



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

ĐỒ ÁN NGHIÊN CỨU CỨ NHÂN

Đề tài:

**Thiết kế và triển khai bộ mã hóa tích chập và giải mã
Viterbi được tăng tốc bởi phần cứng dưới dạng server**

*Giảng viên hướng dẫn :
Sinh viên thực hiện:*

**PGS.TS. Nguyễn Hữu Trung
Vũ Tuấn Minh - 20214015**

ONE LOVE. ONE FUTURE.



HUST

NỘI DUNG

- I. Tổng quan đề tài
- II. Thiết kế hệ thống
- III. Triển khai – Kiểm thử
- IV. Kết luận – Hướng phát triển



HUST

PHẦN I

Tổng quan đề tài

I. Tổng quan đề tài

1.2 Mục tiêu đề tài

- Thiết kế hệ thống server FPGA tích hợp bộ mã hóa tích chập và giải mã Viterbi.
- Cung cấp khả năng truy cập từ xa qua giao thức TCP.



I. Tổng quan đề tài

1.3 Yêu cầu của hệ thống

- Hỗ trợ tốc độ mã $1/2$, $1/3$, chiều dài ràng buộc 3-9 và tất cả các đa thức sinh.
- Độ sâu truy ngược lớn hơn 5 lần chiều dài ràng buộc.
- Kiến trúc Radix-4 để tối ưu thông lượng.
- Độ sâu truy ngược lớn hơn 5 lần chiều dài ràng buộc.

Standard	Code rate	Constraint length	Number of states
GPRS	$1/2$	5	16
GSM	$1/2$	5	16
WiMAX	$1/2$	7	64
802.11a/g	$1/2$	7	64
DVB	$1/2$	7	64
CDMA IS-95A	$1/2$	7	64
LTE	$1/3$	7	64
CDMA2000	$1/2, 1/3$	9	256
3G	$1/3$	9	256



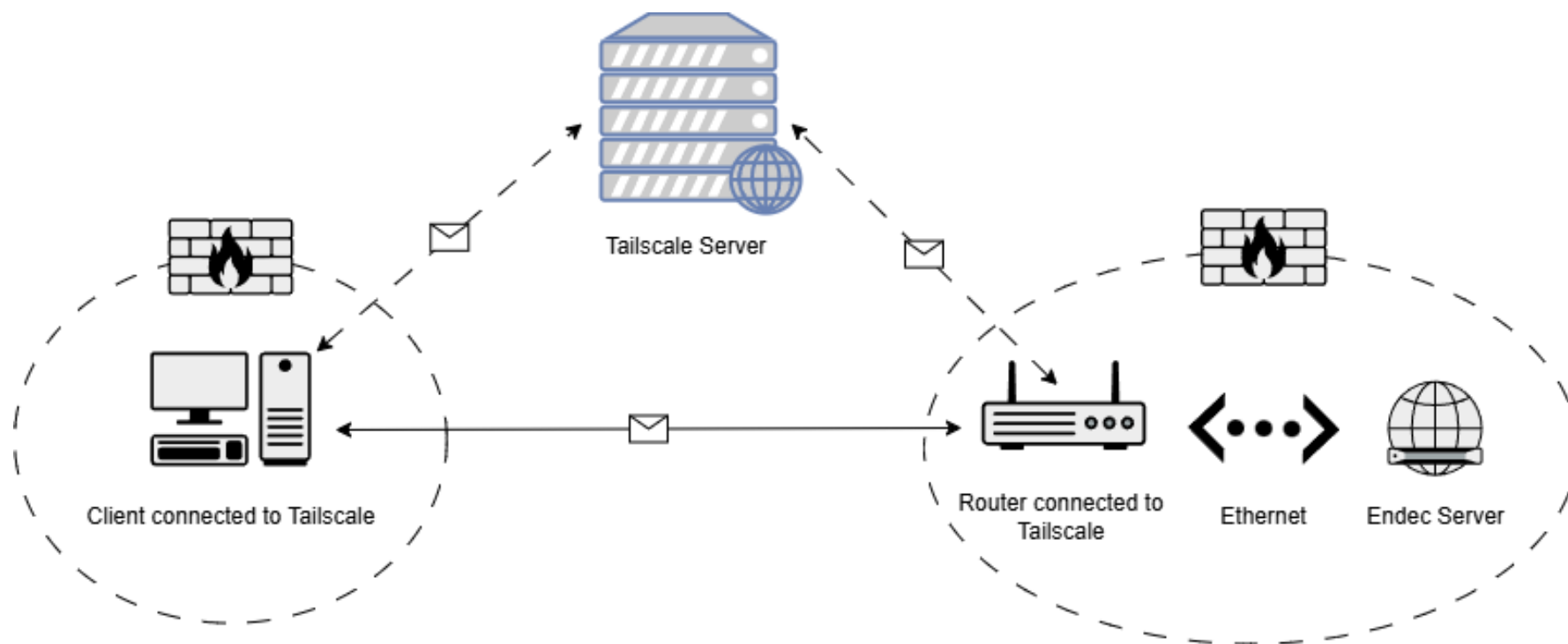
HUST

PHẦN II

Thiết kế hệ thống

II. Thiết kế hệ thống

2.1 Tổng quan hệ thống

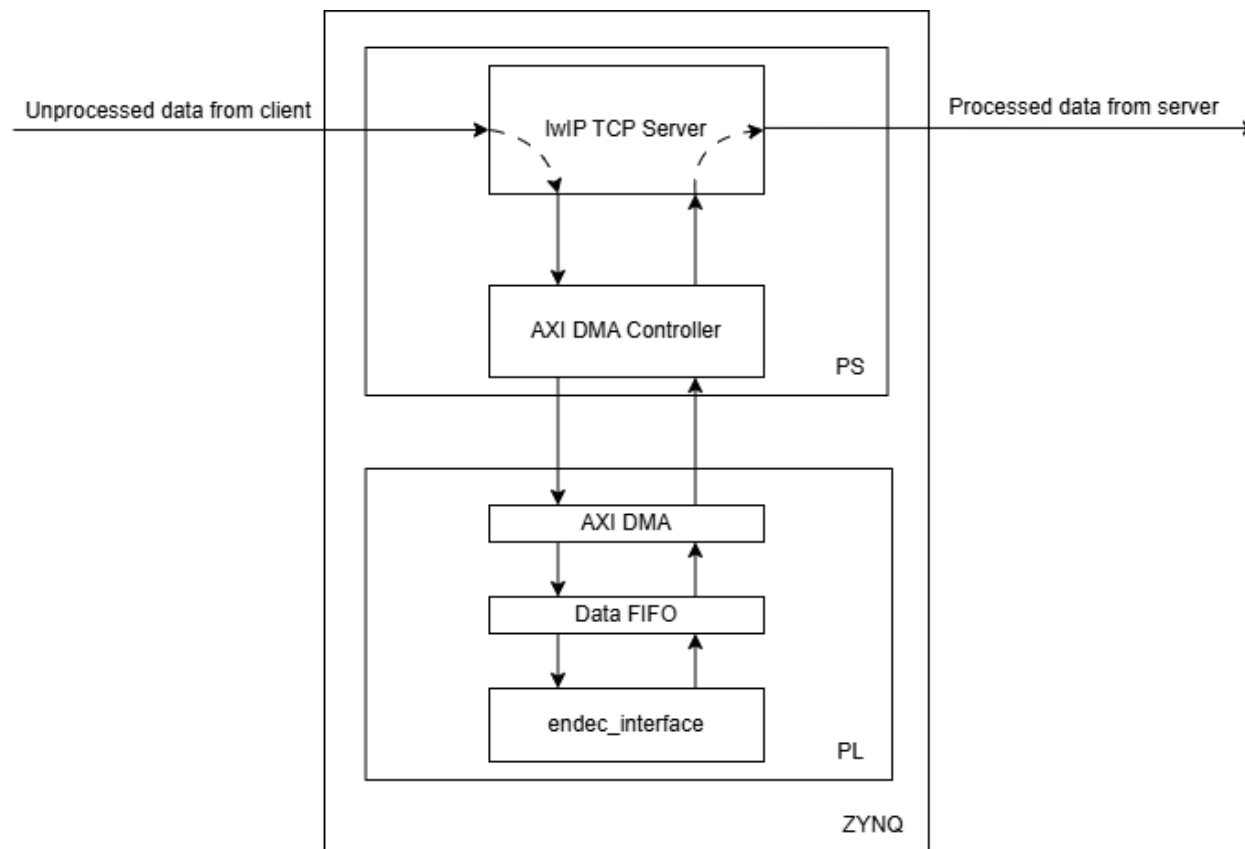


Mô hình hệ thống

II. Thiết kế hệ thống

2.1 Tổng quan hệ thống

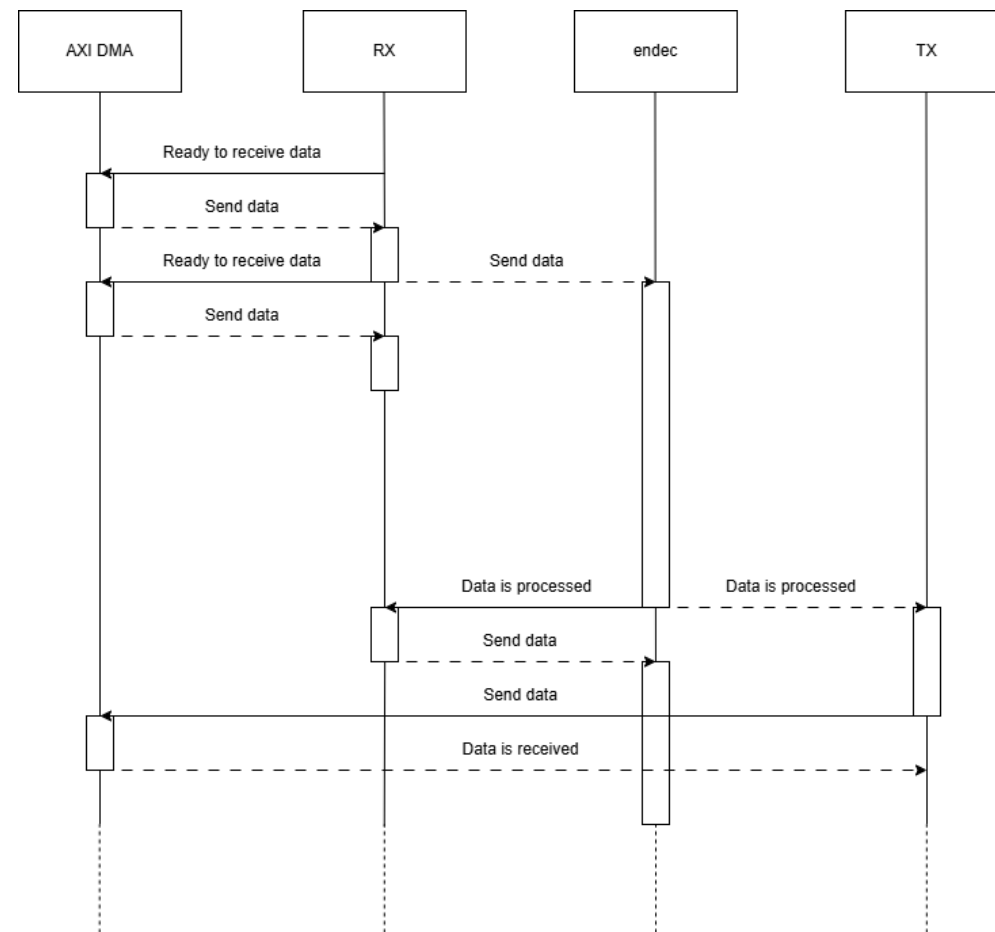
- ❑ **Phần cứng sử dụng:** PYNQ-Z2.
- ❑ **PL (FPGA):** Xử lý song song các thuật toán mã hóa/giải mã.
- ❑ **PS (ARM Cortex):** Điều khiển giao tiếp mạng và quản lý luồng dữ liệu.
- ❑ **Kết nối:** Sử dụng AXI DMA để truyền dữ liệu tốc độ cao giữa PS và PL.



II. Thiết kế hệ thống

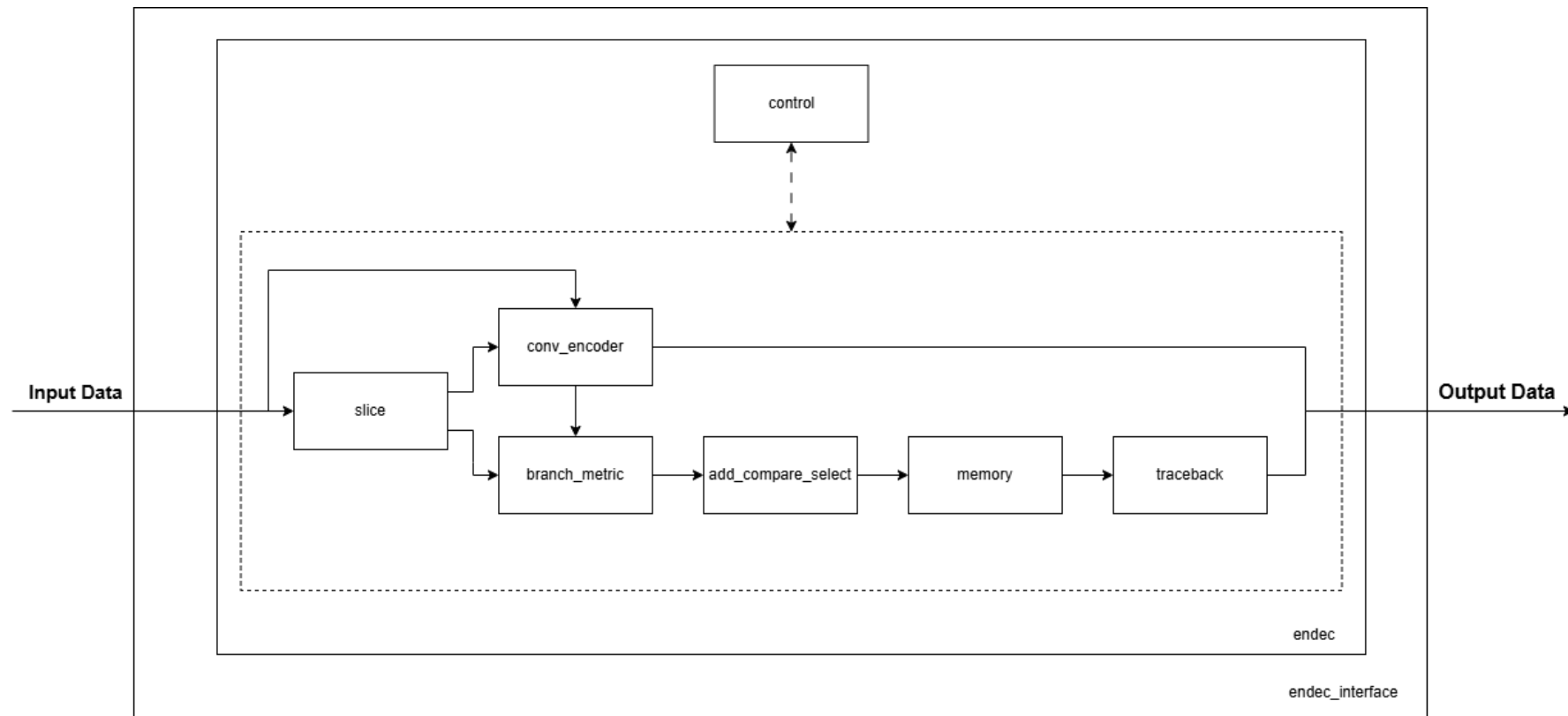
2.2 Thiết kế PL

- ❑ **Vai trò:** Giao diện trung gian kết nối AXI DMA và khối endec.
- ❑ **Cấu trúc 3 module độc lập:**
 - RX: Nhận dữ liệu từ DMA.
 - endec: Xử lý dữ liệu.
 - TX: Truyền dữ liệu đi.
- ❑ **Ưu điểm đa luồng:**
 - Song song hóa: RX nhận dữ liệu mới ngay khi endec xử lý dữ liệu cũ → Giảm độ trễ, tăng thông lượng.
 - Tự động reset sau mỗi chu kỳ → Đảm bảo ổn định, tránh lỗi tích lũy.



II. Thiết kế hệ thống

2.2 Thiết kế PL

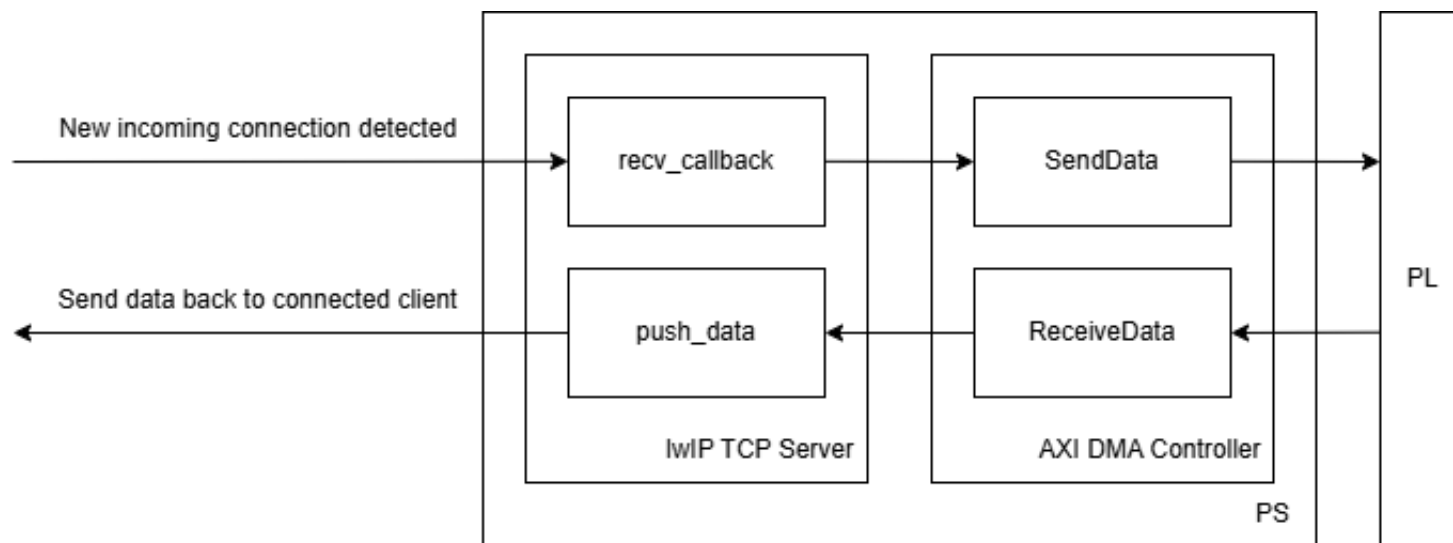


Luồng dữ liệu trong khối endec_interface

II. Thiết kế hệ thống

2.3 Thiết kế PS

- ❑ **AXI DMA Controller:** Quản lý truyền dữ liệu giữa PS và PL.
- ❑ **lwIP TCP Server:** Hỗ trợ giao tiếp mạng với client.
- ❑ **Luồng dữ liệu:** Nhận dữ liệu từ client → Truyền tới PL → Xử lý → Gửi kết quả về client.





HUST

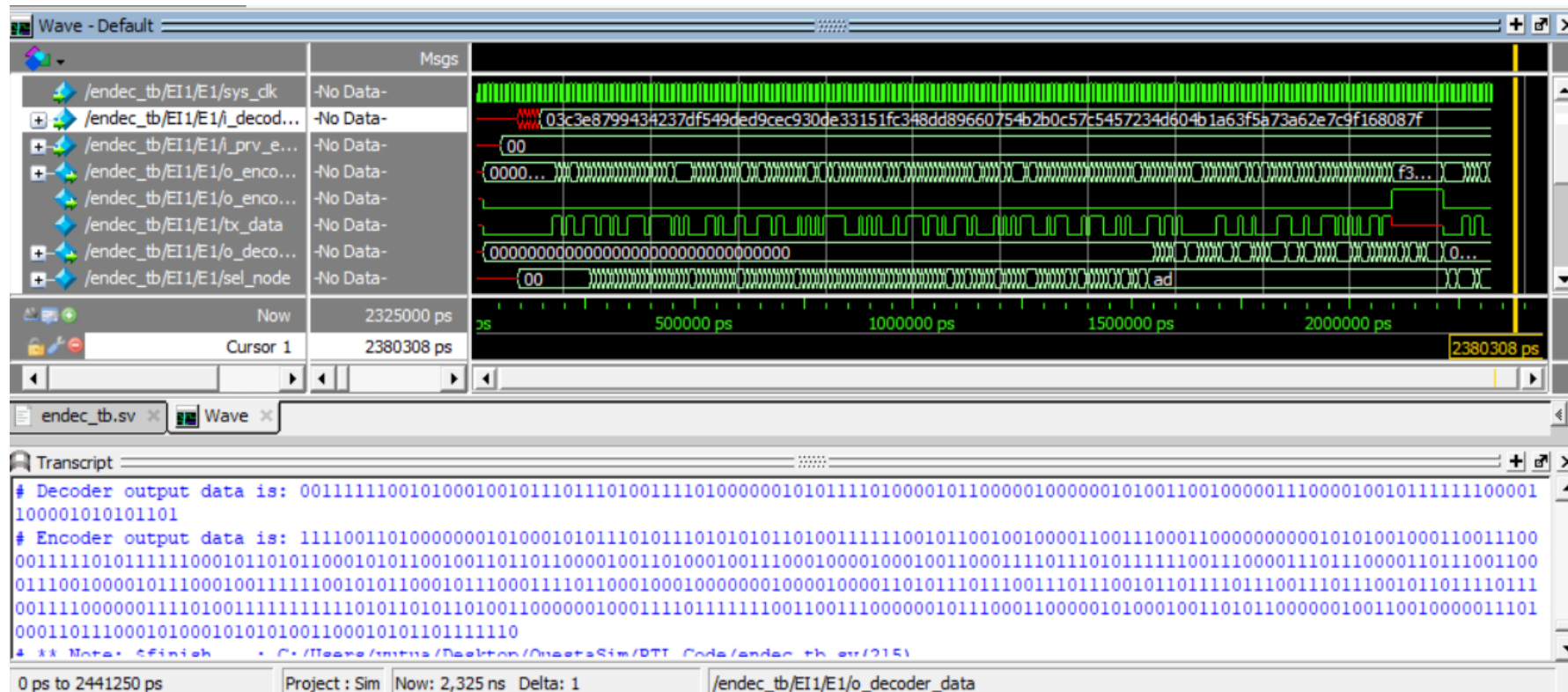
PHẦN III

Triển khai – Kiểm thử

III. Triển khai – Kiểm thử

3.1 Kiểm thử bằng mô phỏng

- ❑ **Phần mềm sử dụng:** Kiểm thử chức năng với Questa Sim.
- ❑ **Phương pháp kiểm thử:** Sử dụng bộ dữ liệu mẫu từ MATLAB để kiểm tra tính chính xác.



III. Triển khai – Kiểm thử

3.2 Kiểm thử bằng ILA

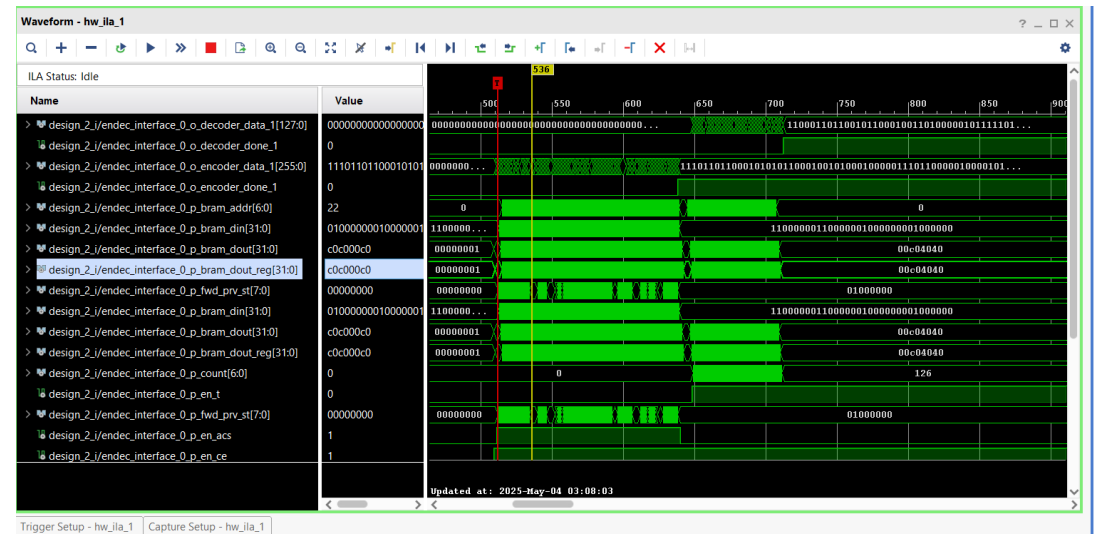
❑ Quy trình triển khai:

- Tổng hợp thành LUT/FF trên Vivado → Triển khai lên PYNQ-Z2.
- 2 bước kiểm thử phần cứng:
 - Nạp dữ liệu vào thiết kế.
 - So sánh đầu ra với dữ liệu mẫu.

❑ Công cụ sử dụng:

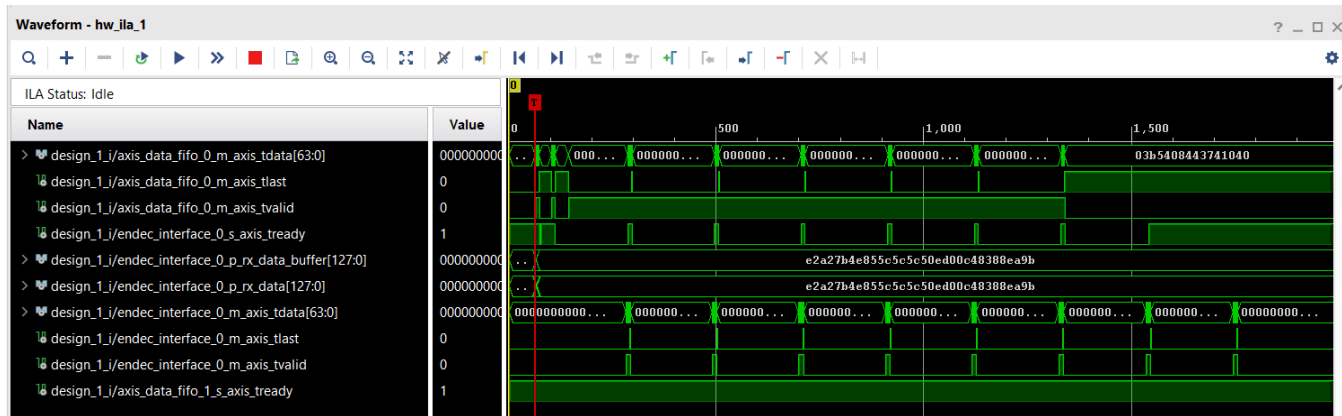
- VIO (Virtual Input/Output): Gửi dữ liệu đầu vào tới endec_interface.
- ILA (Integrated Logic Analyzer): Đọc và hiển thị tín hiệu đầu ra dạng sóng trên Vivado.

hw_vio_1					
Name	Value	Activity	Direction	VIO	
> design_2_i/vio_0_probe_out5[255:0]	[B] 1110_1101_1000_1010_1011_0001_0010_1000_1000_0011_1011_0000_0100		Output	hw_vio_1	
> design_2_i/vio_0_probe_out3[26:0]	[B] 000_0000_0011_0011_0111_1110_1101		Output	hw_vio_1	
> design_2_i/vio_0_probe_out0	[B] 1		Output	hw_vio_1	
> design_2_i/vio_0_probe_out6[7:0]	[B] 0000_0000		Output	hw_vio_1	
> design_2_i/vio_0_probe_out4[127:0]	[B] 1100_0110_1100_1011_0001_0011_0100_0001_0111_1101_1011_1011_0011		Output	hw_vio_1	
> design_2_i/vio_0_probe_out2	[B] 0		Output	hw_vio_1	
> design_2_i/vio_0_probe_out1	[B] 1		Output	hw_vio_1	
> design_2_i/endec_interface_0_o_encoder_done	[B] 1		Input	hw_vio_1	
> design_2_i/endec_interface_0_o_encoder_data[255:0]	[B] 1110_1101_1000_1010_1011_0001_0010_1000_1000_0011_1011_0000_0100_0010		Input	hw_vio_1	
> design_2_i/endec_interface_0_o_decoder_done	[B] 1		Input	hw_vio_1	
> design_2_i/endec_interface_0_o_decoder_data[127:0]	[B] 1100_0110_1100_1011_0001_0011_0100_0001_0111_1101_1011_1011_0011_0001		Input	hw_vio_1	



III. Triển khai – Kiểm thử

3.3 Kiểm thử trực tiếp



Client gửi dữ liệu

```
PS C:\Users\vutua\Downloads> python test_tcp.py
Server Connected
Data packet received value is: 2af70d6bf067d37ed41dcca704c38b4ff3
40515d755a7e59219c600548ce1f5f8b58ac9b61344e21131eebf38770dcc7217
13f2b171ec440421aee772dee772dee781e9ffd6b4c08f7f3381718289ac09907
46e28aa62b7e
Encoder output data is: f340515d755a7e59219c600548ce1f5f8b58ac9b6
1344e21131eebf38770dcc721713f2b171ec440421aee772dee772dee781e9ffd
6b4c08f7f3381718289ac0990746e28aa62b7e
Decoder output data is: 2af70d6bf067d37ed41dcca704c38b4f
Received 704 bytes total, 704 in this recv
Finished receiving data!
PS C:\Users\vutua\Downloads>
```

Client nhận dữ liệu

III. Triển khai – Kiểm thử

3.3 Kiểm thử trực tiếp

```
PowerShell
PS C:\Users\vutua\Downloads> python tcp_transmission.py
----- lwIP TCP Encoder-Decoder Server CONNECTED -----
[STATUS] Sending data to Server
[STATUS] Data received from Server
Encoder output data is: f340515d755a7e59219c600548ce1f5f8b58a
c9b61344e21131eebf38770dcc721713f2b171ec440421aee772dee772dee
781e9ffd6b4c08f7f3381718289ac0990746e28aa62b7e
Decoder output data is: 2af70d6bf067d37ed41dcca704c38b4f
PS C:\Users\vutua\Downloads>
```

Kiểm thử client cùng mạng LAN

```
PowerShell
PS C:\Users\vutua\Downloads> python tcp_transmission.py
----- lwIP TCP Encoder-Decoder Server CONNECTED -----
[STATUS] Sending data to Server
[STATUS] Data received from Server
Encoder output data is: f340515d755a7e59219c600548ce1f5f8b58a
c9b61344e21131eebf38770dcc721713f2b171ec440421aee772dee772dee
781e9ffd6b4c08f7f3381718289ac0990746e28aa62b7e
Decoder output data is: 2af70d6bf067d37ed41dcca704c38b4f
PS C:\Users\vutua\Downloads> python tcp_transmission.py
----- lwIP TCP Encoder-Decoder Server CONNECTED -----
[STATUS] Sending data to Server
[STATUS] Data received from Server
Encoder output data is: f340515d755a7e59219c600548ce1f5f8b58a
c9b61344e21131eebf38770dcc721713f2b171ec440421aee772dee772dee
781e9ffd6b4c08f7f3381718289ac0990746e28aa62b7e
Decoder output data is: 2af70d6bf067d37ed41dcca704c38b4f
PS C:\Users\vutua\Downloads>
```

Kiểm thử client kết nối với Tailscale

3.4 Kiểm thử toàn diện

- Tự động sinh 15000 khung dữ liệu bằng MATLAB.
- Tham số đầu vào được ngẫu nhiên hóa theo khoảng giá trị cho trước.
- Truyền dữ liệu đến Endec Server qua giao thức thiết lập sẵn.

Tham số	Khoảng giá trị
Chiều dài ràng buộc	4 : 9
Tốc độ mã	$1/2 : 1/3$
Đa thức sinh cho từng đầu ra	$0 : 2^9 - 1$
Trạng thái trước của thanh ghi dịch	$0 : 2^8 - 1$
Dữ liệu cần mã hóa	$0 : 2^{192} - 1$
Dữ liệu cần giải mã	$0 : 2^{384} - 1$

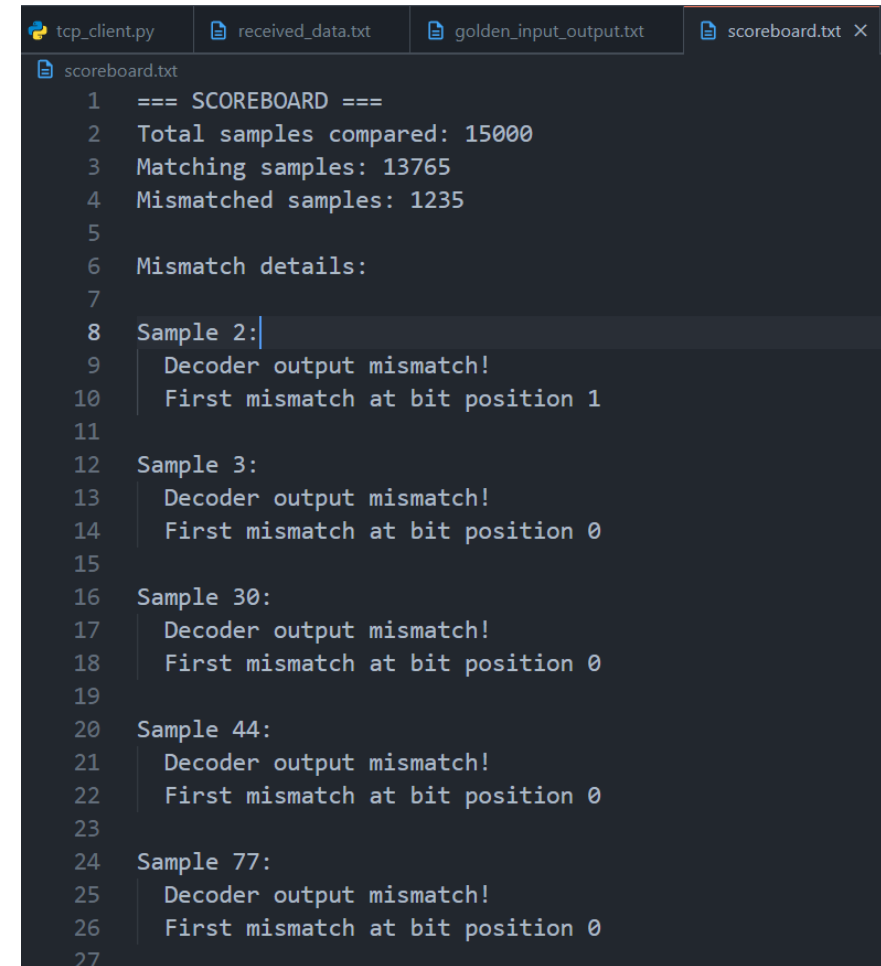
3.4 Kiểm thử toàn diện

❑ Hiện tượng quan sát:

- Dữ liệu mã hóa luôn chính xác.
- 8.23% lỗi giải mã xuất hiện.

❑ Nguyên nhân nghi ngờ:

- Sử dụng đa thức sinh không tối ưu.
- Ngẫu nhiên hóa đa thức sinh có thể dẫn đến:
 - Đa thức không đạt chuẩn khoảng cách tự do.
 - Xuất hiện đa thức gây lỗi thảm họa (catastrophic).



```
tcp_client.py  received_data.txt  golden_input_output.txt  scoreboard.txt X
scoreboard.txt
1  === SCOREBOARD ===
2  Total samples compared: 15000
3  Matching samples: 13765
4  Mismatched samples: 1235
5
6  Mismatch details:
7
8  Sample 2:
9  Decoder output mismatch!
10 First mismatch at bit position 1
11
12 Sample 3:
13 Decoder output mismatch!
14 First mismatch at bit position 0
15
16 Sample 30:
17 Decoder output mismatch!
18 First mismatch at bit position 0
19
20 Sample 44:
21 Decoder output mismatch!
22 First mismatch at bit position 0
23
24 Sample 77:
25 Decoder output mismatch!
26 First mismatch at bit position 0
27
```

3.4 Kiểm thử toàn diện

❑ Giải pháp khắc phục:

- Điều chỉnh thuật toán sinh ngẫu nhiên trong MATLAB.
- Chỉ sử dụng các đa thức sinh đã được kiểm chứng từ hệ thống thông tin thực tế.

❑ Kết quả kiểm thử:

- Đạt độ chính xác 100%.
- Khẳng định nguyên nhân lỗi trước đó:
 - Do đặc tính đa thức sinh.
 - Không phải do lỗi hệ thống mã hóa/giải mã.

Chiều dài ràng buộc	Tốc độ mã 1/2	Tốc độ mã 1/3
4	15 – 17	15 – 17 – 13
5	23 – 35	25 – 33 – 37
6	53 – 75	55 – 64 – 71
7	133 – 171	133 – 145 – 171
8	247 – 371	247 – 371 – 357
9	561 – 753	557 – 663 – 711



```
tcp_client.py  received_data.txt  golden_input_output.txt  scoreboard.txt X
scoreboard.txt
1  === SCOREBOARD ===
2  Total samples compared: 15000
3  Matching samples: 15000
4  Mismatched samples: 0
5
6  All samples matched perfectly!
7  |
```

3.4 Kiểm thử toàn diện

❑ Độ trễ xử lý:

- Trung bình: 0.1 giây/15000 khung dữ liệu.
- Tính từ khi client gửi dữ liệu đến khi nhận kết quả.

❑ Thông số kiểm thử mỗi khung:

- 192 bit đầu vào (mã hóa).
- 128 bit đầu ra (giải mã).

❑ Kết quả thông lượng:

- Mã hóa: 28.8 Mbps.
- Giải mã: 19.2 Mbps.

```
PS C:\Users\vutua\Desktop\Golden_Data> python tcp_client.py
----- lwIP TCP Encoder-Decoder Server CONNECTED -----
[STATUS] Sending 1 large packet (1200000 bytes)
[STATUS] Processed 15000 samples saved to received_data.txt
[STATUS] Scoreboard generated in scoreboard.txt
time.perf_counter: 0.102676 seconds
[STATUS] All 1320000 bytes received and processed
PS C:\Users\vutua\Desktop\Golden_Data>
```

$$\text{Thông lượng mã hóa} = \frac{\text{Số bit mã hóa đầu vào}}{\text{Thời gian xử lý}}$$

$$\text{Thông lượng giải mã} = \frac{\text{Số bit giải mã đầu ra}}{\text{Thời gian xử lý}}$$

3.5 Đánh giá thông số hoạt động

❑ Hiệu suất vượt trội

- Tốc độ mã hóa: 28.8 Mbps (nhanh hơn 22 lần so với MATLAB).
- Tốc độ giải mã: 19.2 Mbps (nhanh hơn 83 lần so với MATLAB).

❑ Tiết kiệm năng lượng tối ưu:

Mức tiêu thụ điện chỉ 1.64W, thấp hơn rất nhiều so với MATLAB (45W).

Tiêu chí	MATLAB	Endec Server
Quá trình cài đặt	Cần có license và tải về MATLAB	Kết nối với Mesh Network qua Tailscale
Tài nguyên yêu cầu	Lớn, không phù hợp với các hệ thống nhúng	Có kết nối mạng
Tài nguyên sử dụng	Tiêu tốn tài nguyên khi thực hiện mã hóa/giải mã	Gần như không tiêu tốn tài nguyên
Công suất tiêu thụ	45 W (CPU)*	1.64 W
Độ ổn định	Dễ gặp hiện tượng treo, nghẽn dữ liệu	Luôn hoạt động ổn định
Thông lượng xử lý	Mã hóa: 1.31 Mbps* Giải mã: 0.23 Mbps*	Mã hóa: 28.8 Mbps Giải mã: 19.2 Mbps

* Được thực hiện trên CPU Ryzen 5 5600H



HUST

PHẦN IV

Kết luận – Hướng phát triển

IV. Kết luận – Hướng phát triển

❑ Đạt được:

- Thiết kế và triển khai thành công hệ thống server FPGA với kiến trúc Radix-4.
- Đạt thông lượng cao, hỗ trợ cấu hình động và truy cập từ xa.
- Khắc phục hạn chế của các giải pháp truyền thống.

❑ Hạn chế:

- Chưa tận dụng được lõi đôi trên PS.
- AXI DMA hoạt động ở chế độ polling thay vì interrupt.

❑ Hướng phát triển:

- Tích hợp giải mã quyết định mềm.
- Sử dụng kiến trúc lai Radix-2/Radix-4 để hỗ trợ chiều dài ràng buộc 3.
- Lập trình lõi đôi trên PS theo chế độ AMP.
- Sử dụng AXI DMA với interrupt.



HUST

THANK YOU !