Energy-efficient short-time Fourier transform for partial window overlapping

Eleftheriadis, C., Garrido, M., & Karakonstantis, G. (2023). Energy-efficient short-time Fourier transform for partial window overlapping. In *Proceedings of the IEEE International Symposium on Circuits and Systems 2023* (IEEE International Symposium on Circuits and Systems: Proceedings). Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/ISCAS46773.2023.10181940

Published in:
Proceedings of the IEEE International Symposium on Circuits and Systems 2023

Document Version:
Peer reviewed version

Energy-Efficient Short-Time Fourier Transform for Partial Window Overlapping

Charalampos Eleftheriadis[1], Mario Garrido[2], Georgios Karakonstantis[1]
[1]Institute of Electronics, Communications and Information Technology (ECIT), Queen's University Belfast.
[2]Department of Electrical Engineering, ETSI de Telecomunicación, Universidad Politécnica de Madrid.
Email: C.Eleftheriadis@qub.ac.uk, mario.garrido@upm.es, G.Karakonstantis@qub.ac.uk

***Abstract*—This paper presents an energy-efficient short-time Fourier transform (STFT) architecture. The proposed architecture is called frequency decomposition STFT (FD-STFT) and it achieves significant computational complexity reduction by effectively re-utilizing previously computed spectrums between overlapped sampling windows. Such an algorithmic modification not only reduces the required hardware units, but also achieves low accumulative error compared to conventional approaches. In addition, the quality of the resulting spectrogram is improved by integrating an efficient Hanning windowing technique that replaces the multiplication in the time domain with a low-cost filtering in the frequency domain. For an $N = 256$-point window with $R = 32$ overlapping samples, our results indicate that our approach achieves up-to 40.86% and 65.56% area and power savings respectively, compared to recent approaches.**

***Index Terms*—Short time Fourier transform (STFT), fast Fourier transform (FFT), pipelined architecture.**

I. Introduction

Over the last few years, power spectral analysis (PSA) has been utilized in various audio [1], biomedical [2, 3] and communications applications [4] that are increasingly executed on energy-constrained devices. The most insightful PSA approach is the so-called time-frequency analysis, which is widely performed by the short-time Fourier transform (STFT) [5] offering a clear trade off between spectral and temporal resolution. The STFT may provide precious insights about the evolution of a signal over time, but this information comes at a high computational and power cost, since the signal's spectrum must be estimated at multiple time instants. This reality turned recently the attention to the design of low-complexity STFTs [6–8] for limiting the consumed power.

For this reason, some recent proposals aimed at addressing the high computational complexity of the conventional STFT [6], which evaluates consecutive sliding sampling windows, by executing in parallel several pipelined FFTs. The iterative architecture [7] reduced the complexity by decimating the required frequencies at independent channels and computing them repetitively for each time sample. The main drawbacks of this method is the increased accumulative error induced by the consecutive quantizations and the restriction of the window hop size to 1, which leads to redundant time sampling at the output spectrogram, and thus to excessive computational overhead. In addition, the feedforward STFT architecture [8] reutilized operations between consecutive FFTs that have only one time sample difference. This reduced the complexity, and the accumulative error, but the application to hop sizes different that one is still unresolved.

Apart from only considering hop sizes equal to one, another limitation of [7, 8] is that they only support rectangular window functions, which limits the quality of the output spectrum due to introduced noise by side lobe leakage [5]. To this end, we have developed a novel STFT architecture, called frequency decomposition short-time Fourier transform (FD-STFT). The contributions of this paper can be summarized as follows:

1) Decimate the STFT spectrum, for several overlapping window ratios, into two terms called the base and the correction spectrum, which are generated by the overlapping windowed data and the difference between the non-overlapping data, accordingly. This allow us to re-utilize the spectrum that is calculated in prior windows.
2) Propose the FD-STFT architecture, which targets hop sizes larger that one and requires compared with recent works, the least amount of multipliers and adders while also employing a reduced number of memory cells. Also, our approach reduces the accumulative error induced by the consecutive corrective operations between sampling windows, with respect to the iterative approach.
3) Integrate within the proposed FD-STFT, a Hanning window function as a filtering operation in the frequency domain instead of a multiplication in the time domain. This modification allows us to depart away from noise prone rectangular windows and circumvent the relevant power-hungry multiplications.
4) Implement and demonstrate the efficacy of the proposed FD-STFT architecture in terms of hardware complexity, area and power savings compared to prior approaches. Our results in $45\,\mathrm{nm}$ process technology show that our FD-STFT architecture, when compared with other schemes [6–8], achieves up-to 40.86% and 65.56% area and power savings, respectively.

The rest of the paper is organized as follows. Section II discusses the background and challenges existing STFT methods. Section III presents our proposed FD-STFT. Section IV compares our work with previous ones in terms of power savings and hardware units. Section V draws the final conclusions.

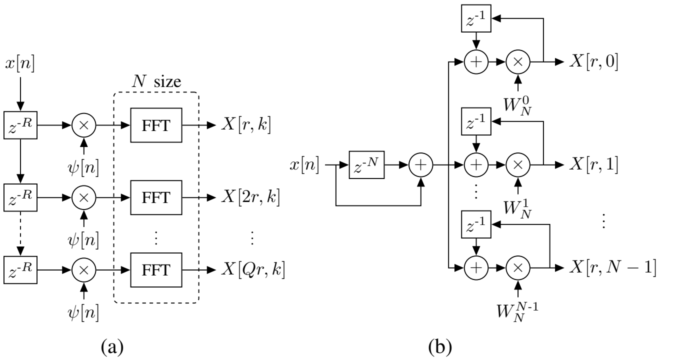

Fig. 1: Existing STFT architectures. (a) FFT based STFT. (b) Iterative STFT for $\psi[n] = 1$ and $R = 1$.

II. Background and Challenges

The STFT [5] for a discrete time sequence $x[n]$ can be computed according to:

$$X[r,k] = \sum_{n=0}^{N-1} x[rR+n]\psi[n]W_N^{kn} \mid k \in [0, N-1], \tag{1}$$

where N is the number of output frequency samples, R is the window hop size, $W_N = e^{-j2\pi/N}$, $x[n]$ is the input time sequence, $\psi[n]$ is the N-length window function and $X[r,k]$ is the output spectrum of the time step r. The conventional FFT-based approach [6] calculates $X[r,k]$ every R samples, by performing an N-sized FFT on the input data sequence, from samples $x[rR]$ to $x[rR+N-1]$, as shown in Fig. 1(a). In this case, the input data are decimated in $Q = N/R$ distinct channels, which are then multiplied by the window function followed by an N-sized FFT. The computational complexity of the required multipliers and adders of such an architecture has order $O(\frac{N^2}{R}\log_2 N)$, which is prohibitively large to be implemented on power-constrained devices.

To circumvent this, another approach relies on an iterative architecture proposed in [7]. So, the chosen window function is the rectangular ($\psi[n] = 1$) and $R = 1$, in which case the STFT of (1) is represented via the recursive formula:

$$X_{rec}[r+1,k] = W_N^{-k}\left[X_{rec}[r,k] + \big(x[r+n+N] - x[r+n]\big)W_N^{km}\right] \tag{2}$$

The resulting iterative STFT architecture is illustrated in Fig. 1(b). Although, this architecture requires only N multipliers and $2N$ memory elements, it is designed for a hop step $R = 1$, thus leading to redundant time sampling for most time-frequency analysis applications [3]. Also, if the iterative approach is employed for $R > 1$, then all the intermediate spectrums from $X[r,k]$ to $X[r+R,k]$ have to be computed. This not only results in a large number of power-consuming operations, but also in a high accumulative error, as discussed in [8]. Finally, another approach is the recently introduced feedforward STFT architecture [8]. This scheme, had no accumulative error but required $O(N\log N)$ memory complexity and was limited to $R = 1$.

By carefully examining the prior approaches, we observe that they have repetitively computed the whole [7] or a part [8] of the spectrum of each preceding sliding window, resulting in the repetition of many operations in case of $R > 1$. To better explain this, Fig. 2(a) illustrates an example with a spectral window of size $N = 256$ and hop size $R = 32$. The iterative [7] and feedforward [8] approaches, operating with a hop step $R = 1$, compute all the intermediate spectrums from $X[1,k]$ to $X[30,k]$, even when only one spectrum every 32 times step were needed. Note that even the traditional FFT based approach [6] does not reuse any overlapping data and computes independently $X[0,k]$ and $X[31,k]$, thus still performing redundant operations that could be avoided via intelligent algorithmic modifications.

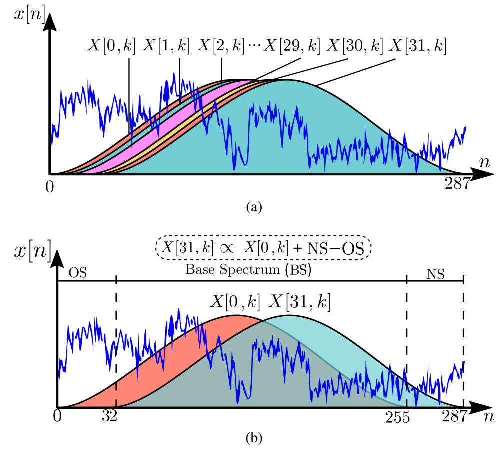

Fig. 2: (a) Computation of all intermediate spectrums [7][8] approaches. (b) Direct computation of our FD-STFT approach.

III. Proposed Approach

Aiming to reduce the redundant intermediate operations of the iterative [7] and feedforward [8] approaches that appear when $R > 1$, while also avoiding the independent spectrum computation of the FFT-based method, we are introducing a new STFT architecture, named FD-STFT, which employs a novel spectrum reutilization. Specifically, we can observe in Fig. 2(b), for N = 256 and R = 32, that there are operations that are common between the spectrums $X[0,k]$ and $X[31,k]$, as they have N-R = 224 overlapping data that can be used for the computation of both window spectrums. In order to do this, the spectrum of the first sampling window $X[0,k] = DFT\{\psi[0:255]x[0:255]\}$ is decomposed to a base spectrum (BS) derived from the 224 overlapping data between the two windows and an old spectrum (OS) stemmed from the preceding 32 non-overlapping data. Similarly, the spectrum of the second sampling window $X[32,k] = DFT\{\psi[0:255]x[32:287]\}$ can be decomposed into the BS of the first 224 samples and a new spectrum (NS) that is derived from the 32 proceeding non-overlapping data. Therefore, $X[31,k]$ can now be calculated by properly updating $X[0,k]$ using the OS and NS, as shown in Fig. 2(b), instead of computing all the intermediate spectrums $X[1,k]$ to $X[30,k]$ as shown in Fig. 2(a) and discussed in the previous section.

A. Spectrum Update

To properly update the spectrum of a proceeding time window, the correction term, based on NS and OS, has to be derived. To this end, the summation of (1), for a rectangular window function ($\psi[n] = 1$), an N-sized window with hop size R, can be decomposed as:

$$X_{rec}[r,k] = \sum_{n=0}^{R-1} x[rR+n]W_N^{kn} + \sum_{n=R}^{N-1} x[rR+n]W_N^{kn}. \quad (3)$$

For $r \to r+1$, the window function progresses R samples. Thus, by applying a change of variable $n \to n+R$, (3) can be written as:

$$X_{rec}[r+1,k] = W_N^{-kR}\left[\sum_{n=R}^{N-1} x[rR+n]W_N^{kn} + \sum_{n=N}^{N-1+R} x[rR+n]W_N^{kn}\right]. \quad (4)$$

By substituting (3) in (4), combined with a change of variables for $n \to n+N$ on the second summation term of (4), we obtain:

$$X_{rec}[r+1,k] = W_N^{-kR}\Bigg[X_{rec}[r,k] - \sum_{n=0}^{R-1} x[rR+n]W_N^{kn} + W_N^{kN}\sum_{n=0}^{R-1} x[rR+n+N]W_N^{kn}\Bigg]. \quad (5)$$

As $W_N^{kN} = 1$ and $W_N^{-kR} = W_Q^{-k}$, we finally obtain:

$$X_{rec}[r+1,k] = W_Q^{-k}\Bigg[X_{rec}[r,k] + \sum_{n=0}^{R-1}\big(x[rR+n+N] - x[rR+n]\big)W_N^{kn}\Bigg]. \quad (6)$$

By observing (6), it is evident that the new spectrum $X_{rec}[r+1,k]$ will be calculated by updating the old term $X_{rec}[r,k]$ with the correction term $Y[k] = \sum_{n=0}^{R-1} y[n]W_N^{kn}$, where $y[n] = x[rR+n+N] - x[rR+n]$. Afterwards, a multiplication with W_N^{-kR} is conducted as in (6). In this case, the OS and NS can be derived from $x[rR+n]$ and $x[rR+n+N]$, respectively, and thus, their difference in the frequency domain can be computed via a total correction term $Y[k]$. Note that (6) for $R = 1$ is essentially simplified to the iterative formula (2), making our approach a more general formulation.

B. Frequency Decomposition

As previously discussed, $X_{rec}[r+1,k]$ is calculated by updating $X_{rec}[r,k]$ according to the correction term $Y[k]$, followed by the multiplication with the twiddle factor W_Q^{-k}. If examined closely, the correction term $Y[k]$ can be computed with an N-sized FFT of the R-sized $y[n]$ with an additional $N-R$ zero padding, which results in the same computational complexity as the FFT-based approach. Thus, in order to overcome this problem an additional decomposition step is employed [9], as shown in (7):

$$Y[k] = Y[k_1 + Qk_2] = \sum_{n=0}^{R-1}\left(y[n]W_N^{k_1 n}\right)W_R^{k_2 n}, \quad (7)$$

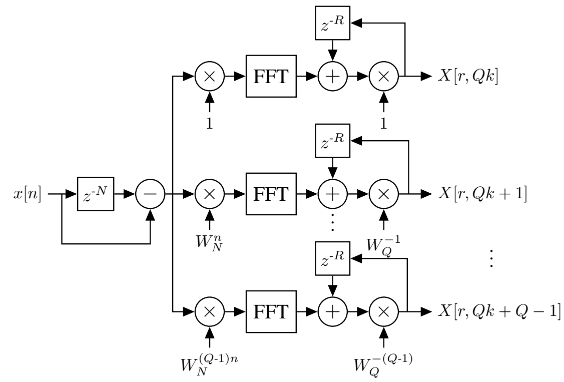

Fig. 3: Proposed FD-STFT architecture: rectangular window

where $y[n] = x[rR+n+N] - x[rR+n]$, $Q = N/R$, $k_1 \in [0, Q-1]$, $k_2 \in [0, R-1]$ and $W_R^{nk_2} = W_N^{Qk_2 n}$. Consequently, (7) can be computed by multiplying $y[n]$ with the twiddle factors $W_N^{k_1 n}$, which results in $R \cdot Q = N$ total multiplications, while the summation can be calculated with Q total R-sized FFTs, leading to $O(N \log R)$ additional operations.

C. Hanning Window Optimization

The utilization of a window function is crucial. It not only limits the spectral leakage of the side lobes compared with the traditional rectangular window [5], but also clears the resulting spectrum from any noise-imposing artifacts that may appear due to the discontinuities at the beginning and end of the input signal [5]. While being very beneficial for the STFT, the required multiplications at the time domain implies the need for additional multiplications and memory for the operations involving the windowing coefficients, leading to further hardware overhead. Additional, aside from the FFT-based STFT architecture, the iterative [7] and the feedforward [8] approaches support only a noise-prone rectangular window function. Therefore, in our approach we have replaced the rectangular window with a periodic Hanning window. To limit the relevant overhead, we have substituted the required multiplications in the time domain with a convolution in the frequency domain as in [10]. Thus, the spectrum of a Hanning windowed data sequence in our case is given by [10]:

$$X_{han}[r,k] = \tfrac{1}{2}X_{rec}[r,k] - \tfrac{1}{4}X_{rec}[r,\langle k+1\rangle_N] - \tfrac{1}{4}X_{rec}[r,\langle k-1\rangle_N], \quad (8)$$

where $\langle k+1\rangle_N$ and $\langle k-1\rangle_N$ stand for a right and left circular shift over N total frequency samples, respectively. It is important to note that the multiplication coefficients $\frac{1}{2}$ and $\frac{1}{4}$ can be easily implemented with a right shift of 1 and 2 bit positions, respectively. Therefore, by employing a Hanning window in the frequency domain, we completely avoid all the multiplications in the time domain, which were needed by existing approaches, thus reducing the complexity on this last phase of our architecture, while avoiding the spectral noise introduced by the traditionally used rectangular window [5].

D. FD-STFT architecture

Fig. 3 illustrates our proposed FD-STFT for $\psi[n] = 1$, size N and hop size R. It is composed of an initial N-sized delay

TABLE I: Comparison of STFT hardware complexity

Architecture	Complex Multipliers	Complex Adders	Complex Memory	Hop size (R)	Acc. Error
FFT-based [6]	$\frac{N}{R}(\log_2 N - 1)$	$\frac{N}{R} 2 \log_2 N$	$\frac{N^2}{R} + N$	Any	No
Iterative [7]	N	N	$2N$	1	High
Feedforward [8]	$N - 1$	$2N - 1$	$\frac{N}{2} \log_2 N$	1	No
Proposed (Rect)	$\frac{N}{R}(\log_2 R + 1)$	$\frac{N}{R}(2 \log_2 R + 1)$	$3N - \frac{N}{R}$	Any	Low

TABLE II: Comparison of STFT hardware resources

Architecture N=256, W=16, F=10 @1.1V, 45 nm,150 MHz	Hop size (R)	Area (μm)	Power (mW)	Input Samples	Latency (cycles)
FFT-based [6]	32	528812	123.34	1	160
Iterative [7]	1	414510	45.71	1	256
Feedforward [8]	1	518667	69.26	1	255
Proposed (Rect.)	32	29674	40.97	1	272
Proposed (Hann.)	32	312718	42.48	1	272

buffer that computes the difference $y[n] = x[rR + n + N] - x[rR+n]$, which is then decimated in $Q = N/R$ total channels in order to be multiplied with the proper twiddle factors $W_N^{k_1 n}$. After the initial multiplication, for each of the Q total channels, an R-sized SDF decimation in frequency (DIF) FFT [11] is employed to compute $Y[k]$ of (7). Therefore, the spectrum $X_{rec}[r+1, k]$ is calculated by updating $X_{rec}[r, k]$ (which is saved in the R-sized buffers) with the correction term $Y[k]$ (NS-OS) of the FFTs followed by the second twiddle factor multiplication (W_Q^k). Lastly, in case of a Hanning window a supplementary network of adders and bit shifters is required, acting as a weighted average filter, which is added at the output of the architecture for calculating the spectrum of (8).

IV. Results

To demonstrate the efficacy of the FD-STFT architecture we implemented it in $45nm$ technology, and compared it with [6–8], in terms of hardware complexity, power and area.

A. Complexity Analysis

Table I compares the required hardware complexity of our proposed approach with the FFT-based [6], the iterative [7] and the feedforward [8] approaches for window size N and hop size R. Note that these estimations are for a rectangular window, while in our approach, for a Hanning window we may use $\frac{2N}{R}$ additional complex adders. From the table, we can observe that the FFT-based architecture has the highest hardware complexity, as it employs Q total N-sized FFTs in parallel, which translates in high multiplier, adder and memory complexity, while our approach requires Q total FFTs of size R. Also, for $R > 1$ our proposed FD-STFT architecture employs the least number of complex multipliers and adders, compared to the previous approaches, while requiring only $\frac{R-1}{R}N$ more complex memory cells than the iterative approach. Furthermore, as R increases, the total number of hardware elements remains unchanged, for the iterative and the feedforward architecture as they are designed for $R = 1$, while in our approach the number of multipliers and adders is greatly reduced by a fraction of $\frac{\log R+1}{R}$ and $\frac{2(\log R+1)}{R}$, respectively (compared to the iterative approach).

B. Measurements

Table II compares our proposed architecture with previous ones in terms of area (μm^2) and power (mW), in case of

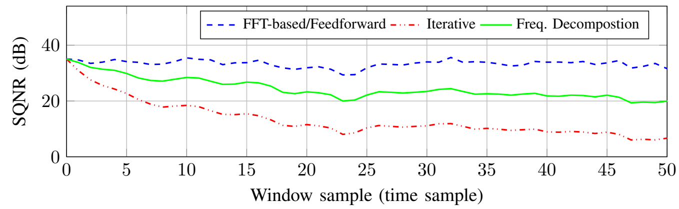

Fig. 4: Average SQNR degradation of the spectrum for 50 windows with $N = 256$, $R = 32$, $W = 16$ and $F = 10$.

$N = 256$, $R = 32$ for the FFT-based [6] and our approach and $R = 1$ for [7, 8], with word length $W = 16$ and fractional part $F = 10$. Note that, for our approach we have employed both the rectangular and Hanning windows, while for the others the rectangular window. For obtaining the results, we synthesized the circuits using the 45 nm NangateOpenCell library at 1.1 V at clock frequency of 150 MHz. Specifically, regarding the total area, our approach combined with the hanning window outperforms all the existing ones achieving 24.55%, 39.70% and 40.86% area reduction, compared with [7], [8] and [6], respectively. Also, our approach consumes 7.07%, 38.66% and 65.56% less power compared to [7], [8] and [6], respectively. Finally, all the tested architectures process 1 input sample per clock cycle, while our approach has the highest initial latency of 272 clock cycles as a N-sized delay buffer along with R-sized pipelined FFTs are utilized.

C. Quality degradation

Regarding the quality of the computed spectrums, we have calculated the average signal to quantization noise ratio (SQNR) of the outputs of our synthesized circuits by performing a gate-level simulation with 1000 audio samples [12] as inputs. Specifically, as illustrated in Fig. 4, we have computed the spectrum using our proposed approach, the iterative and FFT-based STFT, for 50 sampling windows for $N = 256$, $R = 32$, $W = 16$ and $F = 10$. The FFT-based and feedforward STFTs [6, 8] achieve SQNR of approximately 37 dB for all windows with no accumulative error. Also, when the iterative approach [7] is applied, the SQNR drops to 2.1 dB at the 50^{th} window calculation due to the $50 \cdot 32 = 1600$ intermediate spectrum computations. Finally, considering our approach, using the $R = 32$ sized FFTs allows us to efficiently reuse the base spectrum of the overlapping input data, while also inducing a lower quantization noise that results in 17.9 dB higher SQNR compared to the iterative approach for the 50^{th} window. Note that the R in our approach can be adjusted to any other value that suits the target data.

V. Conclusion

This paper, has presented a new approach, the FD-STFT, for the computation of the STFT algorithm with overlapping windows of hop size R greater than one. It does not only lead to substantial area and power savings by up to 40.86% and 65.56%, for $R = 32$ respectively, but also improves the achieved output SQNR by 17.9 dB compared to the iterative approaches.

REFERENCES

[1] Y. Tu and M.-W. Mak, "Aggregating frame-level information in the spectral domain with self-attention for speaker embedding," *IEEE/ACM Trans. Audio Speech Lang.*, vol. 30, pp. 944–957, Feb. 2022.

[2] K. Li, H. Rüdiger, and T. Ziemssen, "Spectral analysis of heart rate variability: Time window matters," *Front. Neurol.*, May 2019.

[3] C. De la Fuente, E. Martinez-Valdes, J. I. Priego-Quesada, A. Weinstein, O. Valencia, M. R. Kunzler, J. Alvarez-Ruf, and F. P. Carpes, "Understanding the effect of window length and overlap for assessing sEMG in dynamic fatiguing contractions: A non-linear dimensionality reduction and clustering," *J. Biomechanics*, vol. 125, June 2021.

[4] Q. Lv and H. Qin, "An improved method based on time-frequency distribution to detect time-varying interference for GNSS receivers with single antenna," *IEEE Access*, vol. 7, pp. 38 608–38 617, Mar. 2019.

[5] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing, 3rd edition*, 2009.

[6] S. Zhang, D. Yu, and S. Sheng, "A discrete STFT processor for real-time spectrum analysis," in *APCCAS*, April 2006, pp. 1943–1946.

[7] K. R. Liu, "Novel parallel architectures for short-time fourier transform," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 12, pp. 786–790, Dec. 1993.

[8] M. Garrido, "The feedforward short-time Fourier transform," *IEEE Trans. Circuits Syst. II*, vol. 63, no. 9, pp. 868–872, Sep. 2016.

[9] H. Sorensen and C. Burrus, "Efficient computation of the DFT with only a subset of input or output points," *IEEE Trans. Signal Process.*, vol. 41, no. 3, pp. 1184–1200, Mar. 1993.

[10] D. Petranovic, S. Stankovic, and L. Stankovic, "Special purpose hardware for time frequency analysis," *Electron. Lett.*, Mar. 1997.

[11] H. Groginsky and G. Works, "A pipeline fast Fourier transform," *IEEE Trans Comp.*, vol. C-19, no. 11, pp. 1015–1019, Nov. 1970.

[12] A. Coucke, M. Chlieh, T. Gisselbrecht, D. Leroy, M. Poumeyrol, and T. Lavril, "Efficient keyword spotting using dilated convolutions and gating," in *IEEE ICASSP*, April 2019, pp. 6351–6355.