A Reproducible Ring-108 and Base-27 Signal–Symbol Framework for CVNSS4.0: Cyclic Algebra, Mixed Binary–Ternary Coordinates, and Acoustic 27-FSK

Bản thảo preprint nghiên cứu / Working paper v1.0 — 16 November 2026
Tác giả: Long Ngo
Dự án và mã nguồn: https://github.com/xulytiengviet/kientrucvong
Trạng thái: Bản thảo chưa bình duyệt. Phép đo chỉ là mô phỏng kênh số AWGN, chưa bao gồm truyền thực qua loa–microphone.

Tóm tắt

Nghiên cứu này tách ba đối tượng thường bị đồng nhất khi bàn về CVNSS4.0 và kiến trúc vòng: (i) lưới 108 ô mô hình hóa tổ hợp sáu lớp nguyên âm, ba nhóm dấu phụ và sáu thanh; (ii) nhóm cyclic $\mathbb{Z}_{108}$ và đẳng cấu CRT với $\mathbb{Z}_4 \times \mathbb{Z}_{27}$; (iii) kênh dữ liệu số chuyển 27 ký hiệu ASCII gồm A–Z và khoảng trắng thành dạng sóng âm thanh bằng 27-FSK, rồi giải điều chế ngược. Chúng tôi trình bày ánh xạ $n \mapsto e^{2\pi i n/108}$, chứng minh tính bảo toàn pha tương đối dưới dịch chuyển đồng thời, và phân tích phép đối cực 180 độ. Một giao thức WAV cụ thể được công bố gồm preamble, độ dài, payload và CRC-16; bên nhận sử dụng bộ dò Goertzel. Kiểm thử thuộc tính trên mọi 108 trạng thái và truyền nhận số trong điều kiện mô phỏng được cung cấp dưới dạng mã nguồn. Trong thí nghiệm có hạt giống cố định với 24 khung/mức SNR, tỷ lệ thành công bằng 100% ở các mức từ −15 dB trở lên *trong các mức được thử*, và giảm xuống 54,17% ở −20 dB; không tồn tại lời bảo đảm 100% cho mọi kênh vật lý. Công thức 108 không phải ánh xạ đơn ánh từ toàn bộ âm tiết tiếng Việt; CVNSS4.0 nguyên thủy vẫn có các trường hợp giải mã nhập nhằng. Hệ thống này là bằng chứng về tính xây dựng và khả năng tái lập của một đường truyền ký hiệu bằng sóng, không phải chứng minh về ngữ nghĩa, ASR hoặc hiệu năng NLP.

Từ khóa: CVNSS4.0; Chinese remainder theorem; cyclic groups; ring embedding; ternary digits; acoustic M-FSK; Goertzel; reversible signal–symbol conversion; reproducibility.

Abstract

This working paper separates three distinct layers: a proposed 108-cell Vietnamese vowel–modifier–tone lattice, the cyclic group $\mathbb{Z}_{108}$ together with its CRT decomposition $\mathbb{Z}_4 \times \mathbb{Z}_{27}$, and an independent acoustic 27-ary FSK transport for CVNSS4.0 ASCII symbols. We prove the circle embedding, phase-shift invariance and 180-degree antipodal operation; construct an explicit 27-FSK PCM WAV framing and decoding pipeline; and release exhaustive algebraic tests and seeded additive white Gaussian noise (AWGN) evaluations. In the fixed simulation, all frames are recovered at selected SNRs of −15 dB or higher, whereas the success rate is 54.17% at −20 dB. These bounded measurements do not support unconditional reliability claims. Crucially, the 108-state lattice is not injective on Vietnamese syllables, the supplied CVNSS4.0 codec exposes many-to-one mappings, and our single-frequency cyclic construction is not identical to multi-frequency rotary position embeddings. The contribution is a reproducible digital signal-to-symbol implementation with explicit guarantees and limits, not a universal lossless speech-to-text or semantic mapping.

1. Phạm vi và câu hỏi nghiên cứu

- **RQ1.** Có thể biểu diễn nhất quán 108 trạng thái trừu tượng bằng hai tọa độ số học gồm 2 bit và 3 trit cân bằng không? Có; chứng minh CRT và kiểm tra vét cạn.
- **RQ2.** Có thể chuyển chuỗi ký hiệu CVNSS4.0 thuộc bảng `A–Z + SPACE` thành PCM WAV rồi tái tạo chính chuỗi đó không? Có, trong giả định đồng bộ khung, biết chính xác tần số lấy mẫu và kênh được điều khiển; phải đo xác suất lỗi khi có nhiễu.

- **RQ3.** Có thể suy ra chuyển văn bản Unicode tiếng Việt bất kỳ sang sóng rồi phục hồi nguyên bản chỉ từ 108 trạng thái không? Không. Phải lưu biểu diễn ký hiệu đầy đủ và quản lý nhập nhằng ở bộ giải mã CVNSS.
- **RQ4.** Mô hình này có chứng minh CVNSS4.0 tốt hơn mã chữ khác, hoặc RoPE liên tục? Không, cần benchmark ngoài phạm vi nghiên cứu sơ bộ.

Hai không gian mã có vai trò khác nhau: **108 trạng thái** dành cho hình học/đại số của lưới nguyên âm trừu tượng; **27 ký hiệu** là bảng chữ dữ liệu thực sự được gửi qua sóng. Không đồng nhất ký tự thuộc bảng 27 với một ô đại số 108 hay với toàn bộ âm tiết.

2. Đặc tả bảng 108 ô

Giả định một lưới $G = V \times M \times T$ với $V = \{a, e, i, o, u, y\}$, $M = \{\text{plain}, \text{circumflex}, \text{hook/breve}\}$, và $T = \{0, 1, 2, 3, 4, 5\}$. Định nghĩa tọa độ $\nu \in [0, 5]$, $\mu \in [0, 2]$, $\tau \in [0, 5]$, với:

$$E_G(\nu, \mu, \tau) = 18\nu + 6\mu + \tau \in \{0, \dots, 107\}.$$

Đây là một **song ánh giữa các ô của lưới trừu tượng** và $\mathbb{Z}_{108}$. Tập con hợp lệ theo danh sách dạng chữ viết là:

Nguyên âm cơ sở	Dạng cho phép	Số ô
a	a, â, ă	18
e	e, ê	12
i	i	6
o	o, ô, ơ	18
u	u, ư	12
y	y	6
Tổng		**72**

Số 72 là **đếm ô trên lưới do mô hình chọn**; không chứng minh rằng 72 ô phủ toàn bộ âm tiết hay cấu trúc ngữ âm tiếng Việt. Một âm tiết chứa nhiều thông tin hơn như phụ âm đầu, cụm nguyên âm, phụ âm cuối và vị trí dấu. Chẳng hạn a và ba cùng có nguyên âm cơ sở và thanh điệu nhưng là hai âm tiết khác nhau. Do vậy không tồn tại song ánh $\Sigma \leftrightarrow \mathbb{Z}_{108}$ chỉ nhờ công thức trên đối với tập âm tiết Σ giàu hơn 108 phần tử. Trong ứng dụng hiện tại chúng tôi **không sử dụng E_G làm mã nén văn bản**.

3. Các mệnh đề đại số và chứng minh

Định lý 1 — CRT đúng với 108

Vì $108 = 4 \cdot 27$ và $\gcd(4, 27) = 1$, ánh xạ

$$\psi([n]_{108}) = ([n]_4, [n]_{27})$$

là **đẳng cấu vành** $\mathbb{Z}_{108} \cong \mathbb{Z}_4 \times \mathbb{Z}_{27}$ theo định lý thặng dư Trung Hoa. Đặt $q = n \bmod 4$, $r = n \bmod 27$. Vì $27 \equiv 3 \pmod 4$ và $3^{-1} \equiv 3 \pmod 4$, phép khôi phục tường minh là:

$$n = r + 27\,[3(q - r)]_4.$$

Hai thành phần q và r có miền đúng 4 và 27 giá trị. Mã hóa $q = b_0 + 2b_1$ bằng **2 bit**, $r = (t_0 + 1) + 3(t_1 + 1) + 9(t_2 + 1)$ bằng **3 trit cân bằng** $t_j \in \{-1, 0, +1\}$. Mỗi trạng thái được khôi phục duy nhất; kiểm thử thực hiện trên đủ 108 giá trị. Lưu ý ánh xạ từ nhóm $\mathbb{Z}_4$ sang cặp bit

chỉ là **song ánh tập hợp**, không phải đẳng cấu nhóm với $\mathbb{Z}_2 \times \mathbb{Z}_2$, vì hai nhóm có cấu trúc khác nhau. Tương tự, ba chữ số tam phân là biểu diễn giá trị, không phải phép cộng từng chữ số không nhớ.

Thuật ngữ chính xác là **2 bit + 3 trit**, không phải "5 trit". Nếu dùng đúng 5 trit thuần túy sẽ có $3^5 = 243$ cấu hình, không phải 108.

Định lý 2 — Phép nhúng vòng đơn vị

Với $\omega = e^{2\pi i/108}$, đặt $\phi(n) = \omega^n$. Ta có $|\phi(n)| = 1$, $\phi(n + 108) = \phi(n)$, và ϕ là đơn ánh trên $n \in [0, 107]$ với ảnh là nhóm 108 nghiệm của đơn vị $\mu_{108} \subset S^1$. Góc mỗi bước bằng $2\pi/108 = \pi/54$ radian hoặc $10/3$ độ.

Định lý 3 — Phép đối cực 180°

$$\phi(n + 54) = e^{2\pi i(n+54)/108} = \phi(n)e^{i\pi} = -\phi(n).$$

Mốc 27 bước cho 90°, 54 bước cho 180°, 81 bước cho 270°. Đây là hệ quả của việc 108 chia hết cho bốn, *không phải mệnh đề về đối nghĩa từ vựng*. Bất cứ nhóm cyclic bậc bội số của 4 cũng có các mốc tương tự.

Định lý 4 — Bất biến pha tương đối

Cho hai trạng thái a, b và cùng một phép dời k:

$$\frac{\phi(a + k)}{\phi(b + k)} = \omega^{a-b} = \frac{\phi(a)}{\phi(b)}.$$

Đẳng thức đúng hoàn toàn trong mô hình số phức lý tưởng. Đây là một trường hợp của phép xoay đẳng cự và giống **một tính chất hình thức** của RoPE. Nó không tương đương với RoPE của Su và cộng sự, vốn xoay nhiều cặp chiều theo nhiều tần số. Mô hình một vòng 108 bị alias khi độ dịch chênh nhau 108, nên không thể suy ra lợi ích ngữ cảnh dài.

4. Ánh xạ ký hiệu → trit → vector → tín hiệu

Đặt bảng ký hiệu chính xác $\mathcal{A} = \{\text{SPACE}, A, B, \dots, Z\}$ với chỉ số $0, 1, \dots, 26$. Một ký hiệu c có mã $j(c) \in [0, 26]$ và 3 trit cân bằng:

$$j = (t_0 + 1) + 3(t_1 + 1) + 9(t_2 + 1), \qquad t_i \in \{-1, 0, 1\}.$$

Đây là một song ánh: $27 = 3^3$. Nếu muốn một **vector tín hiệu hình học** cho ký hiệu, có thể định nghĩa $\mathbf{v}_j = (\cos(2\pi j/27), \sin(2\pi j/27))$, và phép lượng tử hóa góc ngược khi không nhiễu. Mã `symbolToVector` và `vectorToSymbol` hiện thực hóa phép song ánh này với kiểm tra chuẩn hóa/độ lệch vector; `npm test` duyệt đủ 27 ký hiệu và khôi phục chuỗi vector khi tọa độ không bị biến đổi. Tuy nhiên, trong bộ truyền thử nghiệm chúng tôi dùng **27-FSK**, tức mỗi ký hiệu ứng với một tần số:

$$s_j(t) = A \sin(2\pi(f_0 + j\Delta f)t), \quad 0 \le t < T_s.$$

Thông số đã cố định trong mã nguồn: $f_s = 24\,000$ Hz, $T_s = 0.100$ s, $f_0 = 600$ Hz, $\Delta f = 80$ Hz, $j = 0 \dots 26$. Bởi vậy các sóng mang nằm trong khoảng **600–2680 Hz** và có khoảng cách tần số $\Delta f T_s = 8$. Với các cửa sổ chữ nhật lý tưởng, tần số mang có số chu kỳ nguyên trên một ô thời

gian, tích vô hướng của hai sóng mang khác nhau bằng 0. Trong mã nguồn có taper 5 ms ở hai đầu ô, làm giảm tiếng click nhưng khiến tính trực giao chỉ còn **xấp xỉ**. Điều này phải phân biệt với định lý trực giao lý tưởng.

Có thể dùng đại lượng năng lượng tại từng ứng viên do Goertzel tính:

$$P_j = |\sum_{n=0}^{2399} x[n]\, e^{-2\pi i k_j n/2400}|^2, \quad k_j = (600 + 80j)\, T_s = 60 + 8j.$$

Chọn $\hat{j} = \arg\max_j P_j$ trên mỗi ô. Đây là bộ dò không đồng bộ pha; không sử dụng ASR, không tái tạo tiếng nói con người từ âm thanh, và không phải speech recognition. Với mẫu PCM do chính bộ phát sinh ra, đầu thu biết biên thời gian của ô đầu tiên và không có trôi đồng hồ.

4.1. Giao thức bản tin

Khung mã hóa theo giá trị Base27 (mỗi giá trị điều chế thành một âm):

```
[26,0,26,0]           : 4 ký hiệu magic/preamble
[length0,length1,length2]: 3 chữ số base27, little-endian
[payload]              : mã 0..26 của các ký tự A–Z / SPACE
[crc0,crc1,crc2,crc3] : CRC-16/CCITT-FALSE (poly 0x1021, init 0xffff)
```

Định nghĩa CRC áp dụng cho **chuỗi byte của chỉ số ký hiệu**, không tính preamble/độ dài. Một khung ký hiệu có $L + 11$ ô, với $L \leq 1800$, kéo dài $0.1(L + 11)$ giây. Giao thức chỉ chấp nhận các ký tự đúng bảng. Dấu câu, số, tab và chữ Unicode ngoài bảng bị từ chối thay vì cắt mất dữ liệu. Không có khả năng đồng bộ khung từ bản ghi microphone tùy ý; sóng WAV nhập cần được tạo theo đặc tả này, bắt đầu đúng tại mẫu 0, PCM16 mono 24 kHz. CRC chỉ phát hiện nhiều kiểu lỗi, không thể đảm bảo tuyệt đối không có va chạm.

Thông lượng thô là $\log_2(27)/0.1 \approx 47.55$ bit/s, chưa trừ overhead 11 ô/khung; không phải phép nén tối ưu và không cạnh tranh với modem hiệu suất cao.

5. CVNSS4.0 v5 audit-safe: giới hạn của tầng Unicode

Hệ thống sử dụng nguyên bản tệp `cvnss4_0_converter.pro.v5_0.audit_safe(20261116-003134).js` được người dùng cung cấp, được lưu trong `vendor/`. Hàm `fromCqn()` chuyển tiếng Việt đầu vào sang mã CVNSS; `fromCvss()` phục hồi theo chính sách **canonical** và `inspectWord()` cho biết các ứng viên cạnh tranh. Trong bộ tệp đính kèm, `selfTest()` trả về **23/23 đạt**, và `audit()` cho thấy **56 nhóm nhập nhằng**, **5 nhóm va chạm quan trọng**, bảng cơ sở 758 mục và 336 bản vá. Đây là kết quả tự kiểm thử của tệp v5 đã kiểm tra, không chứng nhận đầy đủ ngôn ngữ học.

Ta cần tách hai phép thử:

1. **Đảo ngược ký hiệu trên kênh:** `CVNSS_ASCII == decode(encode(CVNSS_ASCII))`. Kiểm thử có ý nghĩa khi bộ giải điều chế cùng bộ phát tuân thủ đặc tả và không có lỗi ký hiệu.
2. **Đảo ngược Unicode:** `Unicode_in == canonical_decode(CVNSS_encode(Unicode_in))`. Điều này **không bảo đảm** trên toàn bộ đầu vào vì bộ mã CVNSS có many-to-one mapping, có thể khác chữ hoa/thường hoặc các biến thể nhập nhằng. Không một phép dựng vòng số nào tự nó sửa được vấn đề này.

Để bảo toàn 100% mọi văn bản UTF-8, cần thêm một **kênh escape/metadata** hoặc truyền payload UTF-8 trực tiếp qua mã dữ liệu có định dạng và checksum; không thể chỉ dùng 108 trạng thái/âm tiết như một phép nén duy nhất.

6. Thiết kế thí nghiệm và tính tái lập

Mã kiểm thử dùng Node.js built-in `node:test` và không cần cài package bên ngoài. Bộ thí nghiệm mô phỏng thực hiện 24 khung độc lập cho mỗi mức SNR từ −35 đến 20 dB, với 4 chuỗi ASCII thử luân phiên và bộ tạo nhiễu giả ngẫu nhiên xorshift có seed cố định. Các sóng được điều chế thành Float32 rồi cộng AWGN; bên thu đo **SER**, số khung toàn vẹn và số khung bị từ chối bởi giao thức/CRC. Đây là **tỉ lệ thành công mẫu**, không phải cận xác suất lỗi hay bảo đảm định lý.

Cấu hình tái lập:

```
node --version
npm test
node experiments/benchmark.js
```

Lệnh cuối ghi dữ liệu chi tiết tại `experiments/results.json`. Ví dụ nguồn phát `Tiếng Việt` được converter ánh xạ thành `Tizb Vidf`; sau chuẩn hóa ASCII để truyền là `TIZB VIDF`.

6.1. Kết quả số của bản chạy ngày 16/11/2026

SNR (dB)	Khung đúng / 24	Tỷ lệ thành công	Lỗi ký hiệu / 522	SER
−35	0	0%	481	0.9215
−30	0	0%	421	0.8065
−25	0	0%	235	0.4502
−20	13	54.17%	14	0.02682
−15	24	100%	0	0
−10	24	100%	0	0
0	24	100%	0	0
20	24	100%	0	0

Bảng có tổng cộng **192 khung** (8 SNR × 24). SNR âm là tương quan công suất tổng tín hiệu với AWGN trong miền mẫu. Mức −15 dB đạt 24/24 ở seed và payload đã chọn nhưng không đại diện cho mọi khả năng nhiễu, lệch đồng hồ, méo loa hay tiếng vang. Đặc biệt kết quả 54.17% tại −20 dB phản bác mệnh đề "hoàn hảo 100% trong mọi điều kiện". Ngưỡng đầu thu, bảng ký hiệu, payload và mã mô phỏng phải được giữ cố định khi đối chứng.

6.2. Ý nghĩa của số liệu

Kết quả 100% của một **tập thử hữu hạn** chỉ cho thấy số lỗi bằng 0 trong tập đó, không suy ra tỷ lệ lỗi kỳ vọng bằng 0. Muốn đánh giá độ tin cậy cần nhiều hạt giống, nhiều payload có phân bố khác nhau, khoảng tin cậy và mô hình truyền kênh thực (môi trường phòng, biến thiên tần số lấy mẫu, AGC, nén âm thanh, clipping, Doppler, đồng bộ preamble).

7. Hạn chế, đối chứng và tính mới

- Vành $\mathbb{Z}_{108}$ là hình thức hóa của một **lưới do người nghiên cứu thiết kế**, không phải chứng minh tiếng Việt có 108 âm tiết nguyên thủy hay một cấu trúc nhóm ngữ nghĩa tự nhiên.
- Số 72 ô chỉ đếm sự tương thích dấu phụ của sáu **ký tự nguyên âm cơ sở**, không phải tập các vần hợp lệ, tần suất xuất hiện hay tối ưu mã hóa.
- Việc gắn góc với một ô chỉ là một phép nhúng theo nhóm cyclic; các cơ số chẵn khác cũng có trục 180° và mọi bội số của bốn có các mốc 90°.
- 108 không "tối ưu" cho tất cả mục tiêu. Hệ 27-FSK là thiết kế trình diễn dễ kiểm toán, tốn băng thông, chậm và phụ thuộc đồng bộ.
- 2 bit + 3 trit không có nghĩa mạch phần cứng tam phân thực sự cần 6.755 bit bộ nhớ: phải tính biểu diễn vật lý, bố trí bộ nhớ, giao thức và chi phí toán học.

- Không có kết quả huấn luyện language model, so sánh perplexity, ESP32-S3/FPGA, thực nghiệm sóng loa–microphone, hay chứng minh chất lượng ASR/semantics.
- Trước khi công bố như một preprint học thuật chính thức, cần xác minh độc lập provenance của converter và đưa vào test golden corpus đủ lớn, đánh giá các nhóm nhập nhằng, đối chứng các hệ mã số khác.

Điểm mới trong phạm vi **bản triển khai này** là sự ghép nối có kiểm chứng giữa mô hình vòng minh họa, chuẩn dữ liệu 27 ký hiệu/trit, truyền thông sóng FSK có khung CRC, và bộ converter audit-safe. Không khẳng định các thành phần toán học hay FSK đã được phát minh mới.

8. Kết luận

Các mệnh đề $\mathbb{Z}_{108} \cong \mathbb{Z}_4 \times \mathbb{Z}_{27}$, $\phi(n+54) = -\phi(n)$ và bất biến pha dưới phép dời đồng thời là các kết quả toán học chặt chẽ. Bộ phát 27-FSK và đầu thu Goertzel chứng minh mang tính **xây dựng** rằng có thể biến chuỗi ký hiệu CVNSS ASCII thành sóng PCM rồi tái tạo qua cùng một giao thức khi thỏa các giả thiết xác định. Dữ liệu nhiễu cho thấy hiệu suất thực nghiệm thay đổi theo SNR, nên khái niệm "khả thi 100%" phải được định nghĩa theo **miền thử và điều kiện kênh**, không phát biểu như bảo đảm phổ quát. Tầng Unicode và ngữ nghĩa cần được kiểm tra riêng. Bản nghiên cứu cung cấp mã, kiểm thử và số liệu để cộng đồng tái lập hoặc bác bỏ từng giả thuyết.

9. Tài liệu tham khảo

1. Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., Liu, Y. (2021, updated 2023). *RoFormer: Enhanced Transformer with Rotary Position Embedding*. arXiv:2104.09864. https://arxiv.org/abs/2104.09864
2. Proakis, J. G. & Salehi, M. *Digital Communications*, 5th ed. McGraw-Hill. Các chương về M-ary orthogonal FSK và noncoherent detection.
3. Chinese Remainder Theorem. Kết quả kinh điển về đẳng cấu vành modulo các mô-đun nguyên tố cùng nhau.
4. Goertzel, G. (1958). *An Algorithm for the Evaluation of Finite Trigonometric Series*. The American Mathematical Monthly, 65(1), 34–35.
5. CVNSS4.0 converter v5.0.0-audit-safe, tệp tác giả/người dùng cung cấp ngày 16/11/2026; bản lưu trong `vendor/` của dự án, với public API `audit()`, `inspectWord()`, `selfTest()`.

Phụ lục A — kiểm thử điều kiện phản bác

- **A1**: Thuật toán phải phát hiện lỗi dữ liệu khi CRC sai; test kiểm tra sửa một ký hiệu payload trong khung.
- **A2**: Đầu vào `VIỆT` hoặc `ABC.` không được âm thầm truyền thành chuỗi khác; test khẳng định hệ thống báo lỗi cho ký tự ngoài bảng.
- **A3**: Tất cả 108 mã phải khôi phục qua CRT và cặp `2 bits + 3 balanced trits`.
- **A4**: Mọi 27 ký hiệu gồm khoảng trắng phải khôi phục qua đúng 3 trit.
- **A5**: Mô phỏng AWGN có seed cố định cho phép kiểm chứng lại tỷ lệ thành công; không suy ra xác suất lỗi tổng quát.

Phụ lục B — chỉ dẫn phát hành

Đặt repository ở nhánh `main`, bật **GitHub Pages** từ `/ (root)` để hiển thị `index.html`. Chạy local bằng `python -m http.server 8000`; truy cập `http://localhost:8000`. Công bố preprint trên kho lưu trữ học thuật chỉ sau khi rà soát phản biện kỹ thuật, license và dữ liệu benchmark.