DOI: 10.35598/mcfpga.2026.026

IMPLEMENTATION OF A HARDWARE ACOUSTIC MASKING SYSTEM FOR SPEECH INFORMATION BASED ON FPGA

Oleksandr Vorgul
ORCID 0000-0002-7659-8796
Department of the Media Engineering and Information Radio Electronic Systems,
Kharkiv National University of Radioelectronics
Kharkiv, Ukraine
oleksandr.vorgul@nure.ua

Oleksii Bilotserkivets
ORCID 0000-0002-8514-9650
Department of the Media Engineering and Information Radio Electronic Systems,
Kharkiv National University of Radioelectronics
Kharkiv, Ukraine
oleksii.bilotserkivets@nure.ua

***Abstract* — This paper presents an FPGA-based architecture for dynamic acoustic masking that generates speech-like (babble) interference in real time to protect confidential speech from acoustic leakage. The system combines dynamic time-scrambling with spectral frequency inversion, implemented as a fully pipelined DSP core. A key optimization eliminates all DSP48E1 multipliers by reducing the inversion operation to a single XOR gate, achieving end-to-end latency of approximately 125 μs and logic utilization below 1% of the Artix-7 LUT budget. These figures make successful cancellation by adaptive filters practically impossible, supporting the development of mobile, high-assurance jamming platforms for secure environments.**

***Keywords* — emission security, acoustic jamming, babble noise, FPGA, speech scrambling, frequency inversion, side-channel protection..**

I. INTRODUCTION

The protection of language information in the premises for confidential negotiations is one of the key tasks of technical information protection (TII). The biggest threat is remote and contact methods of removing acoustic signals through enclosing structures (windows, walls, engineering communications) using laser, vibroacoustic and stethoscope means of interception. Traditional white or pink noise generators create a significant acoustic load on a person and are easily subject to adaptive digital filtering by modern machine learning methods.

The most resistant to defiltering is speech-like interference (the so-called "babel noise"), which has identical time-frequency and timbral characteristics with the useful signal. The implementation of such systems based on classic microcontrollers or universal DSPs is limited by strict requirements regarding the processing delay time and the need for multi-channel parallel interference synthesis. The work is devoted to the research and development of the architecture of an active acoustic masking device based on modern FPGAs, which ensures minimal hardware delay and a high level of flexibility of speech destruction algorithms [1].

II. MATHEMATICAL MODEL OF SPEECH SIGNAL DESTRUCTION

The basis of the proposed protection method is the transformation of the input speech signal $S(t)$ into an unreadable acoustic mixture by means of dynamic time scrambling and frequency inversion. The input digitized signal is divided into a sequence of time windows (segments) with a duration of $\Delta\tau$=10...30 ms, which corresponds to the average duration of speech by the microphone. Mathematically, the process of forming masking interference $N_k(t)$ for a k-channel system of vibroacoustic emitters is described as a superposition of destructured copies of the original signal with random transmission coefficients and time shifts:

$$N_k(t) = \sum_{i=1}^{M} \alpha_{k,i}(t) \cdot S_{inv}\left(t - \tau_{k,i}(t)\right) \quad (1)$$

where $S_{inv}(t)$ is the frequency-inverted speech signal, $\alpha_{k,i}(t)$is the random amplitude weighting coefficient of the i-th ray for the k-th channel, $\tau_{k,i}(t)$is the dynamic time delay governed by a pseudo-random sequence (PRS), and $M$is the number of synthesized virtual speech sources (interfering voices).

Frequency inversion is performed by multiplying the input signal with a reference harmonic oscillation at the cutoff frequency of the speech band ($F_0 = 3.4$ kHz), followed by digital low-pass filtering:

$$S_{inv}(t) = [S(t) \cdot \cos(2\pi F_0 t)] * H_{LPF}(t) \quad (2)$$

where $H_{LPF}(t)$is the impulse response of the digital low-pass filter. As a result of this transformation, the high-frequency components of the speech spectrum (formants) become low-frequency, and the low-frequency ones become high-frequency. The resulting signal completely loses informative intelligibility, yet ideally preserves the integral energy spectrum of the speaker's voice. [2]

III. HARDWARE ARCHITECTURE AND FPGA PROCESSING SCHEME

To keep the emission-to-jamming delay within $\Delta t \leq 2$ ms, we deploy the entire system directly into the FPGA fabric as a fully pipelined custom DSP core. The processing pipeline includes the following blocks:

- ADC / I2S Interface — receives a serial digital data stream from an external audio codec at a sampling rate of $f_s = 16$kHz (16-bit resolution).
- Digital inversion modulator — performs sample-by-sample multiplication of the samples by an

alternating sign sequence, which shifts and mirrors the frequency components of the sound.

- FIFO Buffer (Block RAM) — an array of built-in dual-port FPGA memory organized as a ring delay line for the dynamic storage of temporary windows of the speech signal [3].
- PRNG — a pseudo-random number generator based on a linear feedback shift register, which generates random memory read control addresses and window mixing indices.
- Time mixing multiplexer — performs a chaotic change in the sequence of issuing sound blocks from memory, destroying the time correlation of speech.
- Digital mixer — combines several time-shifted and inverted streams, creating the effect of a multi-voiced hum (Babel-noise).

IV. OPTIMIZATION OF THE DIGITAL INVERSION MODULATOR FOR THE FPGA PIPELINE ARCHITECTURE

To achieve strong mathematical resistance of the acoustic jamming signal, the semantic structure of the speaker's speech is broken down inside a dedicated FPGA block. In our design, this task falls on the spectral frequency inversion module, which mirrors the frequency components around the central axis of the speech band. In a conventional digital signal processing (DSP) system, frequency shifting and inversion are implemented by multiplying the input discrete signal $S(n)$by a harmonic reference oscillation at the cutoff frequency F_0:

$$S_{inv}(n) = S(n) \cdot \cos\left(\frac{2\pi F_0 n}{f_s}\right) \qquad 3$$

where f_s is the sampling frequency of the audio stream (16 kHz). Direct evaluation of the cosine and floating-point multiplication would consume substantial FPGA fabric — specifically DSP48E1 slices — and add pipeline latency to the data path. To reduce the hardware cost, we propose a method that locks the reference frequency $F_0$to the codec sampling rate [4]. By placing the inversion point exactly at $F_0 = f_s/4 =$ 4 kHz — the mathematical and physical midpoint of the speech band — the discrete cosine samples for successive clock cycles $n = 0,1,2,3, \ldots$ reduce to a simple repeating sequence:

$$\cos\left(\frac{2\pi \cdot (f_s/4) \cdot n}{f_s}\right) = \cos\left(\frac{\pi n}{2}\right) = \{+1,\ 0,\ -1,\ 0,\ +1,\ 0 \ldots\} \qquad (4)$$

An alternative approach that shifts the entire spectrum by half the sampling rate reduces the modulator's hardware function to a pure sign-alternating sequence:

$$(-1)^n = \{+1,\ -1,\ +1,\ -1 \ldots\} \qquad 5$$

This algebraic reduction fundamentally reshapes the computational core inside the FPGA. Instead of deploying a bulky hardware multiplier, the digital inversion modulator collapses into a single sign-inverter — a purely combinatorial multiplexer driven by the least significant bit (LSB) of the sample counter n. The operation, which in a classical DSP would occupy a dedicated DSP48E1 slice and consume several pipeline stages, is now reduced to two trivial steps:

- For **even samples** ($n = 2k$), the signed input word $S(n)$passes through unchanged.
- For **odd samples** ($n = 2k + 1$), the sign bit is flipped, which is mathematically equivalent to an instantaneous multiplication by -1.

At the register-transfer level (RTL), this is implemented as a simple bitwise XOR with a mask derived directly from the sample counter. Because the operation resolves without requiring a carry-propagation adder, the critical path of the modulator shrinks to just a single XOR gate. [5]

The quantitative impact is substantial. The inversion block no longer consumes any of the 240 DSP48E1 slices in the Artix-7 target, leaving them free for other tasks. Logic utilization drops below 1% of the total LUT budget, and crucially, the modulator introduces **zero additional pipeline stages**. The XOR operation resolves within a single system clock, keeping the maximum achievable frequency f_{max} completely unaffected — a decisive advantage when every nanosecond of latency matters for the sub-2 ms protection window.

Ultimately, this structural simplicity gives the FPGA an indisputable edge over classical DSPs and microcontrollers. A general-purpose DSP would still need multiple clock cycles to fetch coefficients and execute a multiply-accumulate sequence, introducing deterministic latency that is nearly impossible to hide inside such a tight protection window. The FPGA, by contrast, turns the inversion into a wire-level transformation that rides alongside the data stream without ever slowing it down.

CONCLUSION

The proposed FPGA-based architecture for dynamic acoustic masking offers a practical path toward high-assurance emission security. Analytical estimates and preliminary block-level simulation both point to strong masking performance, enabled by full computational parallelism and native use of Block RAM. End-to-end latency settles around 125 μs, a figure that, according to our analysis, places successful cancellation by adaptive filters out of reach. Locking the modulator to the $f_s/4$ frequency point eliminates the need for hardware multipliers and keeps the logic footprint minimal. The design-stage results support moving forward with a mobile, high-security jamming platform built on this architecture.

REFERENCES

[1] X. Ge, G. Sun, B. Zheng, and R. Nan, "FPGA-Based Voice Encryption Equipment under the Analog Voice Communication Channel," Information, vol. 12, no. 11, p. 456, 2021, doi: 10.3390/info12110456.

[2] A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2009, 1108 p.

[3] U. Meyer-Baese, Digital Signal Processing with Field Programmable Gate Arrays, 4th ed. Berlin, Heidelberg: Springer, 2014, 793 p., doi: 10.1007/978-3-642-45309-0.

[4] 7 Series DSP48E1 Slice: User Guide (UG479), Xilinx Inc., San Jose, CA, USA, 2018, 74 p.

[5] O. Vorgul, "Approaches Half Band Filter Realization for Means FPGA," in Proceedings of the International Conference on Microcontrollers and FPGAs, Kharkiv, Ukraine: KNURE, 2019, pp. 44–47, doi: 10.35598/mcfpga.2019.015.