

Implementação de FFT de 1600 pontos em microcontrolador ARM Cortex-M7 para cálculo de harmônicas em sistema elétrico trifásico

Gustavo Souza

Universidade Federal do Pará
Belém, Pará, Brasil
gustavo.gomes@itec.ufpa.br

Abraão Ferreira

Universidade Federal do Pará
Belém, Pará, Brasil
abrahao.ferreira@itec.ufpa.br

Flávio Nunes

Universidade Federal do Pará
Belém, Pará, Brasil
flavio.nunes@itec.ufpa.br

Adalbery Castro

Universidade Federal do Pará e Polícia
Científica do Pará
Belém, Pará, Brasil
adalbery@ufpa.br

Leonardo Ramalho

Universidade Federal do Pará
Belém, Pará, Brasil
leonardolr@ufpa.br

Aldebaro Klautau

Universidade Federal do Pará
Belém, Pará, Brasil
aldebaro@ufpa.br

RESUMO

O monitoramento contínuo da qualidade da energia elétrica (QEE) é essencial para fornecedores e consumidores de energia elétrica. Dentre as métricas avaliadas, as distorções harmônicas são distúrbios em frequência indesejados nos sistemas elétricos e, por isso, são importantes parâmetros de QEE. O cálculo da Transformada Rápida de Fourier (FFT) é uma das maneiras de estimar as harmônicas e pode exigir um hardware relativamente sofisticado quando implementado em tempo real e com número de pontos relativamente alto. Buscando equipamentos de relativo baixo custo, este artigo apresenta estratégias de processamento de sinais e de implementação do cálculo de uma FFT de 1600 pontos para estimativa das distorções harmônicas. O algoritmo desenvolvido foi embarcado em um equipamento para monitorar sistemas de alimentação elétrica de corrente alternada trifásicos, que deve ser de classe S e atender às normas IEC 61000-4-7 e IEC 61000-4-30. Serão mostradas, em diferentes implementações, quais foram as otimizações realizadas em software e hardware para a realização do cálculo desta FFT, utilizando um microcontrolador ARM Cortex-M7. Foram avaliados as quantidades de memória flash e RAM requeridas, o número de operações complexas realizadas e o tempo necessário para a execução da transformada, levando em consideração a utilização ou não das memórias cache e da TCM (Memória Altamente Acolpada) presentes no microcontrolador empregado no projeto. Os resultados mostraram uma diminuição significativa do custo computacional, especialmente quando se utilizou as memórias cache e implementou-se a FFT com decimação no tempo.

KEYWORDS

redes elétricas trifásicas, FFT, memória cache, TCM, ARM Cortex-M7, IEC 61000-4-30, IEC 61000-4-7, ATSAMV71, ADE9000

1 INTRODUÇÃO

O mercado de sistemas embarcados não para de crescer e cada vez mais tem-se buscado soluções conectadas de menor custo. Neste âmbito, os autores projetaram hardware de equipamento para realizar o monitoramento e aquisição de dados da rede elétrica trifásica e, entre as diversas especificações do equipamento, inclui o levantamento das distorções harmônicas totais e individuais até a 50ª

harmônica para corrente e tensão nas três fases, sendo o cálculo da transformada de Fourier para atender esta especificação o principal objeto deste artigo.

O equipamento projetado foi denominado de Unidade de Aquisição de Corrente e Tensão Trifásica (UACT-3F) e é constituído de um sistema embarcado que utiliza circuito integrado especializado em realizar medidas de grandezas elétricas em redes trifásicas (ADE9000) fabricado pela Analog Devices e um microcontrolador ARM Cortex-M7 (ATSAMV71Q21B) fabricado pela Microchip Technology Inc., devido a sua versatilidade, facilidade de desenvolvimento e suporte nativo a vários periféricos. Entre outras características, o equipamento possui comunicação Modbus [8] via RS-485, por onde se comunica com módulo externo de terceiros como forma de integração à rede LoRa implementada no campus universitário.

O cálculo das distorções harmônicas individuais não é realizado pelo ADE9000 e é uma das partes mais críticas do sistema, pois é desejada para as três tensões e quatro correntes (incluindo corrente no neutro) e a norma ABNT NBR IEC 61000-4-30:2011 [1] prevê em seu item 5.8.4 (remete aos itens 4.4 e 4.5) a possibilidade de lacunas na agregação das tensões harmônicas, mas pelo menos um valor de 10/12 ciclos deve ser utilizado a cada 50/60 ciclos. Isto implica que o cálculo das distorções harmônicas deve ocorrer para as tensões a cada segundo. Aplicando a mesma regra para as correntes, então são sete cálculos de distorções harmônicas individuais por segundo para atender à norma quanto a agregação em equipamento classe S.

Deve-se obter a forma de onda através dos conversores analógico/digital (ADC) do ADE9000 para então realizar a análise em software, o que pode representar um gargalo, tanto no tempo para receber os dados via SPI (Serial Protocol Interface), quanto no tempo para calcular as harmônicas a partir dos sete sinais obtidos.

Para calcular as distorções harmônicas de um sinal digital de comprimento finito, deve-se utilizar a Transformada Direta de Fourier (DFT – Direct Fourier Transform). O cálculo de uma DFT é custoso computacionalmente, pois tem complexidade $O(N^2)$ e envolve o cálculo de senos e cossenos. Desde a publicação do algoritmo de Cooley e Tukey [3], vários algoritmos de complexidade $O(N \log N)$ foram propostos para o cálculo rápido de uma DFT, o que ficou conhecido como Transformada Rápida de Fourier (FFT – Fast Fourier Transform). Esses algoritmos utilizam o método conhecido como

“dividir para conquistar”, que divide o sinal de entrada em várias partes e aglutina essas partes progressivamente em estágios, até obter-se a saída matematicamente equivalente à definição da DFT.

A maioria dos algoritmos de FFT implementados utiliza-se de sinais de comprimento igual a uma potência inteira de 2 (128, 256, 512, 1024, etc.) devido à facilidade de implementação e à alta eficiência de FFTs de base 2. Entretanto, as especificações para a UACT-3F exigem que sejam utilizados 200 ms do sinal (12 ciclos a 60 Hz) e que a transformada obtida tenha resolução de 5 Hz. O ADE9000 possui taxa de amostragem de 8000 amostras por segundo, fazendo com que 200 ms de sinal tenha 1600 pontos, que não é uma potência inteira de 2. Decidiu-se não reamostrar o sinal, o que permitiria trabalhar com um sinal de comprimento igual a uma potência inteira de 2, pois quis-se evitar acrescentar uma fonte de erros devido a uma interpolação. Trabalhos como o [5] escolheram outra abordagem, realizando a interpolação.

Dada a dificuldade de encontrar uma biblioteca que implementasse diretamente o código de um FFT que servisse para tal número de pontos, optou-se por implementar essa função para este caso específico na linguagem de programação C, considerando o microcontrolador escolhido. Este trabalho descreve as etapas de implementação de uma FFT de 1600 pontos, em busca das otimizações tanto em nível matemático quanto em nível dos recursos de hardware disponíveis pelo microcontrolador.

A Seção 2 descreve os requisitos que motivaram este trabalho. Os recursos de hardware disponíveis são apresentados na Seção 3. As etapas de implementação do algoritmo de FFT são descritas ao longo da Seção 4. A Seção 5 expõe a metodologia de avaliação das diferentes etapas e configurações do sistema. Os resultados são apresentados e discutidos na Seção 6. Enfim, as considerações finais são abordadas na Seção 7.

2 MONITOR DE SISTEMAS ELÉTRICOS TRIFÁSICOS

O equipamento elaborado (UACT-3F) e apresentado parcialmente neste artigo consiste em um monitor da qualidade de energia da rede trifásica associada a um sistema de geração fotovoltaico e eletropostos (para veículos elétricos) disponíveis no campus universitário. Este equipamento deve medir, entre outras coisas, tensão RMS, corrente RMS, frequência, potências ativa, reativa e aparente, distorções harmônicas total e individuais e eventos extremos de tensão (elevação e abaixamento de tensão). O equipamento pronto e instalado é mostrado na Figura 1.

O monitoramento realizado para indicar problemas na qualidade da energia elétrica é importante para reduzir prejuízos econômicos, que vão desde aumento significativo no consumo de energia até efeitos catastróficos, como queima de equipamentos da rede de transmissão de energia e consequentemente interrupção na distribuição de energia [2].

Após pesquisa no mercado por circuitos integrados que poderiam ser utilizados para auxiliar no monitoramento desejado, decidiu-se utilizar o ADE9000, da Analog Devices Inc., que é especializado em medir grandezas elétricas em sistemas elétricos trifásicos. Ele é provido de entradas capazes de medir tensão e corrente alternada e de uma unidade de processamento de sinais (DSP) para realizar o processamento das medições sobre 12 ciclos em rede de 60Hz (ou



Figura 1: UACT-3F instalada ao lado de um quadro elétrico de baixa tensão para medir as grandezas elétricas de interesse.

10 ciclos em rede de 50 Hz), para obter tensão e corrente RMS em tempo real, seguindo a norma IEC 61000-4-30 para equipamentos de classe S [1, 4].

A Figura 2 ilustra o fluxo de dados que passa pelo chip ADE9000: condicionadores de sinais para tensão e corrente conectados à rede elétrica capturam os sinais para que sejam digitalizados pelo ADE9000, que armazena o sinal no waveform buffer e calcula os valores de RMS, de potência, entre outros. Os dados são consultados via SPI pelo microcontrolador, que realiza conversões dos dados brutos obtidos do ADE9000, calcula as agregações e grandezas restantes não obtidas do ADE9000 e armazena as informações. Os dados são disponibilizados através do protocolo Modbus utilizando uma porta USART/RS485.

Entre as medidas a serem obtidas, a distorção harmônica individual é uma importante métrica para avaliar a qualidade da energia, pois esta medida mostra a relação entre a amplitude da componente fundamental e as harmônicas da rede elétrica. Atualmente é extremamente importante avaliar esse parâmetro, devido ao aumento do uso de cargas não lineares conectadas na rede elétrica, tais como lâmpadas LED, computadores, televisores, impressoras, etc, principalmente em prédios comerciais [6].

O ADE9000 permite a leitura da forma de onda dos sinais digitalizados a partir de um buffer interno, servindo para análises de distorção harmônica individual até a desejada quinquagésima (50ª) harmônica. Para tanto são necessários 200 ms do sinal (12 ciclos em rede de 60 Hz) e necessita-se de uma resolução de 5 Hz para os cálculos de grupos e subgrupos de frequências harmônicas (uso de frequências vizinhas no cálculo do valor das harmônicas), de acordo com a norma IEC 61000-4-7. Logo, dada a frequência de amostragem, deve-se obter 1600 pontos, o que permite alcançar o espaçamento espectral adequado [7].

3 MICROCONTROLADOR ARM

Para implementar a UACT-3F, foi escolhido o microcontrolador ATSAMV71Q21B da Microchip Technology Inc [10], utilizando o software Microchip Studio para realizar a programação do sistema

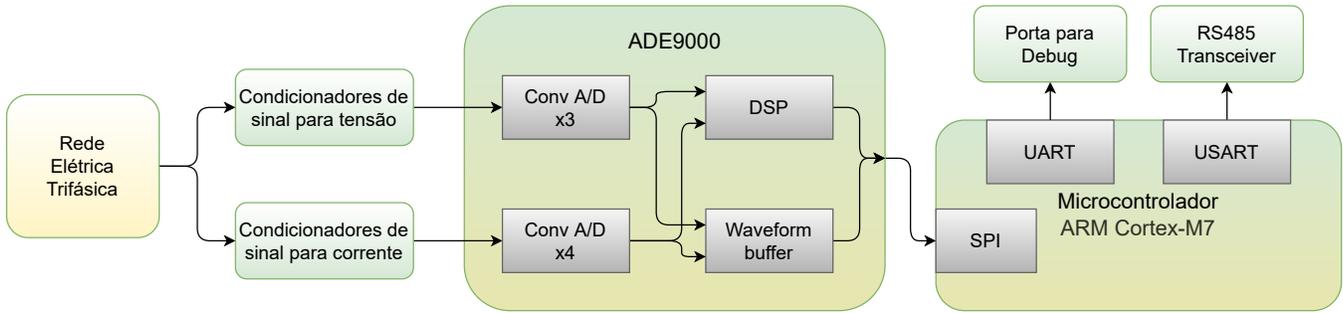


Figura 2: Diagrama simplificado da Unidade de Aquisição de Corrente e Tensão Trifásica (UACT-3F).

e a ferramenta online Atmel START [9] para gerar o código de inicialização e configuração do microcontrolador. Esse microcontrolador contém um processador ARM Cortex-M7 de um núcleo funcionando a 300 MHz com 384 KB de memória RAM, 2 MB de memória Flash, 16 KB de memória cache de dados, 16 kB de memória cache de instruções, unidade de ponto flutuante de precisão simples e dupla, 16 kB de memória de bootloader, suporte a memória SDRAM externa, unidade de Acesso Direto à Memória (DMA – Direct Memory Access) e suporte a vários periféricos, como SPI, I²C, Ethernet, UART e cartão SD. A memória RAM pode ser segmentada em blocos que constituem a TCM (Tightly Coupled Memory – Memória Altamente Acoplada, em tradução livre) de dados e instruções de até 128 KB cada.

Durante os testes iniciais do sistema, foi utilizada a placa de desenvolvimento SAM V71 Xplained Ultra [11], que possui suporte a vários periféricos e um debugger que inclui uma porta UART, LEDs e uma interface de troca de dados SPI. Essa placa permitiu testar o funcionamento de vários componentes do sistema como a comunicação UART e a comunicação SPI com o ADE9000 antes da conclusão da placa de circuito impresso da UACT-3F.

Para a otimização do sistema, foi decidido estudar o impacto das memórias cache e da TCM.

As memórias cache são duas memórias separadas da memória RAM e são configuradas com funções disponibilizada na API do Microchip Studio. A memória cache de dados não pode ser usada juntamente com algum periférico que acesse diretamente a memória (ex.: DMA), pois isso pode gerar inconsistências no programa, já que neste caso o processador não acessa a memória RAM diretamente, mas sua cópia na cache. Para solucionar isso, as variáveis a serem trabalhadas com os periféricos devem ser colocadas em uma região que não vai para a memória cache, como a região da TCM.

A TCM é implementada como parte da memória RAM principal que passa a ser acessível diretamente pelo processador com sua velocidade de trabalho (300 MHz), que difere do restante da memória RAM, que é acessível através de um barramento e, consequentemente, mais lenta. Para uso da TCM, é necessária sua configuração na memória flash específica para configurações do microcontrolador (bits GPNVM), permitindo os tamanhos de 0, 32, 64 ou 128 KB. Para a UACT-3F, foi utilizada somente a TCM de dados pela simplicidade de configuração e, para tanto, as variáveis globais a serem postas na TCM são configuradas explicitamente em sua declaração usando a palavra chave `__attribute__`.

4 ETAPAS DE IMPLEMENTAÇÃO DA FFT

Para implementar a FFT de 1600 pontos, uma série de etapas foram feitas buscando diminuir o tempo de cálculo e o consumo de memória sem prejudicar a exatidão do resultado.

Para testar a exatidão dos resultados durante a implementação, foi inicializado um vetor de 1600 posições (200 ms a 8 ksps) com algumas harmônicas de 60 Hz, com o fim de testar a saída da FFT.

Para efetuar cálculos com números complexos, foi utilizada a biblioteca `tgmath`, que, entre outras coisas, define uma notação em linguagem C para números complexos e suas principais operações, incluindo funções trigonométricas [13].

As subseções a seguir descrevem as etapas tomadas para implementar a FFT de uma forma cada vez mais eficiente.

4.1 Definição de uma DFT

Primeiramente, foi implementado o código sem otimizações da definição da Transformada Discreta de Fourier (DFT) de um sinal discreto $x[n]$ de comprimento N [12]:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad (1)$$

onde $k = 0, 1, \dots, N - 1$, e $W_N = e^{-j(2\pi/N)}$. Os coeficientes W_N são calculados diretamente a cada iteração de n .

Esta versão foi rotulada como DFT1600.

4.2 Implementação da FFT

É possível decompor o valor 1600 em fatores primos, obtendo $2 \times 2 \times 2 \times 2 \times 2 \times 5 \times 5$ ou $2^6 \times 5^2$. Então foi implementado algoritmo de FFT com decimação (ou dizimação) no tempo, onde são empregados sete estágios de decomposição para o cálculo da FFT. A Figura 3 ilustra o conceito da decimação da FFT, da direita para a esquerda. O estágio 0 (zero) resulta em 64 DFTs de 25 pontos. Então, para o cálculo da FFT, os estágios fazem aglutinação em pares de blocos do estágio anterior, começando de 25 em 25 pontos e dobrando a quantidade de pontos até chegar em 1600 pontos no último estágio.

4.5 Incorporação do código em uma única função

A próxima etapa consistiu em incorporar o código da função de cálculo da FFT de 25 pontos na função principal para diminuir o tempo consumido em chamadas de função, pois antes o cálculo da FFT de 25 pontos estava numa função à parte para facilitar a legibilidade do código.

Esta versão foi rotulada de FFT1600D.

4.6 Usar variáveis de precisão dupla

Os vetores de ponto flutuante usados para todas as etapas de cálculo da FFT tiveram sua precisão alterada de simples (`float`) para dupla (`double`) para verificar o impacto na performance do código. Espere observar mais consumo de memória RAM e flash.

Esta versão foi rotulada de FFT1600E.

4.7 Redução do cálculo do vetor de saída

Por fim, retornou-se ao uso de variáveis de precisão simples e o último estágio da FFT foi reduzido para calcular somente os $X[k]$ necessários para o sistema final, limitando os valores de k na Equação 2. Neste caso, para o cálculo do grupo da 50ª harmônica, necessita-se das frequências até 3030 Hz (606ª posição no vetor).

Esta versão foi rotulada de FFT1600F.

5 METODOLOGIA

Para realizar os testes do sistema, foi elaborado um código de base a ser testado na placa de desenvolvimento SAM V71 Xplained Ultra usando a ferramenta Atmel START, incluindo o uso da interface UART para debug, o RTC e um LED de sinalização. Em seguida, foram adicionados os códigos para lidar com as memórias cache, TCM e o contador de ciclos do processador (para contar o tempo de execução dos trechos de código relevantes).

A cada versão da FFT, foi feita uma cópia do código de base, adicionado a respectiva implementação da FFT, acrescentada de códigos para contar o tempo total do cálculo e os números de somas e multiplicações complexas realizados.

Primeiro foi contado o tempo de cálculo para cada uma das seguintes configurações:

- Sem cache e sem TCM,
- Com cache de dados e de instruções,
- Com TCM,
- Com TCM e cache de instruções,
- Com TCM e cache de dados e de instruções.

Como todas as variáveis usadas no cálculo da FFT foram colocadas na TCM quando esta foi usada, foi testado se haveria diferença entre ativar só a cache de instruções em associação à TCM e ativar ambas as caches em associação à TCM.

Em seguida, foi feita a contagem do número de operações de somas e multiplicações complexas, excluindo-se as contas para encontrar os coeficientes W_N , e foram obtidas do compilador a quantidade de memória flash e RAM usadas no sistema para cada implementação (excluindo-se os códigos para contagem de tempo e de número de operações).

Para verificar a precisão dos resultados, utilizou-se o software para computador MATLAB para realizar os cálculos de FFT. A

implementação nativa de FFT foi utilizada neste caso. O sinal de entrada consistiu em um sinal emulado de 60 Hz a 8 ksp/s de 1600 pontos com as 3ª, 9ª, 20ª, 31ª, 38ª e 48ª componentes harmônicas não nulas. Os resultados foram comparados com os resultados da implementação FFT1600F.

Para cada uma das implementações descritas, as estatísticas de tempo e consumo de recursos correspondem ao cálculo de uma FFT de 1600 pontos. Para a implementação da UACT-3F, deve-se considerar sete vezes os valores de tempo descritos neste artigo, de forma a se considerar os três sinais de tensão e quatro sinais de corrente. O cálculo das distorções harmônicas utiliza a FFT implementada, mas os resultados apresentados a seguir correspondem à implementação de um único cálculo de FFT.

6 RESULTADOS

A Tabela 1 mostra os resultados obtidos comparando as implementações da FFT: os tempos de cálculo da FFT usando diferentes configurações de memória, os números de somas e de multiplicações de cada implementação, e as quantidades de memória flash e RAM utilizadas em cada implementação.

Pode-se notar que as sucessivas alterações na implementação da FFT diminuíram bastante o tempo de cálculo, principalmente a implementação de estágios (FFT1600A) e a fixação dos coeficientes na flash (FFT1600B). Somente a substituição por variáveis de precisão dupla (FFT1600E) se mostrou menos eficiente no tempo, levando mais que o dobro do que usando precisão simples, além de consumir muito mais memória flash e RAM. Por isso, resolveu-se retornar ao uso de variáveis de precisão simples na versão seguinte.

Quanto ao uso de memória cache ou TCM, notou-se que o uso das memórias caches contribuiu para a diminuição do tempo de cálculo (em mais de 6 vezes), enquanto o uso de TCM para dados mostrou ter tido pouco impacto no tempo de cálculo. Por outro lado, o uso de memória cache de dados mostrou-se efetivo mesmo quando todas as variáveis usadas pelas rotinas de FFT estavam na TCM (região não armazenável em cache).

O número de operações complexas reduziu bastante quando a FFT foi implementada em estágios (FFT1600A) e depois de se levar em conta a simetria do sinal de entrada (FFT1600C), ficando quase inalterado com as outras alterações. As quantidades de memória flash e RAM utilizadas, por outro lado, aumentaram nessas mesmas implementações.

Para efeito de comparação, o firmware utilizado de base para as implementações descritas foi posteriormente compilado tendo como única diferença a remoção da chamada da função da FFT na função principal, o que gerou um executável utilizando 20.688 bytes na Flash (contra 30.288 na DFT1600 e 33.464 na FFT1600F) e 24.048 bytes na RAM (contra 24.064 na DFT1600 e 36.896 na FFT1600F). A diferença de pelo menos 9.600 bytes na flash se deve aos códigos de cálculo das funções complexas envolvidas e posteriormente dos coeficientes fixados nela, enquanto a diferença no uso da RAM se deve à necessidade de se armazenar mais valores entre os estágios.

Quanto à diferença de precisão entre simples e dupla, a Figura 5 mostra em cima o sinal simulado para comparar o resultado da FFT (versão FFT1600F) em valores absolutos no meio e em dBV embaixo, pelo código implementado em microcontrolador (em vermelho) e em computador (em azul), utilizando o software MATLAB

Tabela 1: Resultados obtidos pelas implementações da FFT de 1600 pontos em termos absolutos (em cima) e relativos a FFT1600F (embaixo).

Versão	Só RAM	Cache	TCM	Cache I + TCM	Cache + TCM	Somas	Multiplicações	Flash (B)	RAM (B)
DFT	7 min 3 s	1 min 4 s	6 min 42 s	1 min 30 s	1 min 3 s	2.560.000	2.560.000	30.288	24.064
FFT1600A	11,984 s	1,873 s	12,621 s	2,613 s	1,865 s	48.000	41.601	33.816	35.880
FFT1600B	291,307 ms	34,309 ms	280,454 ms	66,484 ms	33,765 ms	48.000	41.601	35.608	36.888
FFT1600C	167,559 ms	20,064 ms	155,868 ms	36,986 ms	19,836 ms	29.568	23.169	33.272	36.888
FFT1600D	165,183 ms	20,092 ms	157,394 ms	36,362 ms	19,744 ms	29.568	23.169	33.272	36.896
FFT1600E	328,748 ms	44,575 ms	314,890 ms	82,790 ms	43,521 ms	29.568	23.169	41.312	68.900
FFT1600F	162,713 ms	19,657 ms	161,959 ms	36,650 ms	19,459 ms	28.576	22.976	33.464	36.896
DFT	2599,67	3255,84	2482,11	2455,66	3237,58	89,586	111,42	0,9051	0,6521
FFT1600A	73,65	95,28	77,93	71,30	95,84	1,6797	1,8106	1,0105	0,9725
FFT1600B	1,7903	1,7454	1,7316	1,8140	1,7352	1,6797	1,8106	1,0641	0,9998
FFT1600C	1,0298	1,0207	0,9624	1,0092	1,0194	1,0347	1,0084	0,9943	0,9998
FFT1600D	1,0152	1,0221	0,9718	0,9921	1,0146	1,0347	1,0084	0,9943	1,0000
FFT1600E	2,0204	2,2676	1,9443	2,2589	2,2365	1,0347	1,0084	1,2345	1,8674
FFT1600F	1	1	1	1	1	1	1	1	1

para executar os cálculos de FFT. Percebe-se a diferença de precisão no piso (componentes harmônicos não presentes no sinal), pois o MATLAB usa variáveis de precisão dupla, enquanto nossa última implementação usa precisão simples. Verificou-se que, mesmo usando variáveis de precisão simples, conseguiu-se obter valores em geral abaixo de -70 dBV ($10^{-3,5}$ V). Os picos obtidos somente pela implementação no microprocessador em frequências maiores que 3030 Hz são devidos à implementação parcial do último estágio (versão FFT1600F). Estes picos são irrelevantes, pois serão desconsiderados pela aplicação final onde se deseja informações apenas até a 50ª harmônica (3000 Hz).

7 CONCLUSÃO

A implementação de uma FFT é sempre um processo delicado cheio de nuances em se tratando da plataforma que realiza o cálculo e do tamanho e características do sinal de entrada. Como esse cálculo faz parte de um sistema maior de baixo custo e fácil configuração, foi escolhido utilizar um microcontrolador como núcleo do equipamento desenvolvido, a UACT-3F.

Encontrar uma implementação satisfatória que atendesse com folga os requisitos de tempo sem consumir muita memória foi um exercício interessante, que combinou habilidades em processamento de sinais discretos e em arquitetura de microcontroladores. A implementação da decimação no tempo e de memória cache apresentaram melhorias significativas ao sistema quanto ao tempo de cálculo, embora o consumo de memória flash e RAM tenham aumentado devido ao aumento da complexidade do código (comparando DFT com FFT). O acréscimo de constantes para armazenamento dos coeficientes W_N , calculados previamente, também contribuiu para a redução do tempo de execução e aumento do uso da memória flash.

O uso da memória cache, que é facilmente configurável, permitiu reduzir em mais de 6 vezes o tempo de cálculo, o que consiste uma melhoria significativa para a aplicação desejada. Sem esta melhoria o equipamento não conseguiria atender as normas quanto ao tempo de agregação. Por outro lado, a TCM não apresentou

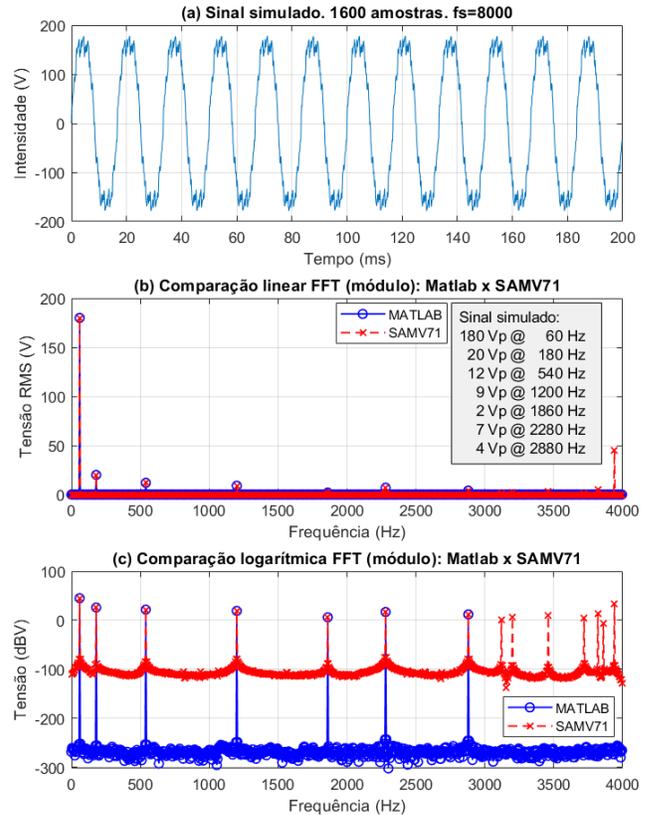


Figura 5: Sinal usado para testes da FFT (em cima) e comparação dos valores absolutos em escala linear (no meio) e em escala logarítmica (embaixo) obtidos em simulação e pelo microprocessador.

o efeito esperado, que permitiria uma comunicação mais rápida

XIV Computer on the Beach

30 de Março a 01 de Abril de 2023, Florianópolis, SC, Brasil

com o processador. Quando a TCM foi utilizada, o uso da memória cache de instruções (Cache I + TCM) apresentou um desempenho intermediário entre a versão sem cache (TCM) e a versão com as duas caches (Cache + TCM), mostrando que a memória cache de dados teve influência mesmo quando os dados dos cálculos estavam todos na TCM.

Outras implementações devem ser avaliadas, como o emprego de 3 estágios de base 4 no lugar de 6 estágios de base 2, cálculo usando variáveis de ponto fixo ou o uso de interpolação para gerar um sinal com comprimento igual a uma potência inteira de 2. Além disso, será feito o estudo da integração da implementação da FFT com os demais elementos do sistema, como o sistema operacional de tempo real utilizado no equipamento e os outros cálculos que o sistema deverá realizar.

ACKNOWLEDGMENTS

Agradecimentos à Norte Energia pelo financiamento necessário para o desenvolvimento da UACT-3F. À Universidade Federal do Pará e ao Parque de Ciência e Tecnologia Guamá pelo espaço, laboratório e instrumentos disponibilizados. À Polícia Científica do Pará pelo incentivo à capacitação dos Peritos Criminais.

REFERÊNCIAS

- [1] ABNT NBR IEC 61000-4-30:2021 2021. *ABNT NBR IEC 61000-4-30:2021. Compatibilidade eletromagnética Parte 4-30: Técnicas de medição e ensaio – Métodos de*

- medição de qualidade da energia*. International Standard. Associação Brasileira de Normas Técnicas.
- [2] João L. Afonso and Júlio S. Martins. 2003. As Oportunidades das Ameaças: A Qualidade da Energia Elétrica. Ciclo de Seminários, Departamento de Electrónica Industrial, Universidade do Minho Campus de Azurém, Guimarães.
- [3] James Cooley and John Tukey. 1965. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comp.* 19, 90 (1965), 297–301.
- [4] Analog Devices. 2017. *ADE9000: High Performance, Multiphase Energy, and Power Quality Monitoring IC*. <https://www.analog.com/media/en/technical-documentation/data-sheets/ade9000.pdf>
- [5] Giovanni Artale et al. 2020. PQ and Harmonic Assessment Issues on Low-Cost Smart Metering Platforms: A Case Study. (Nov. 2020).
- [6] M. Jawad Ghorbani and Hossein Mokhtari. 2015. Impact of Harmonics on Power Quality and Losses in Power Distribution Systems. *International Journal of Electrical and Computer Engineering (IJECE)* 5, 1 (Feb. 2015), pag. 166–174.
- [7] IEC 61000-4-7:2002 2002. *IEC 61000-4-7:2002*. International Standard. International Electrotechnical Commission.
- [8] Modbus Organization Inc. 2005. *Modbus Specifications and Implementation Guides*. Retrieved 20 de outubro de 2022 from <https://www.modbus.org/specs.php>
- [9] Microchip Technology Inc. 2022. *Atmel START*. Retrieved 26 de outubro de 2022 from <https://start.atmel.com/>
- [10] Microchip Technology Inc. 2022. *ATSAMV71Q21 | Microchip Technology*. Retrieved 20 de outubro de 2022 from <https://www.microchip.com/en-us/product/ATSAMV71Q21>
- [11] Microchip Technology Inc. 2022. *SAM V71 XPLAINED ULTRA EVALUATION KIT | Microchip Technology*. Retrieved 20 de outubro de 2022 from <https://www.microchip.com/en-us/development-tool/atsamv71-xult>
- [12] Alan V. Oppenheim and Ronald W. Schaffer. 2013. *Processamento em tempo discreto de sinais* (3ª ed. ed.). Pearson, São Paulo. 688 pages.
- [13] Cpp Reference. 2022. *Type-generic math - cppreference.org*. Retrieved 28 de outubro de 2022 from <https://en.cppreference.com/w/c/numeric/tgmath>