PHẦN NÀY CHỈ TẬP TRUNG VÀO VIỆC ĐIỀU KHIỂN ORX RECEIVER

# Điều khiển ORX Receiver

Đường ORX có thể được chia sẻ bởi nhiều ứng dụng khác nhau, ví dụ như:

* Digital Pre-Distortion (DPD) sử dụng đường ORX để lấy dữ liệu ngay sau Power Amplifier (PA). DPD sẽ sử dụng dữ liệu này để tính toán và bù lại méo được sinh ra do ảnh hưởng của PA đồng thời làm tăng hiệu suất của PA.
* Tracking Calibration cũng sử dụng đường ORX để lấy thông tin của hệ thống, từ đó liên tục calibrate các phần tử bên trong chain giúp đảm bảo các thông số LO leakage hay QEC luôn ở trong mức cho phép.

Tại một thời điểm, đường ORX chỉ hỗ trợ được một ứng dụng duy nhất. Do đó, việc trao quyền sử dụng đường ORX cho ứng dụng nào, tại thời điểm nào, đóng vai trò rất quan trọng trong việc đảm bảo chức năng của toàn bộ hệ thống.

Ví dụ: một hệ thống đồng thời chạy hai ứng dụng: “Tracking Calibration” và “Digital Pre-Distortion”. Mỗi ứng dụng có những yêu cầu về timing như sau để đảm bảo chức năng của chúng hoạt động bình thường:

* “Tracking Calibration” yêu cầu phải sử dụng đường ORX tối thiểu 400 ms cho mỗi 2s.
* “DPD” yêu cầu phải dùng tối thiểu liên tục 10 ms cho mỗi lượt sử dụng đường ORX.

Việc điều khiển ORX receiver để trao quyền cho 2 ứng dụng trên phải tuân thủ chặt chẽ hai yêu cầu này.

## Làm thế nào để điều khiển ORX Receiver

Có hai phương pháp được dùng đó là:

* Sử dụng API function:
  + ARM của Zynq sẽ là thành phần đóng vai trò điều khiển trao quyền sử dụng đường ORX cho các ứng dụng có nhu cầu. Quá trình trao quyền được diễn ra theo các bước dưới đây:
    - ARM của Zynq sẽ gửi lệnh thông qua giao tiếp SPI tới ARM trong AD9371 yêu cầu trao quyền sử dụng đường ORX cho một ứng dụng cụ thể.
    - ARM trong AD9371 sẽ làm gì tiếp ??????????????
* Sử dụng PIN CONTROL
  + IP core trong FPGA sẽ đóng vai trò điều khiển.
  + Tuy nhiên trước đó, ARM trong AD9371 phải được báo (hoặc được cấu hình) 2 thông tin sau:
    - Đường GPIO nào sẽ được IP core sử dụng để điều khiển
    - Báo cho ARM (AD9371) biết sẽ sử dụng PIN CONTROL mode để điều khiển ORX receiver

***MYKONOS\_setArmGpioPins(mykonosDevice\_t \*device)***

*Báo chân GPIO nào sẽ được sử dụng bởi IP core trong FPGA.*

***MYKONOS\_setRadioControlPinMode(mykonosDevice\_t \*device)***

*Chọn mode điều khiển ORX receiver (API mode hoặc PIN CONTROL mode) – thông tin cụ thể được mô tả ở bảng dưới đây.*

|  |
| --- |
|  |
| *Figure 1: ARM GPIO configuration Structure member* |

Cấu hình đối tượng “***orxPinMode***” để chọn một trong hai mode “API MODE” hay “PIN CONTROL MODE” để điều khiển ORX receiver (trao quyền sử dụng ORX cho các ứng dụng khác nhau).

Ngoài ra, cấu hình đối tượng “***txRxPinMode***” để “tắt/bật” TX/RX chain bằng “API mode” hoặc bằng “PIN CONTROL MODE”

Câu hỏi: Khi nào sử dụng API và khi nào sử dụng PIN CONTROL?

* PIN CONTROL được sử dụng cho các ứng dụng có yêu cầu chặt về timing. Trong thực tế, nếu Transceiver hoạt động trong FDD (Frequency Division Duplex) mode, người ta có thể sử dụng API để điều khiển ORX receiver. Nếu hoạt động ở TDD (Time Division Duplex), thì bắt buộc việc trao quyền phải được kiểm soát thông qua PIN CONTROL.

# ENABLE/ DISABLE TX và RX chain trong TDD mode

Pin control mode được dùng để enable và disable TX/RX chain

PIN CONTROL MODE

IP core có thể setup 4 mode sau

|  |  |  |
| --- | --- | --- |
| **Mode** | **CODE** | **Mô tả** |
| OFF | X”000” | Đường ORX không được sử dụng |
| ORX1 |  | Đường ORX sẽ lấy data từ Antenna số 1 của AD9371 |
| ORX2 |  | Đường ORX sẽ lấy data từ Antenna số 2 |
| ARM CALIB |  | Đường ORX được dụng cho chức năng Arm-based Calibration |

Process

1. Ở direct hardware control mode khi chọn feedback antenna, DPD software sẽ gửi yêu cầu cần lấy dữ liệu từ đường ORx\_n thông qua đường m\_axis\_srx\_ctrl (\_tdata sẽ mang thông tin antenna nào đang được yêu cầu, \_tvalid = ‘1’).

m\_axis\_srx\_ctrl\_tdata sẽ có format như sau:

|  |
| --- |
|  |
| Phân bố thông tin tại 8 bit của m\_axis\_srx\_ctrl\_tdata |

Trong đó:

* + Trường Request type có các option như sau:

|  |
| --- |
|  |

* + Trường Antenna đưa ra số thứ tự của antenna mà DPD software yêu cầu. Tùy thuộc từng mode hoạt động của DPD software mà, thông số này thay đổi theo những cách khác nhau:
    - Linear mode: số thứ tự sẽ tăng dần từ 0 đến 7 sau đó lại quay lại 0
    - Random mode: số thự tự này sẽ thay đối ngẫu nhiên

Trong thiết kế này, người viết sẽ chọn “Linear mode” để đơn giản hóa việc lập lịch cũng như chủ động hơn về mặt điều khiển quá trình internal calibration ở từng RF transceiver.

1. *Yêu cầu (1) từ DPD software sẽ được coi là được chấp nhận khi m\_axis\_srx\_ctrl\_tready và m\_axis\_srx\_ctrl\_tvalid đều bằng ‘1’*.

Chú ý: \_tready không phải lúc nào cũng bằng ‘1’. Câu hỏi đặt ra là trạng thái tín hiệu của đường m\_axis\_srx sẽ như thế nào? Hình dưới đây sẽ trả lời cho câu hỏi trên:

|  |
| --- |
|  |
| M\_axis\_srx\_ctrl stream behavior when an external system holds off tready |

Ngoài ra, một điều (như phần chú ý trong pg706, page 23 – dpd user guide) là \_tdata chỉ valid khi \_tvalid =’1’. Mọi thời điểm khác, **không thể** sử dụng \_tdata để chọn antenna. Do đó, trong thiết kế này, tready chỉ được bật ( =’1’) khi khối hardware điều khiển nhận được ack đã switch kênh thành công từ transceiver. DPD software hiểu rằng: yêu cầu feedback của nó đã được chấp thuận. Câu hỏi đặt ra:

* Data tại thời điểm ngay sau: tready = ‘1’ có phải là data từ antenna yêu cầu hay không?

1. Data từ ORX sẽ được forward về cho đường s\_axis\_srx của DPD. Trong đó:
   * \_tdata chứa dữ liệu feedback về
   * \_tuser (8 bits) chứa thông tin: hiện tại antenna nào đang đẩy thông tin về.

Thiết kế tổng quan:

* Software khi khởi động sẽ thực hiện trao quyền điều khiển INTERNAL CALIB, mode ORX1-2 cho dedicated hardware. Thực chất, đối tượng điều khiển trực tiếp những quy trình trên vẫn là ARM core trên AD9371. Tuy nhiên, sự khác nhau ở đây nằm ở:
  + ***API mode***: đối tượng giao tiếp với ARM trên AD9371 là ARM trên SoC thông qua truyền dẫn SPI. Kênh giao tiếp nào độ trễ cao, dẫn tới latency từ lúc ARM trên SoC yêu cầu thực hiện lệnh, đến thời điểm lệnh được thực hiện hoàn thành trên ARM của 9371 thường khá cao.
  + ***GPIO mode***: đối tượng giao tiếp với ARM trên AD9371 là dedicated hardware hay chính xác là một IP core nằm trong FPGA. Phương thức giao tiếp thông qua các chân GPIO. Với giao tiếp này, lệnh được gửi ngay tới ARM trên AD9371. Phần xử lý điều khiển trên ARM của AD9371 thì vẫn như mode API.

Độ lợi về mặt thời gian của phương pháp thứ 2 so với phương pháp đầu tiên chỉ ở quá trình truyền dẫn.

Chú ý: ngay khi software trao quyền điều khiển INTERNAL CALIB hay chọn mode ORX1-2. Dedicate hardware phải cấu hình toàn bộ antenna về chế độ internal calibration ngay. Câu hỏi làm thế này hardware biết được việc này. Giả sử hardware thực hiện cấu hình trên trước khi software trao quyền thì sao?