

Aquisição e Processamento de Áudio PDM em FPGA

Julio Nunes Avelar ¹

Departamento de Sistemas de Computação, Universidade Estadual de Campinas

14 de Abril de 2025

¹Endereço de e-mail: `julio.avelar@students.ic.unicamp.br`

0.1 Introdução

A aquisição de sinais de áudio em plataformas digitais, como FPGAs, microcontroladores e sistemas embarcados baseados em SBCs (Single Board Computers), demanda uma escolha criteriosa do tipo de transdutor de entrada. Dentre as principais tecnologias disponíveis para captura de áudio estão os microfones analógicos — que requerem um conversor analógico-digital (ADC) externo para digitalização do sinal —, os microfones digitais com interface I2S (Inter-IC Sound), que transmitem amostras PCM já digitalizadas via um barramento serial síncrono, e os microfones baseados em modulação PDM (Pulse Density Modulation), que emitem um fluxo binário com densidade proporcional à amplitude do sinal analógico original.

Os microfones PDM apresentam vantagens relevantes para integração com FPGAs, especialmente em aplicações de baixo custo e recursos limitados. Seu sinal digital direto elimina a necessidade de circuitos ADC dedicados, e a interface simplificada — geralmente composta por apenas dois sinais (clock e dados) — facilita o roteamento em projetos de hardware. Contudo, a natureza do sinal PDM impõe a necessidade de um processamento adicional para reconstrução do sinal de áudio em domínio de tempo contínuo, exigindo filtragem digital (tipicamente um filtro passa-baixas FIR ou CIC) e subsequente redução da taxa de amostragem (downsampling) para obter um fluxo PCM utilizável.

Este trabalho descreve o desenvolvimento de um conjunto de módulos em SystemVerilog implementados em FPGA, responsáveis pela aquisição do sinal PDM, filtragem digital e redução da taxa de amostragem. O projeto é otimizado para a captura de sinais de baixa frequência, com ênfase em faixas abaixo de 2 kHz, adequadas à reprodução da voz humana e aplicações de reconhecimento de fala, monitoramento acústico e sistemas interativos baseados em comandos vocais.

0.2 Microfones PDM

Em aplicações embarcadas modernas, como *smartphones*, relógios inteligentes e fones de ouvido, é comum a utilização de microfones do tipo MEMS (*Micro-Electro-Mechanical Systems*). Esses dispositivos integram, em uma escala de milímetros, um sensor mecânico — como um pequeno diafragma que vibra com a pressão sonora — e circuitos eletrônicos responsáveis pela conversão do sinal acústico em sinal digital.

Os microfones MEMS são amplamente utilizados por diversos motivos: custo reduzido, dimensões compactas, qualidade satisfatória, facilidade de montagem (podem ser soldados diretamente em placas PCB) e simplicidade do circuito receptor, já que fornecem diretamente dados digitais.

Entre os principais tipos de microfones MEMS, destacam-se:

- **I²S (Inter-IC Sound)** — envia diretamente um fluxo serial de amostras PCM (Pulse-Code Modulation), contendo valores quantizados com múltiplos bits por amostra.
- **PDM (Pulse Density Modulation)** — transmite apenas um bit por ciclo de clock, representando o sinal através da densidade de pulsos ao longo do tempo.

O PDM codifica a informação de amplitude do sinal analógico na densidade de bits '1' em relação aos bits '0'. Assim, uma área com maior densidade de '1's representa uma região de maior amplitude do sinal original, enquanto uma maior presença de '0's representa amplitudes mais baixas. A figura a seguir ilustra esse princípio:

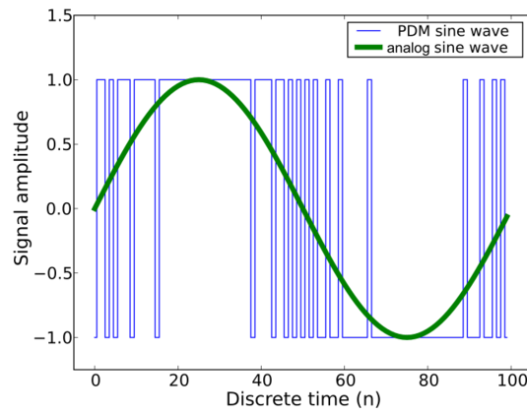


Figura 1: Modulação PDM: densidade de '1's representa a amplitude do sinal. Disponível em: https://handwiki.org/wiki/Pulse-density_modulation

Esse tipo de modulação pode ser visto como uma forma extrema de quantização: ao invés de quantizar o sinal analógico em múltiplos níveis com vários bits por amostra (como no PCM), o PDM quantiza cada ponto no tempo com apenas 1 bit, mas compensando com uma taxa de amostragem muito mais alta. Em vez de depender da resolução da amostra, como ocorre no PCM, o PDM depende da frequência do clock: quanto maior o clock, mais fiel será a representação do sinal analógico através da densidade de bits.

A interface de microfones PDM costuma ser bastante simples, exigindo apenas dois sinais:

- **Clock (PDM_CLK)** — sinal de clock fornecido externamente ao microfone, geralmente com frequências de alguns MHz (tipicamente entre 1 e 3 MHz).
- **Data (PDM_DATA)** — fluxo de bits representando o sinal de áudio em modulação por densidade de pulsos.

Além disso, alguns microfones PDM suportam modos estéreo utilizando multiplexação: dois microfones podem compartilhar a mesma linha de dados, sendo alternadamente habilitados em bordas opostas do *clock*.

0.3 Modulação delta sigma

Tem que escrever e parece um puta trampo

0.4 Filtros

0.4.1 Filtro CIC

O primeiro filtro utilizado é um filtro CIC (Cascaded Integrator-Comb), um tipo de filtro digital baseado apenas em operações de soma e subtração, ideal para redução da taxa de amostragem (*down-sampling*) em sistemas digitais como FPGAs, por sua eficiência e simplicidade de implementação.

O filtro é composto por dois blocos principais:

- **Estágios de integração (integrators):** acumuladores que somam continuamente os valores de entrada.
- **Estágios de diferenciação (combs):** calculam a diferença entre a saída atual e uma saída passada, removendo a tendência acumulada.

Um dado de entrada é lido a cada borda de subida do clock do microfone PDM. A cada `DECIMATION_FACTOR` ciclos, uma nova saída é gerada, reduzindo a taxa de amostragem em um fator igual a esse parâmetro.

O número de estágios de integração e diferenciação é definido pelo parâmetro R , que representa a ordem do filtro. Valores maiores de R aumentam a atenuação de componentes de alta frequência, mas também elevam o ganho e a largura de banda do filtro.

Estágios do Integrador

Dado $x[n]$, o valor recebido do microfone no instante n , temos o seguinte mapeamento:

$$x[n] = \begin{cases} +1, & \text{se } \text{pdm_data} = 1 \\ -1, & \text{se } \text{pdm_data} = 0 \end{cases}$$

Cada estágio do integrador realiza uma soma acumulada:

$$\begin{aligned} y_0[n] &= y_0[n-1] + x[n] \\ y_k[n] &= y_k[n-1] + y_{k-1}[n], \quad \text{para } 1 \leq k < R \end{aligned}$$

Estágios do Comb

Após a decimação, o sinal passa pelos estágios do tipo comb, que aplicam um atraso M (tipicamente igual a 1) e realizam a operação de subtração:

$$\begin{aligned} c_0[n] &= y_{R-1}[n] - y_{R-1}[n-M] \\ c_k[n] &= c_{k-1}[n] - c_{k-1}[n-M], \quad \text{para } 1 \leq k < R \end{aligned}$$

Esses estágios removem a inclinação acumulada pelos integradores e recuperam a forma original do sinal.

Ganho do Filtro

O filtro CIC possui um ganho total definido por:

$$G = (D \cdot M)^R$$

Onde:

- D é o fator de decimação (`DECIMATION_FACTOR`)
- M é o atraso dos estágios do tipo comb
- R é o número de estágios

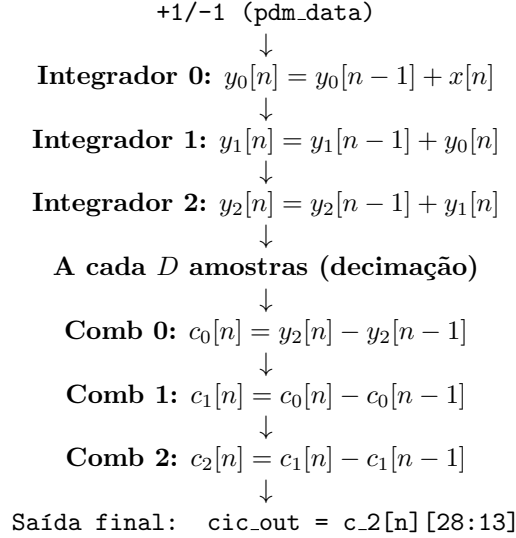
Esse ganho precisa ser considerado para evitar estouros de registradores internos e também pode ser compensado posteriormente com uma operação de divisão, deslocamento à direita ou normalização.

A saída final do filtro é obtida a partir dos 16 bits centrais do último registrador comb, compensando parcialmente o ganho:

$$\text{cic_out} = \text{comb}[R-1][28:13]$$

Fluxograma do filtro CIC (com 3 estágios)

O fluxograma abaixo representa a estrutura de um filtro CIC com $R = 3$ estágios (3 integradores seguidos por 3 combs), e fator de decimação D . O atraso M nos estágios comb é igual a 1.



Ao final dessa cadeia, o sinal encontra-se suavizado e com a taxa de amostragem reduzida, pronto para a posterior filtragem FIR e uso como PCM.

0.4.2 Filtro FIR de pesos iguais (média móvel)

Após o filtro CIC, é aplicado um filtro FIR (Finite Impulse Response) com coeficientes unitários, também conhecido como filtro de média móvel. Esse filtro realiza uma suavização adicional do sinal, atenuando ruídos residuais de alta frequência e melhorando a qualidade perceptível do áudio.

O comportamento do filtro pode ser descrito pela equação:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k]$$

onde $x[k]$ representa a k -ésima amostra saída do filtro CIC, e N é o número de taps (coeficientes), todos com valor igual a 1.

A implementação utiliza um buffer circular que armazena as últimas N amostras. A cada nova entrada, a mais antiga é descartada, e o somatório é atualizado. Como N é uma potência de dois, a média é obtida com um simples deslocamento à direita de $\log_2(N)$ bits, o que reduz significativamente o custo de hardware.

Uma possível melhoria para o filtro FIR é a substituição dos coeficientes unitários por coeficientes pré-calculados, definidos de acordo com uma janela de ponderação (por exemplo, Hamming ou Blackman). Essa abordagem permite atribuir pesos diferentes a cada amostra na média, oferecendo uma resposta em frequência mais controlada e melhor rejeição de ruídos fora da banda de interesse.

0.4.3 Média simples para redução de amostras

Após a saída do filtro FIR, realiza-se uma média simples sobre blocos de N amostras consecutivas. Essas amostras são acumuladas e, em seguida, divididas por N , onde N é escolhido como uma potência de dois para permitir a substituição da divisão por uma operação de deslocamento à direita, otimizando o uso de recursos em hardware.

Esse processo reduz ainda mais a taxa de amostragem, com impacto mínimo na qualidade perceptual do sinal, atuando como uma etapa adicional de suavização e estabilização dos dados.

0.4.4 Downsampling e diminuição de resolução

O *downsampling* ocorre em três etapas distintas ao longo da cadeia de filtragem:

1. **No filtro CIC**, que reduz a taxa de amostragem pela razão de decimação R , correspondente ao valor de `DECIMATION_FACTOR`.
2. **No filtro FIR**, de forma implícita, ao gerar uma nova amostra suavizada com base em múltiplas amostras CIC anteriores — o que pode ser interpretado como uma redução efetiva na taxa, ainda que o clock se mantenha.
3. **Na média final**, onde N amostras são acumuladas e divididas por N , promovendo uma nova etapa de redução da taxa de amostragem por N .

Além da redução de taxa, há uma limitação intencional da resolução ao longo do processamento. A saída do CIC possui 32 bits, mas apenas os bits centrais (por exemplo, [28 : 13]) são utilizados, descartando as extremidades para evitar saturação e reter a parte mais significativa da resposta. Em seguida, o filtro FIR gera um valor final que é truncado para 16 bits, produzindo a amostra PCM.

Na última etapa, esse valor de 16 bits com sinal (intervalo $[-32768, 32767]$) é convertido para um formato sem sinal de 8 bits. Isso é feito ao ajustar o valor para o intervalo $[0, 65535]$ (removendo o sinal com offset de $+32768$) e dividindo por 256 (2^8), resultando em uma amostra final entre 0 e 255.

0.4.5 Parâmetros utilizados

Abaixo estão os principais parâmetros empregados no projeto:

- **DECIMATION_FACTOR**: 256 — número de ciclos de clock PDM entre cada amostra produzida pelo CIC.
- **DATA_WIDTH**: 16 — largura final da amostra PCM antes da conversão para 8 bits.
- **FIR_TAPS**: 64 — número de coeficientes no filtro FIR de média móvel.
- **CLK_FREQ**: 100 MHz — frequência do clock do sistema.
- **PDM_CLK_FREQ**: 1.800 MHz — frequência do sinal de entrada do microfone PDM.
- **CIC_STAGES**: 4 — número de estágios do filtro CIC (R).
- **SECOND_DECIMATION_FACTOR**: 2 - número N utilizado na media simples após o filtro FIR.

Tais parâmetros foram definidos de forma experimental, por meio da avaliação de diferentes combinações e análise do sinal resultante. Foram utilizados dois métodos principais: inspeção visual das formas de onda, a fim de observar distorções ou artefatos indesejados, e testes de escuta, para verificar a qualidade perceptiva do áudio resultante. O objetivo foi encontrar um equilíbrio entre fidelidade sonora, simplicidade de implementação em hardware e uso eficiente dos recursos disponíveis.

Redução cumulativa da taxa de amostragem

A seguir, é apresentada uma estimativa da taxa de amostragem em cada etapa do processamento, assumindo que o clock PDM opera a 1.800 MHz:

Etapas	Fator de Redução	Taxa Resultante (Hz)
Sinal PDM	$1\times$	1.800.000
Filtro CIC	$256\times$	7.031,25
Média Final (8-bit)	$2\times$	3.515,62
Total acumulado	$512\times$	3.515,62 Hz (estimado)

Tabela 1: Redução acumulada da taxa de amostragem ao longo do processamento

Nota: A taxa final pode ser ajustada conforme o intervalo de leitura entre amostras do FIR. O valor apresentado é uma estimativa teórica, considerando que cada saída do FIR passa por uma nova média de $N = 2$ amostras antes de formar a saída final de 8 bits.

Essa cadeia permite transformar um sinal PDM com alta taxa e baixa resolução em uma sequência de amostras PCM com taxa mais baixa e qualidade suficiente para representar sinais de baixa frequência, como voz humana, onde $F < 2kHz$.

0.5 Implementação

A leitura e o processamento do sinal do microfone PDM foram inteiramente implementados em *SystemVerilog*. O projeto é dividido em dois módulos principais:

- **pdm_capture** — responsável pela captura dos dados PDM e aplicação dos filtros CIC e FIR.
- **down_sample_and_resolution** — realiza a média simples sobre as amostras do FIR e converte o sinal resultante para 8 bits.

O código-fonte completo de ambos os módulos está disponível no repositório do projeto no GitHub: https://github.com/Unicamp-Odhin/PDM_Microphone. O repositório também inclui os demais arquivos e módulos auxiliares necessários para a transmissão de áudio, análise do sinal, conversão para o formato WAV, entre outras utilidades.

Além disso, o repositório contém amostras de áudio reais coletadas durante os testes, permitindo validar a qualidade do sistema e servir como referência para futuros desenvolvimentos.

0.6 Resultados