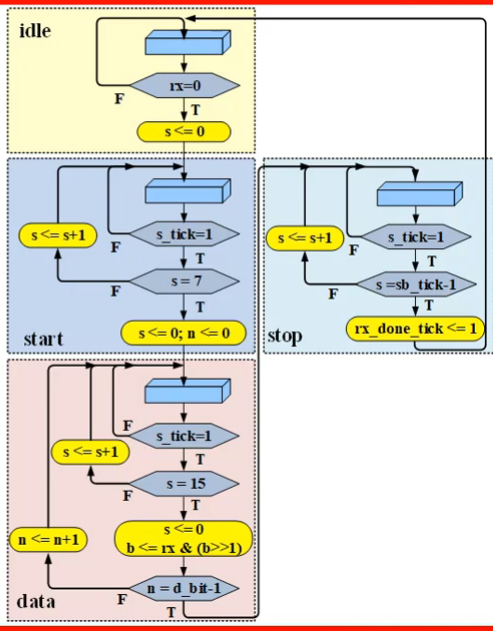
1. Khối tạo tốc độ baud

Tốc độ lấy mẫu bằng 19200\*16 = 307200 xung/1s, còn gọi là xung tick đặt tên theo sơ đồ khối.

* 1. Bộ Nhận UART



1. Điều khiển khối nhận UART
   * 1. Xây dựng lưu đồ ASMD cho UCR

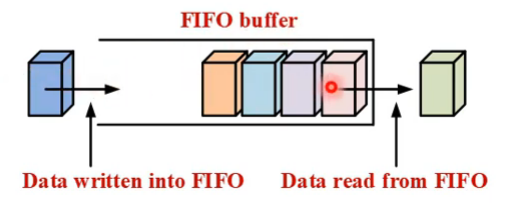


1. Lưu đồ thuật toán khối UCR
   * 1. Code VHDL cho khối UCR
   1. MẠCH GIAO TIẾP FIFO\_RX

Trong hệ thống lớn, khối UART thường là một ngoại vi để phục vụ cho truyền dữ liệu nối tiếp. Hệ thống chính sẽ kiểm tra trạng thái của nó theo chu kỳ thời gian để nhận và xử lý tín hiệu được nhận. Mạch giao tiếp này có hai chức năng:

* Mạch tạo ra tín hiệu báo sẵn sàng cho việc nhận dữ liệu mới và ngăn chặn việc nhận dữ liệu lặp lại nhiều lần.
* Mạch tạo một bộ đệm cho hệ thống chính.

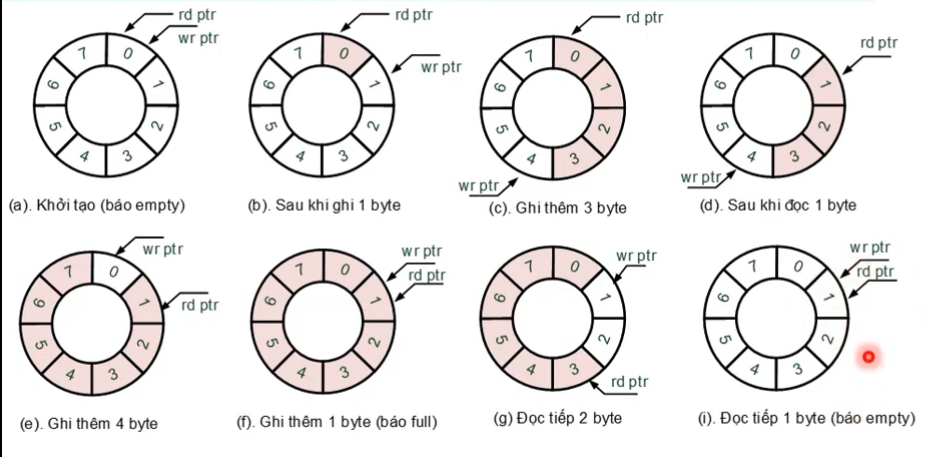
Bộ đệm FIFO là một vùng nhớ đệm dùng để lưu trữ dữ liệu nhận về từ khối UCR. Dung lượng bộ đếm có thể được thiết kế tùy ý để lưu trữ dữ liệu. Trong bộ đệm này thì dữ liệu được lưu vào trước thì sẽ được đọc ra trước.



1. Mô hình bộ đệm FIFO
   * 1. Các tín hiệu ghi dữ liệu vào bộ đệm FIFO

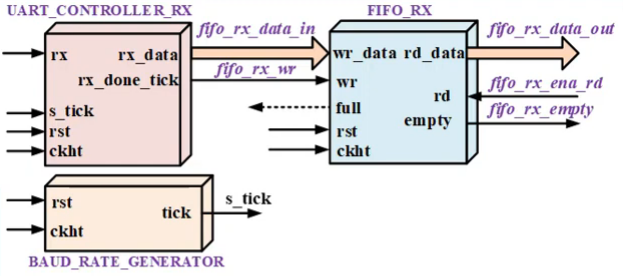
|  |  |
| --- | --- |
| * wr\_data : là ngõ vào nhận dữ liệu 8bit dùng để lưu vào FIFO. * wr : là tín hiệu điều khiển ghi dữ liệu vào FIFO. * full : là tín hiệu báo trạng thái bộ đệm FIFO đầy hay không đầy ( tích cực mức ‘1’). | 1. Khối FIFO |

* + 1. Các tín hiêu đọc dữ liệu ra từ bộ đệm FIFO
* rd\_data : là ngõ ra dữ liệu 8bit khi thực hiện đọc dữ liệu từ FIFO.
* rd : là tín hiệu điều khiển đọc dữ liệu từ FIFO.
* empty : là tín hiệu báo trạng thái bộ đệm FIFO rỗng hay không rỗng (tích cực mức ‘1’).
  + 1. Hoạt động đọc/ghi dữ liệu FIFO có 8 ô nhớ



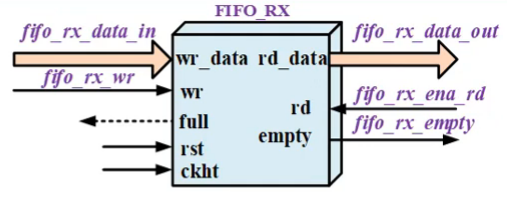
1. Các trạng thái của khối FIFO

* rd\_ptr : là con trỏ đọc dữ liệu.
* wr\_ptr : là con trỏ ghi dữ liệu.
  + 1. Ứng dụng của FIFO trong hệ thống UART

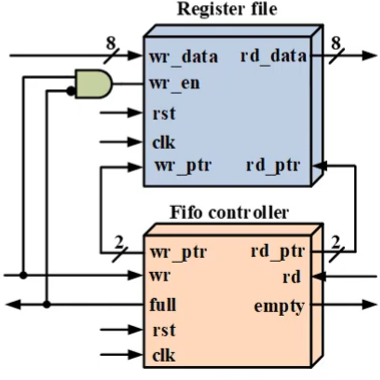


1. Ứng dụng của FIFO trong hệ thống UART

* UCR giao tiếp với FIFO để ghi dữ liệu:
* Tín hiệu “rx\_data” nối đến ngõ vào “wr\_data” của FIFO để ghi dữ liệu sau nhận được vào FIFO.
* Tín hiệu “rx\_done\_tick” nối đến ngõ “wr” của FIFO để điều khiển FIFO ghi dữ liệu.
* Tín hiệu “full” không sử dụng trong ứng dụng này.
* Hệ thống chính (HTC) giao tiếp với FIFO để đọc dữ liệu:
* Tín hiệu “rx\_data” của FIFO nối đến ngõ vào của HTC để nhận dữ liệu.
* Tín hiệu “rd” của FIFO nối đến tín hiệu điều khiển đọc của HTC.
* Tín hiệu “empty” của FIFO sẽ nối đến HTC để báo trạng thái.
  + 1. Sơ đồ khối

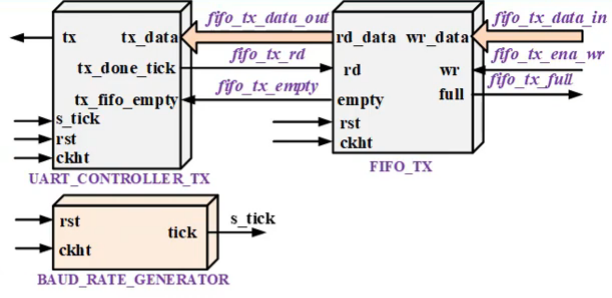


1. Khối FIFO\_RX
   * 1. Code VHDL cho bộ đệm FIFO\_RX



1. Modul của khối FIFO\_RX
2. THIẾT KẾ BỘ PHÁT UART

Cấu trúc bộ phát UART gồm 3 phần:



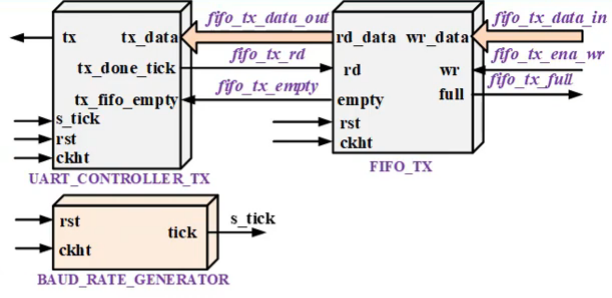
1. Sơ đồ khối phát UART

* UART Controller Transmitter (UCT): Mạch phát dữ liệu lên đường truyền nối tiếp.
* Baud Rate Generator (BRG): Tạo ra xung lấy mẫu.
* Interface Circuit (IC): Mạch giao tiếp.

Mạch phát dữ liệu cũng tương tự bộ phát dữ liệu nhưng ngược lại. hệ thống chính ghi vào bộ đệm FIFO và bộ phát UART sẽ đọc dữ liệu từ bộ đệm FIFO để phát dữ liệu đi.

Bộ phát UART có thanh ghi dịch dữ liệu từng bit ở chân “tx” ở một tốc độ cố định. Tốc độ này được điều khiển bởi tín hiệu tick của khối BRG. Thường thì khối nhận và khối phát trong UART sẽ dùng chung khối BRG.

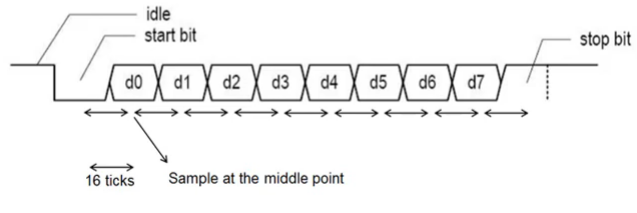
Các tín hiệu vào ra:



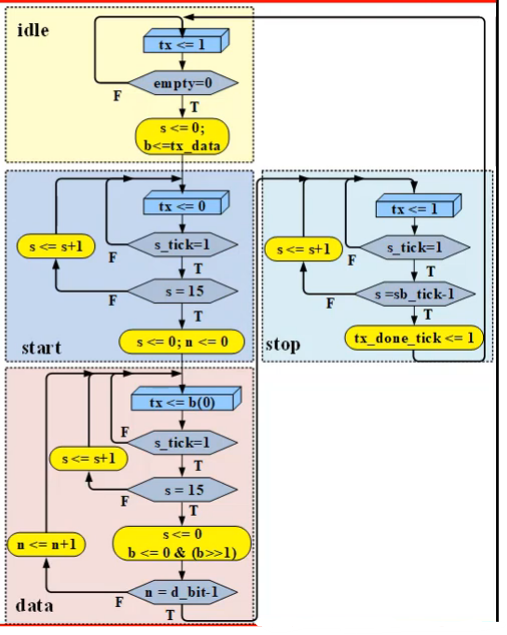
1. Sơ đồ khối phát UART

* Tín hiệu vào “tx\_data” 8 bit nhận dữ liệu từ khối FIFO để truyền đi.
* Tín hiệu ra “tx\_done\_tick” dùng để báo hiệu khi truyền xong một byte dữ liệu.
* Tín hiệu vào “tx\_fifo\_empty” dùng để kiểm tra FIFO rỗng hay không.
* Tín hiệu vào “s\_tick” dùng để nhận xung lấy mẫu từ khồi BRG.
* Tín hiệu ra “tx” dùng để xuất dữ liệu nối tiếp từng bit.
* Các tín hiệu còn lại là “clkht” và “rst”.
  1. Khối phát UART
     1. Xây dựng lưu đồ ASMD cho UCT

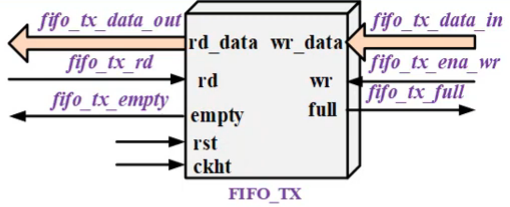
Thực hiện các trạng thái ILDE, START, DATA, STOP để tiến hành truyền nối tiếp từng bit. Một bit sẽ được dịch ra mỗi lần sau khi có 16 xung tick. Khi truyền hoàn thành thì khối UCT sẽ tạo ra 1 xung cho tín hiệu tx\_done\_tick.

****

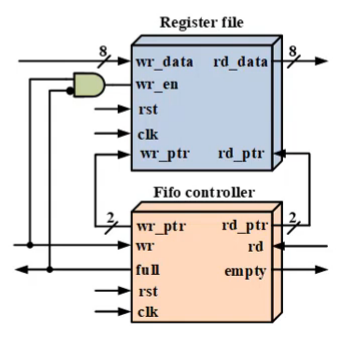
1. Khung dữ liệu của bộ phát UART



1. Lưu đồ thuật toán khối UCT
   * 1. Code VHDL cho khối UCT
   1. Khối FIFO\_TX
      1. Sơ đồ khối

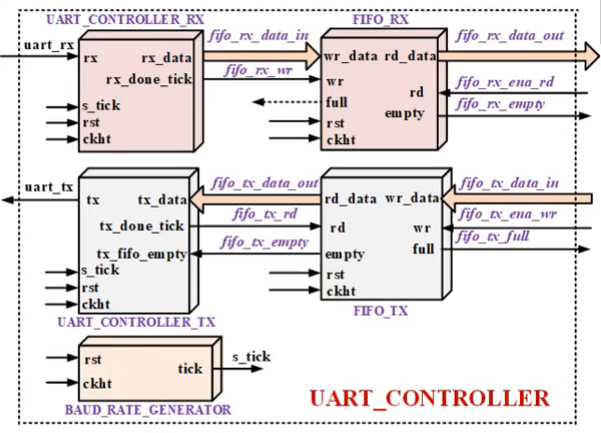


1. Khối FIFO\_TX
   * 1. Code VHDL cho bộ đệm FIFO\_TX



1. Modul của khối FIFO\_TX
2. THIẾT KẾ HỆ THỐNG UART HOÀN CHỈNH
   1. Sơ đồ khối

Bằng cách kết hợp bộ phát và bộ nhận sẽ tạo thành hệ thống UART hoàn chỉnh.

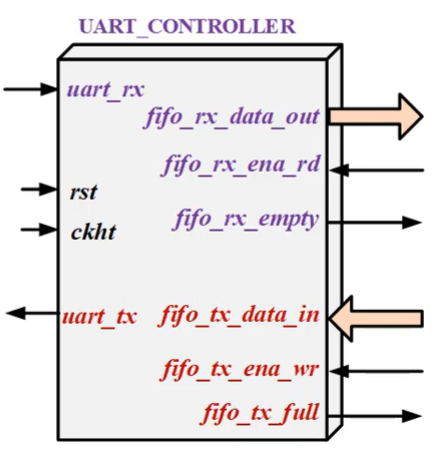


1. Sơ đồ khối hệ thống UART hoàn chỉnh

Tín hiệu xung **clkht** và **rst** cung cấp cho tất cả các khối.

Khối tạo tốc độ baud cung cấp tín hiệu **s\_tick** cho khối UART\_CONTROLLER\_RX (UCR) và khối UART\_CONTROLLER\_TX (UCT).

* 1. Các tín hiệu vào ra



1. Khối UART hoàn chỉnh

* Nhận dữ liệu:
* uart\_rx : ngõ vào nhận dữ liệu nối tiếp để đưa tới FIFO\_RX
* fifo\_rx\_data\_out : ngõ ra 8 bit đọc dữ liệu từ FIFO­\_RX để đưa tới hệ thống chính.
* fifo\_rx\_ena\_rd : ngõ vào nhận tín hiệu điều khiển yêu cầu đọc dữ liệu từ FIFO\_RX.
* fifo\_rx\_empty : ngõ ra báo trạng thái empty của FIFO\_RX.
* Truyền dữ liệu:
* uart\_tx : ngõ ra phát dữ liệu nối tiếp.
* fifo\_tx\_data\_in : ngõ vào 8 bit nhận dữ liệu từ hệ thống chính để đưa tới FIFO\_TX.
* fifo\_tx\_ena\_wr : ngõ vào nhận tín hiệu điều khiển yêu cầu ghi dữ liệu từ FIFO\_TX.
* fifo\_tx\_full : ngõ ra báo trạng thái full của FIFO\_TX.