

CENTRO UNIVERSITÁRIO UNIFACVEST
CURSO ENGENHARIA ELÉTRICA
CLEITON DAL AGNOL

**COMPARAÇÃO ENTRE MICROCONTROLADORES E APLICAÇÃO
DO FPGA NO CONTROLE DO CONVERSOR BOOST**

LAGES

2018

CLEITON DAL AGNOL

**COMPARAÇÃO ENTRE MICROCONTROLADORES E APLICAÇÃO
DO FPGA NO CONTROLE DO CONVERSOR BOOST**

Trabalho de conclusão de curso apresentado
no Centro Universitário UNIFACVEST como
parte do requisito para obtenção de grau de
Bacharelado em Engenharia Elétrica

Orientadora: Prof^ª. Dra. Franciéli Lima de Sá

LAGES,

2018

CLEITON DAL AGNOL

**COMPARAÇÃO ENTRE MICROCONTROLADORES E APLICAÇÃO
DO FPGA NO CONTROLE DO CONVERSOR BOOST**

Trabalho de conclusão de curso apresentado
no Centro Universitário UNIFACVEST como
parte do requisito para obtenção de grau de
Bacharelado em Engenharia Elétrica

Orientadora: Prof^ª. Dra. Franciéli Lima de Sá

Lages, SC ___/___/2018. Nota: ___

Coordenadora: Dra. Franciéli Lima de Sá

LAGES

2018

AGRADECIMENTOS

A minha orientadora Dra. Franciéli Lima de Sá, pelo suporte e ajuda no pouco tempo que lhe coube, pelas suas correções, amizade, incentivos, críticas construtivas, simpatia, atenção, paciência e pelo excelente trabalho prestado durante os anos de formação sendo um bom exemplo e referência profissional à ser seguido que de certa forma contribuiu para meu desenvolvimento intelectual, acadêmico e profissional.

Aos professores membros da minha banca Sergio Luis Brockveld Junior e Silvio Moraes de Oliveira que disponibilizaram seu pouco tempo para dar atenção ao meu trabalho e pela contribuição para que o trabalho atingisse os níveis de qualidade esperado.

A Deus por ter me dado saúde, força para superar as dificuldades, e por me ajudar alcançar todos os objetivos que almejo.

Aos meus pais Narciso e Inês e meu irmão Cesar que apesar das dificuldades sempre ajudaram da melhor forma possível, dando todo que foi necessário para eu chegar até aqui como amor, incentivo e apoio incondicional.

A minha namorada Adrielle Centenaro que esteve presente durante todo o processo de formação, dando suporte e auxílio.

A todos meus amigos que fiz durante a graduação, em especial Diego, Marcelo, Israel e Elias pelos conselhos e principalmente pela nossa amizade.

Ao Centro Universitário Facvest, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, elevado pela acendrada confiança no mérito e ética presentes.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“O único sucesso existente é aquele que é adquirido de forma honesta sem prejudicar o próximo, e o segredo para alcançá-lo é ter força de vontade sem nunca desistir dos seus objetivos.” Aatoria Própria.

RESUMO

Atualmente os microcontroladores cumprem um papel fundamental, pois estão presentes em praticamente todos os equipamentos eletroeletrônicos, como por exemplo, em máquinas e dispositivos indústrias, sistemas de supervisão, veículos, brinquedos, eletrodomésticos, controle de processos e diversas outras aplicações.

Além disso, a variedade de microcontroladores disponíveis no mercado é ampla, causando dúvidas nos estudos de iniciantes, na área de desenvolvimento em projetos que envolvem microcontroladores. Muitas vezes, a escolha incorreta do microcontrolador pode causar dificuldades, podendo super ou subdimensionar o microcontrolador para uma determinada aplicação prática.

Microprocessadores, microcontroladores, processador de sinais digitais (DSP), e FPGA baseado em VLSI são diferentes plataformas digitais disponíveis para controle e pode ser efetivamente implementado algoritmos de controle em plataformas digitais, resultando em facilidade de integração com sistemas digitais. No entanto, as técnicas de controle baseadas no microcontrolador e no processador de sinais digitais (DSP) requerem um processamento sequencial. Todavia o recurso de processamento simultâneo está disponível em um controlador baseado em FPGA. Assim no decorrer do trabalho pretende-se demonstrar as diferenças de funcionalidade, operações físicas, construtivas, execução de tarefas, velocidade de processamento, entre outras características referentes ao Arduino Uno, PIC, ESP32, DSP e FPGA. Este último será aplicado no controle de conversores estáticos de potência (PANDIT, 2017).

Com base nos comentários anteriores, foi proposto neste trabalho apresentar vários tipos de microcontroladores, com o intuito de direcionar o leitor qual microcontrolador escolher em uma determinada aplicação. Por fim, com o intuito de exemplificar a aplicação dos microcontroladores em eletrônica de potência foi escolhido um FPGA para o estudo, análise e implementação do controle de tensão do Conversor CC-CC não isolado Boost.

Palavras-chave: Arduino, PIC, FPGA, ESP32 DSP e Eletrônica de Potência.

ABSTRACT

Currently, microcontrollers play a fundamental role, as they are present in virtually all electronic devices, such as in machines and devices industries, supervisory systems, vehicles, toys, appliances, process control and many other applications.

In addition, the variety of microcontrollers available in the market is broad, causing doubts in the studies of beginners, in the area of development in projects that involve microcontrollers. Often, incorrect choice of the microcontroller can cause difficulties, being able to super or undersize the microcontroller for a given practice application.

Microprocessors, microcontrollers, digital signal processor (DSP), and FPGA based VLSI are different digital platforms available for control and can be effectively implemented control algorithms on digital platforms, resulting in ease of integration with digital systems. However, control techniques based on the microcontroller and the digital signal processor (DSP) require sequential processing. In addition, the simultaneous processing feature is available in an FPGA-based controller. Thus, in the course of the work, we intend to demonstrate the differences in functionality, physical, constructive, task execution, processing speed, among other characteristics related to Arduino Uno, PIC, ESP32, DSP and FPGA. The latter will be applied in the control of static power converters (PANDIT, 2017).

Based on the previous comments, it was proposed in this work to present several types of microcontrollers, in order to direct the reader which microcontroller to choose in a given application. Finally, in order to exemplify the application of the microcontrollers in power electronics, an FPGA was chosen to study, analysis and implementation the voltage control of the non-isolated Boost DC-DC converter.

Keywords: Arduino, PIC, FPGA, ESP32 DSP and Power Electronics.

SUMÁRIO

LISTA DE FIGURAS.....	10
LISTA DE SIGLAS.....	12
LISTA DE ACRÓNIMOS	15
1. INTRODUÇÃO.....	14
1.1 Objetivos	15
1.1.1 Objetivo geral.....	15
1.1.2 Objetivos específicos	15
1.2 Justificativa.....	15
1.3 Aplicação.....	16
1.4 Metodologia	16
2. REVISÃO BIBLIOGRÁFICA DOS MICROCONTROLADORES.....	17
2.1 Introdução	17
2.2 Breve histórico dos microcontroladores	17
2.3 Microcontroladores – Princípio de funcionamento	18
2.3.1 Núcleo CISC e RISC.....	19
2.3.2 Arquitetura Von Neumann e Harvard	19
2.4 Plataforma Arduino Uno.....	20
2.4.1 Introdução	20
2.4.2 IDE do Arduino para desenvolvimento de projetos	21
2.4.3 Características físicas	22
2.4.4 Características funcionais.....	24
2.4.5 Considerações finais.....	24
2.5 PIC.....	26
2.5.1 Introdução	26
2.5.2 O PIC 18F4520	28
2.5.3 Aplicação.....	29
2.6 ESP32	30
2.6.1 Introdução	30
2.6.2 IDE para desenvolvimento de projetos no ESP32	33
2.6.3 <i>Clock e Timer</i>	34
2.6.4 Diagrama de blocos e funções.....	35
2.6.5 Exemplo de aplicação do ESP32.....	36
2.6.6 Considerações finais.....	38

2.7 DSP	39
2.7.1 Introdução	39
2.7.2 IDE para desenvolvimento de projetos.	41
2.7.3 Características físicas e estruturais.....	42
2.7.4 Aplicações	45
2.7.5 Filtros digitais.....	45
2.8 FPGA	47
2.8.1 Introdução	47
2.8.2 IDE software Quartus II.....	48
2.8.3 Modulação de pulso PWM.....	49
2.8.4 Aplicações	50
2.9 Comparação entre os microcontroladores	53
3. CONVERSOR CC-CC BOOST CLÁSSICO	56
3.1 Introdução	56
3.1.1 Aplicações	56
3.2 Princípio de funcionamento do conversor Boost	58
3.2.1 Princípio de funcionamento do conversor Boost operando em condução contínua	60
3.2.2 Simulação do conversor Boost em malha aberta:	62
3.3 Conversor Boost operando em Malha Fechada:	65
3.3.1 Simulação do conversor Boost em malha fechada:.....	69
4. RESULTADOS E DISCUSSÕES	72
4.1 Introdução	72
4.2 Desenvolvimento do protótipo	73
4.2.1 Especificação dos semicondutores	75
4.2.2 Especificação do capacitor.....	75
4.2.3 FPGA utilizado.....	75
4.3 Geração de PWM implementado no FPGA	76
4.4 Conversor Boost operando em malha aberta	78
4.5 Conversor Boost operando em malha fechada	79
4.6 Análise de rendimento do conversor Boost	83
5. CONCLUSÃO	85
REFERÊNCIAS	87
APÊNDICE A - ANÁLISE DE RENDIMENTO DO CONVERSOR BOOST	93
APÊNDICE B - PROJETO DOS PARÂMETROS DO CONVERSOR BOOST	101

APÊNDICE C - ESQUEMÁTICO DO CONVERTOR BOOST NO ALTIUM DESIGNER	107
APÊNDICE D – VISÃO GERAL DO PROGRAMA SINTETIZADO.....	109
APÊNDICE E – CIRCUITO DIGITAL SINTETIZADO.....	111

LISTA DE FIGURAS

FIGURA 1 -	Arquitetura Von Neumann, acessando a memória de dados.....	20
FIGURA 2 -	Arquitetura Harvard, acessando a memória de dados.	20
FIGURA 3 -	Arduino Uno.....	21
FIGURA 4 -	PIC 16F676	27
FIGURA 5 -	PIC 18F4520	29
FIGURA 6 -	Estrutura do sistema e memória do ESP32.	32
FIGURA 7 -	<i>Kit</i> ESP32	33
FIGURA 8 -	Diagrama de Blocos Funcionais.....	36
FIGURA 9 -	Resultados vistos através na tela do smartphone.	37
FIGURA 10 -	DSP - TMS320F28x.....	41
FIGURA 11 -	Composição de um DSP, representado via Diagrama de blocos.	43
FIGURA 12 -	Exemplo simples de aplicação de processamento de sinal	45
FIGURA 13 -	Filtro Digital, etapas de sua configuração.....	46
FIGURA 14 -	Estrutura de um FPGA.	48
FIGURA 15 -	Kit DE0-Nano	52
FIGURA 16 -	Imagem kit Max 10 utilizado para realização dos experimentos.....	52
FIGURA 17 -	Conversor CC-CC Boost clássico	58
FIGURA 18 -	Curva do ganho estática do conversor Boost.	60
FIGURA 19 -	Primeira etapa de operação do conversor em condução contínua.....	61
FIGURA 20 -	Segunda etapa de funcionamento do conversor em condução contínua.	62
FIGURA 21 -	Principais formas de onda do conversor Boost.	62
FIGURA 22 -	Esquemático conversor Boost utilizado na simulação, em MA.....	63
FIGURA 23 -	Principais formas de onda obtidas na simulação do conversor Boost.	64
FIGURA 24 -	Rendimento do conversor Boost simulado no PSIM.	65
FIGURA 25 -	Diagrama de blocos da malha de corrente interna à malha de tensão.....	65
FIGURA 26 -	FTLA da malha interna de corrente.	67
FIGURA 27 -	FTLA da malha externa de tensão.	68
FIGURA 28 -	Esquemático do conversor Boost, controle da corrente e tensão.	69
FIGURA 29 -	Esquemático conversor Boost utilizado na simulação, em MF.	70
FIGURA 30 -	Principais formas de onda obtidas na simulação do conversor em MF.	71
FIGURA 31 -	Imagem em 3D do conversor Boost, com $P_0 = 500 W$	74
FIGURA 32 -	Vista Lateral do protótipo do conversor Boost implementado, $P_0 = 500W$.	74

FIGURA 33 -	Tensão no interruptor (amarelo) e corrente no indutor (roxo)	78
FIGURA 34 -	Tensão de entrada V_{in} (amarelo) e tensão de saída V_o (azul).	79
FIGURA 35 -	Código implementado no controle de corrente do conversor.	80
FIGURA 36 -	Código implementado no controle de tensão do conversor.	81
FIGURA 37 -	Formas de onda experimentais do controle de tensão PI.	82
FIGURA 38 -	Corrente de entrada e tensão de saída com o controlador PI.	82
FIGURA 39 -	Tela do analisador de potência para os testes de rendimento.....	83
FIGURA 40 -	Rendimento experimental do protótipo do conversor Boost construído.....	84
FIGURA 41 -	Análise rendimento com 5% da potência nominal.....	94
FIGURA 42 -	Análise rendimento com 10% da potência nominal.....	94
FIGURA 43 -	Análise rendimento com 15% da potência nominal.....	94
FIGURA 44 -	Análise rendimento com 20% da potência nominal.....	95
FIGURA 45 -	Análise rendimento com 25% da potência nominal.....	95
FIGURA 46 -	Análise rendimento com 30% da potência nominal.....	95
FIGURA 47 -	Análise rendimento com 35% da potência nominal.....	96
FIGURA 48 -	Análise rendimento com 40% da potência nominal.....	96
FIGURA 49 -	Análise rendimento a 45% da potência nominal.....	96
FIGURA 50 -	Análise rendimento com 50% da potência nominal.....	97
FIGURA 51 -	Análise rendimento com 55% da potência nominal.....	97
FIGURA 52 -	Análise rendimento com 60% da potência nominal.....	97
FIGURA 53 -	Análise rendimento com 65% da potência nominal.....	98
FIGURA 54 -	Análise rendimento com 70% da potência nominal.....	98
FIGURA 55 -	Análise rendimento com 75% da potência nominal.....	98
FIGURA 56 -	Análise rendimento com 80% da potência nominal.....	99
FIGURA 57 -	Análise rendimento com 85% da potência nominal.....	99
FIGURA 58 -	Análise rendimento com 90% da potência nominal.....	99
FIGURA 59 -	Análise rendimento com 95% da potência nominal.....	100
FIGURA 60 -	Análise rendimento com 100% da potência nominal.....	100

LISTA DE SIGLAS

IEEE	- Instituto de Engenheiros Eletricistas e Eletrônicos
IoT	- Internet of Things (internet das coisas)
IDE	- Ambiente de Desenvolvimento Integrado
DSP	- Digital Signal Processor (processamento digital de sinal)
FPGA	- Field Programmable Gate Array, em português (Matriz de Portas Programáveis em Campo)
VLSI	- Very-large-scale integration (Integração em escala muito grande)
PWM	- Pulse-width modulation (modulação por largura de pulso)
kB	- Quilo Bytes
MHz	- Mega Hertz
USB	- Universal Serial Bus (barramento serial universal)
N/D	- Não aplicável
ICSP	- In Circuit Serial Programming (programação serial em circuito)
DC	- Direct current (Corrente contínua)
AD	- Analógico para Digital
E/S	- Entrada/saída
mA	- Miliampère
kb	- Quilo Bits
DAC	- Digital-to-Analog Converter (conversor digital-analógico)
ADC	- Analog-to-digital converter (conversor analógico para digital)
DSP	- Digital Signal Processor (Processador Digital de Sinais)
CI	- Circuito integrado
GSM	- Global System for Mobile Communications (Sistema Global para Comunicações Móveis)
PID	- Proporcional Integral Derivativa
PI	- Proporcional Integral
VLSI	- Very-large-scale integration (integração em muito larga escala)
FPGA	- Field Programmable Gate Arrays (Arranjo de Portas Programáveis em Campo)
VHSIC	- Very-High-Speed Integrated Circuit (circuito integrado de velocidade muito alta)
VHDL	- VHSIC Hardware Description Language (Linguagem de descrição de

	hardware VHSIC)
HDL	- Hardware Description Language (Linguagem de descrição de hardware)
SPI	- Serial Peripheral Interface
AMBA	- Aanced microcontroller bus architecture (arquitetura avançada do barramento do microcontrolador)
V_{in}	- Tensão de entrada
V_{ref}	- Tensão de referência
V_0	- Tensão de saída
APB	- AMBA peripheral bus (Barramento periférico AMBA)
LUT	- Look up table
DPWM	- Digital pulse-width modulation (modulação digital por largura de pulso)
DPFM	- Digital pulse-frequency modulator (modulação de frequência de pulso)
RTL	- Register Transfer Level (circuito síncrono digital)
MQTT	- Message Queue Telemetry Transport
SOC	- System On Chip (Sistema em um chip)
SD	- Secure Digital
IP	- Internet Protocol
WLAN	- Wireless Local Area Network (Rede Local Sem Fios)
RTC	- Real-Time <i>Clock</i>
CPU	- Central Processing Unit (Uidade Central de Processamento)
PLL	- Phase Locked Loop (Loop de bloqueio de fase)
DCF	- Distributed Control Function
P2P	- Peer-to-peer (ponto-a-ponto)
BR/EDR	- Bluetooth Basic Rate/Enhanced Data Rate
BLE	- Bluetooth Low Energy
dB	- Decibel
SDIO	- Secure Digital Input Output
MMC	- MultiMediaCard
UART	- Universal asynchronous receiver-transmitter
APIs	- Application Programming Interface (Interface de Programação de Aplicações)
TI	- Texas Instruments

RTDX	- Real Time Data Exchange (processamento em tempo real dos sinais)
ns	- Nano segundos
SIMD	- Single Instruction, Multiple Data (Única Instrução, Dados Múltiplos)
MMX	- MultiMedia eXtension or Multiple Math or Matrix Math eXtension
VLIW	- Very Long Instruction Word (Instrução de Palavra muito Grande)
DMA	- Direct memory access (Acesso Direto à Memória)
DTMF	- Dual-tone multi-frequency signaling (Sinalização multifrequencial de tom duplo)
MFC	- Mel-frequency cepstrum
GPIO	- General-purpose input/output (Entrada/Saída De Uso Geral)
f_s	- Frequência de chaveamento
PFC	- Power Factor Correction (Correção de fator de potência)
MA	- Malha aberta
MF	- Malha fechada

LISTA DE ACRÓNIMOS

RISC	- Reduced Instruction Set Computer (Computador com um conjunto reduzido de instruções)
CISC	- Complex Instruction Set Computer (Computador com um conjunto complexo de instruções)
PROM	- Programmable Read Only Memory (Memória Programável Somente de Leitura)
RAM	- Random Access Memory (Memória de Acesso Aleatório)
USART	- Universal Synchronous/Asynchronous Receiver/Transmitter (Receptor/Transmissor Assíncrono/Síncrono Universal)
Wi-Fi	- Wireless Fidelity (comunicação sem fio)
SRAM	- Static random access memory (memória estática de acesso aleatório)
EPROM	- Erasable programmable read-only memory (memória programável apagável somente de leitura)
ROM	- Read Only Memory (Memória Somente de Leitura)
PIC	- Programmable Interface Controller (Controlador de Interface Programável)
RAM	- Random Access Memory (memórias de acesso aleatório)
TCP	- Transmission Control Protocol (Protocolo de Controle de Transmissão)
MAC	- Media Access Control
ULA	- Unidades lógicas aritméticas
Máx	- Máximo
UPS	- Uninterruptible Power Supply (Fonte Ininterrupta de Energia)
MOSFET	- Metal-Oxide-Semiconductor Field Effect Transistor (Transistor de Efeito de Campo de Óxido de Metal Semicondutor)

1. INTRODUÇÃO

A busca por comodidade e segurança individual torna-se a cada dia mais relevante, e este conforto é proporcionado graças a automação de sistemas que antigamente eram executados através da mão-de-obra humana. Por sua vez, a automatização das máquinas proporcionou desenvolver tarefas com maior agilidade, no qual se pode substituir a mão de obra braçal em atividades perigosas, insalubres, e diversos outros procedimentos que podem ser desenvolvidos por uma máquina comandada por inteligência microcontrolada.

Os microcontroladores são capazes de realizar cálculos aritméticos através de amostras, e/ou dados recebidos, tomar decisões, também podem ser implantados em aplicações de inteligência artificial, internet das coisas (IoT), processamento de imagem e som e uma infinidade de aplicações, tornando-se um dos grandes responsáveis pela evolução tecnológica cotidiana. Com base nas tendências de expansão exponencial nos circuitos eletrônicos, podemos afirmar que é extremamente importante o estudo dos microcontroladores.

No mercado atual dispõe-se de uma grande variedade de microcontroladores, de diferentes fabricantes, que são diferenciados basicamente pela arquitetura e capacidade de processamento. Dentro desse contexto, o presente trabalho aborda os principais modelos de microcontroladores, assim como o FPGA. É importante ressaltar que o FPGA não é um microcontrolador, mas se trata de uma lógica programável, onde é possível sintetizar circuitos lógicos que podem ser processados em paralelo ou sequencialmente.

Finalmente, esse trabalho visa estudar as diferenças entre alguns tipos de microcontroladores aplicados em controle e soluções de engenharia, buscando esclarecer as diferenças entre eles e demonstrando a área de aplicação de cada dispositivo analisado. Além disso, será apresentada uma análise teórica e prática da aplicação do FPGA em conversores estáticos de potência, especificadamente no controle de tensão do conversor elevador Boost.

1.1 Objetivos

1.1.1 Objetivo geral

Este trabalho tem por objetivo geral estudar os diferentes modelos de microcontroladores, visando definir as características de cada microcontrolador de forma comparativa esclarecendo as diferenças e detalhando-os individualmente. Além disso, apresenta-se como objetivo do trabalho implementar na prática o FPGA, para a aplicação do controle de tensão do conversor Boost.

1.1.2 Objetivos específicos

Como objetivo específico será estudado as características de alguns tipos de microcontroladores. Dentre os quais foram escolhidos os seguintes modelos: o Arduino, o PIC, o ESP32, o DSP e o FPGA. Dentre as estruturas escolhidas, será realizado o estudo e análise comparativa com base na literatura vigente.

Também serão desenvolvidos o estudo e a análise do conversor CC-CC Boost elevador de tensão. E por fim o controlador escolhido será empregado para controlar o conversor estático de potência sendo implementando em laboratório, onde são apresentados resultados práticos tanto em malha aberta como em malha fechada.

1.2 Justificativa

Atualmente a forma mais empregada para controlar a maioria dos processos e sistemas é através de controladores digitais. Nas aplicações industriais, por exemplo, os microcontroladores não se encontram conectados diretamente em máquinas, entretanto estão integrados no interior de CLPs inversores de frequência e diversos outros dispositivos de controle. Portanto, os microcontroladores são a base de uma infinidade de equipamentos eletrônicos. Devido ao elevado grau de importância do microcontrolador, assim como sua vasta aplicação, tornam-se importantes pesquisas relacionadas sobre o tema.

1.3 Aplicação

A área de aplicação dos microcontroladores é extremamente vasta. Os Microcontroladores são encontrados em dispositivos e produtos automatizados, como os dispositivos médicos, sistemas de controle de automóvel, acionamento de máquinas elétricas, controle da geração de redes inteligentes, entre outros.

O conhecimento dos microcontroladores é fundamental, pois a maioria dos equipamentos eletrônicos depende de microcontroladores para automatizá-los. Também são muito utilizados para desenvolvimento de protótipos e estudos do desenvolvimento da automação.

1.4 Metodologia

A metodologia adotada para o desenvolvimento do trabalho será uma pesquisa qualitativa, com enfoque prático. O estudo baseia-se fundamentalmente nos manuais dos fabricantes dos modelos dos microcontroladores escolhidos, livros, teses, dissertações e principalmente em artigos provenientes da IEEE.

A primeira etapa consiste de uma revisão bibliográfica de cinco (5) microcontroladores escolhidos para serem estudados, descrevendo suas características gerais, áreas de aplicação e demais informações relevantes. As etapas seguintes referem-se respectivamente a análise comparativa entre os microcontroladores e a escolha de um deles para a implementação prática. Para a execução prática do FPGA escolhido será estudado o conversor CC-CC não isolado Boost. A partir do estudo teórico, análise matemática e resultados de simulação será desenvolvido o protótipo do conversor, acionado e controlado através do FPGA escolhido, apresentando por fim os resultados experimentais em malha aberta e malha fechada, assim como as discussões e a conclusão.

2. REVISÃO BIBLIOGRÁFICA DOS MICROCONTROLADORES

2.1 Introdução

Consideram-se os microcontroladores pastilhas inteligentes constituídas por um processador, pinos de entradas/saídas e memória. Por meio da programação dos mesmos podem-se definir suas saídas e entradas, baseado nas entradas ou em um programa interno (MARTINS, 2005).

Basicamente um microcontrolador incorpora no mesmo encapsulamento o microprocessador, a memória de programa e a memória de dados. Além disso, possui uma série de pinos com entradas e saídas e vários periféricos, resultando em um hardware final complexo. Normalmente a diferença entre estas características é o que diferencia os microcontroladores (MARTINS, 2005).

Dentre as funções de cada etapa que contém um microcontrolador podemos citar:

- Memória de programa – armazenamento permanente das instruções do programa;
- Microprocessador – processamento dos dados e interpretação das instruções do programa;
- Memória de dados – memoriza os resultados das operações do programa;
- Pinos de entradas e saídas – comunica o microcontrolador com o ambiente externo;
- Periféricos – comunicação, controle de interrupção, modulação PWM, entre outros.

2.2 Breve histórico dos microcontroladores

Os microcontroladores forma tomando forma como uma evolução natural dos circuitos digitais graças ao desenvolvimento constante da complexidade dos mesmos. Chegando a um ponto que é mais simples, mais barato e mais compacto, substituir a lógica

das portas digitais por um conjunto de processador e software (PENIDO; TRINDADE 2013).

Os microprocessadores começaram a ser empregados em computadores na por volta da década de 70 com a finalidade de melhorar a eficiência no processamento de dados. A Intel foi uma das primeiras a utilizar os microcontroladores. Só após, despertou-se o interesse no aprimoramento dos mecanismos de processamento de dados através desses tipos de dispositivos. Baseado na topologia de um microprocessador e seus periféricos foi criado um componente integrando todos os processos que envolvem o processamento de dados em uma única unidade, onde o sistema equivalente ao processamento e seus periféricos é chamado de CI (Circuito Integrado) (MARTINS, 2005).

Com o desenvolvimento da tecnologia na área, foi possível reduzir os custos de produção dos microcontroladores, tornando-os então uma das melhores opções na relação entre o investimento e o valor gasto no âmbito do processamento de dados, com alta densidade de processamento e baixo custo (MARTINS, 2005).

2.3 Microcontroladores – Princípio de funcionamento

De acordo com Penido; Trindade (2013) microcontrolador é um CI capaz de realizar operações lógicas e possui internamente os seguintes dispositivos:

- Uma CPU, na qual a finalidade é interpretar as instruções de programa;
- Uma memória PROM cuja finalidade é gravar as instruções do programa;
- Uma memória RAM (Random Access Memory ou Memória de Acesso Aleatório) empregada com a finalidade de memorizar as variáveis utilizadas pelo programa;
- Um barramento de I/O para controlar dispositivos externos ou receber impulsos de sensores, interruptores;
- Um conjunto de dispositivos auxiliares ao funcionamento, ou seja, gerador de *clock*, contadores, USART para comunicação.

2.3.1 Núcleo CISC e RISC

De acordo com Rocha (2012) deve-se considerar nos microcontroladores as suas arquiteturas RISC e CISC. A arquitetura CISC consiste no recebimento pelo processador de um grande volume de instruções de programa que o processador deve identificar. Para isso, a CPU possui uma unidade responsável por decifrar as instruções complexas transformando-as em uma série de instruções simplificadas para que a ULA possa executar. Já a utilização do núcleo CISC, evidencia-se quando a lentidão das memórias prejudica o acesso às instruções de programa. Porém, com o progresso da tecnologia de fabricação das memórias, o acesso a elas tornou-se mais ágil, fazendo com que o núcleo CISC não seja mais utilizado (ROCHA, 2012).

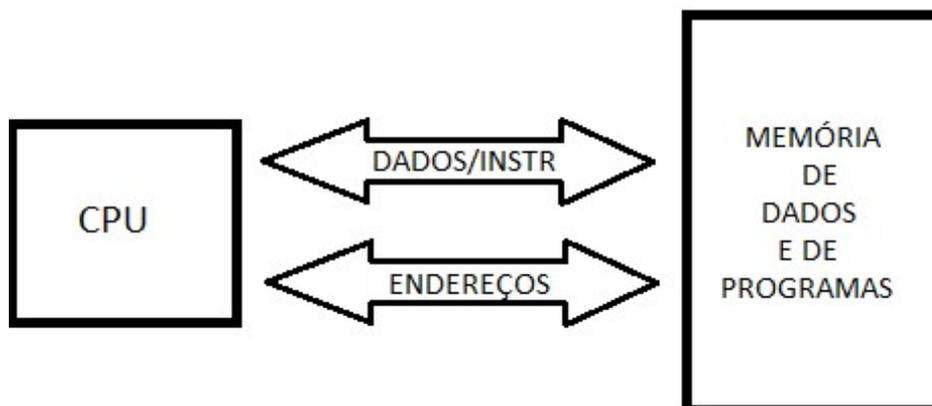
Basicamente o núcleo RISC consiste em um número reduzido de instruções, que são executadas diretamente pelo hardware do processador, assim as instruções são executadas rapidamente. Por fim, elas são carregadas da memória externa para registradores internos da CPU, melhorando assim a velocidade de processamento, por esse motivo, este sistema é o mais utilizado atualmente (ROCHA, 2012).

2.3.2 Arquitetura Von Neumann e Harvard

Segundo Rocha (2012), a disposição interna de um microcontrolador divide-se em duas arquiteturas diferentes, a arquitetura Harvard e Von Neumann.

John Von Neumann definiu a memória como sendo um dispositivo de armazenamento temporário, de forma que os dados e o programa são carregados a partir de um dispositivo de entrada, para posteriormente serem processados pela ULA. A unidade de controle deve processar o tráfego de dados. No modelo criado por Von Neumann os dados e o programa que será executado são guardados em um único dispositivo de memória conectado com um único barramento de dados e endereços, tornando impossível que a CPU acesse uma instrução e um dado ao mesmo tempo, como apresentado na Figura 1 (ROCHA, 2012).

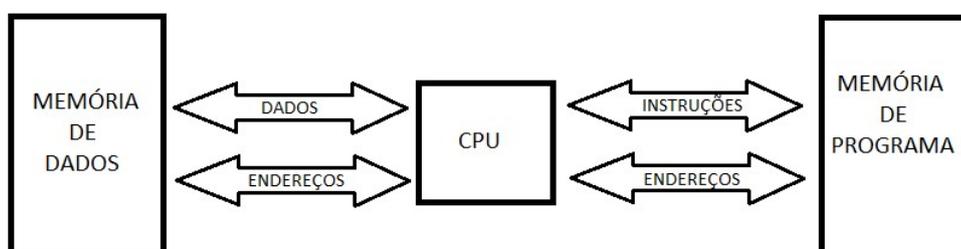
FIGURA 1 - Arquitetura Von Neumann, acessando a memória de dados.



Fonte: ROCHA (2012).

Em comparação com a arquitetura Von Neumann, a arquitetura Harvard apresenta memórias diferentes de dados e programa, sendo que cada uma contém seu próprio barramento de endereço e dados, conforme apresentado na Figura 2. Assim, uma instrução e um dado podem ser acessados de forma simultânea, no qual conseqüentemente amplia a capacidade de processamento (ROCHA, 2012).

FIGURA 2 - Arquitetura Harvard, acessando a memória de dados.



Fonte: ROCHA (2012).

2.4 Plataforma Arduino Uno

2.4.1 Introdução

O Arduino surgiu na Itália, recentemente em 2005, pelo professor chamado Massimo Banzi, visando ensinar de forma interativa eletrônica e programação de computadores para seus alunos de design, para que eles usassem em seus projetos de arte, interatividade e robótica. Porém, ensinar programação e eletrônica à pessoas que não são da área era uma tarefa difícil, e outra dificuldade era a inexistência de placas poderosas e baratas

no mercado. Pensando nisso, Massimo e David Cuartielles decidiram criar sua placa própria, com ajuda do aluno de Massimo, David Mellis, que ficou encarregado de desenvolver a linguagem de programação do Arduino. Várias pessoas conseguiram utilizar o Arduino e fazer diversas atividades de forma fácil e mais em conta sendo utilizada em atividades didáticas até os dias atuais (BEGHINI, 2013).

Na Figura 3 apresenta-se o Arduino UNO, que é a arquitetura escolhida para ser estudada neste trabalho. Também pode ser observado na Figura 3 o microcontrolador Atmega328P, pertencente a este modelo.

FIGURA 3 - Arduino Uno



Fonte: ARDUINO (2018)

2.4.2 IDE do Arduino para desenvolvimento de projetos

Baig Et al., (2014) assim como SOH Et al., (2017) proferem que o Arduino IDE é um software para desenvolver e compilar programas para o Arduino. O Arduino IDE também é utilizado para enviar programas compilados para a memória do programa da placa Arduino. O Arduino possui vantagens com relação a outros microcontroladores em termos de facilidade de uso e saídas lógicas disponíveis de fácil acesso. Além de possuir uma interface muito didática e simples que pode ser programada na linguagem C, o próprio fabricante testou o Arduino e desenvolveu uma grande variedade de módulos de comunicação. Outra grande característica do Arduino é o seu Serial Monitor em seu ambiente de desenvolvimento, possibilitando verificando as saídas desejadas e depurar nossos circuitos usando o Serial Monitor dessa forma facilitando o desenvolvimento e entendimento em aplicações que não exigem tanta complexibilidade. Podemos estabelecer condições para as saídas desejadas,

como desligar, ligar ou pulsar e o Monitor Serial do Arduino mostrará uma mensagem de confirmação e se o nosso dispositivo não responder, podemos fazer com que o nosso Monitor Serial mostre uma mensagem para isso também, assim facilita a identificação de erros no código facilitando a aprendizagem. Todas as funções podem ser acessadas facilmente através os ícones de atalho pois o hardware está configurado, basta apenas programar via software, agilizando assim a prototipagem. Pode ser feito gratuitamente o *download* do *software* no site oficial da Arduino, disponível em www.arduino.cc.

Percebendo o potencial e a abrangência de aplicações do hardware Arduino, fornecedores de software como *Matworks* ou *National Instruments*, incluíram os pacotes Arduino nos acessórios de software do MATLAB e LABView (SILVA Et al., 2015).

2.4.3 Características físicas

De acordo com Raju (2017), Baig Et al., (2014), Kamisan Et al., (2017) e Andriansyah Et al., (2017), o Arduino UNO se trata de uma plataforma de computação física de código aberto desenvolvido em uma simples PCB com 20 E/S (entrada/saída). Seu dispositivo microcontrolador baseada no Atmega328 possui 6 entradas analógicas e 14 pinos digitais podendo ser tanto entrada como saída digital, dentre esses 14 pinos, 6 deles podem ser usados para saídas PWM, além disso ele possui 32 kB de memória flash, um ressonador cerâmico com frequência 16Mhz (na maioria dos casos), uma conexão USB, um conector de energia, um conector ICSP e um botão de reset. É possível conectar-se facilmente a um computador com um cabo USB para poder enviar receber dados, carregar programas, e operá-los. A Tabela 1 apresenta algumas características Arduino uno.

TABELA 1 - Especificações Arduino UNO

Descrição	Atribuição
Tensão de operação	5 V
Tensão de alimentação aconselhável	7 – 12 V
Tensão de alimentação limite	6 – 20 V
Número de pinos digitais E/S	14 (6 pinos PWM)
Número de pinos de entrada analógica	6 pinos
Corrente CC E/S	40 mA
Corrente CC no pino 3.3 V	50 mA
memória Flash	32 kb
SRAM	2 kb
Cristal oscilador	16 MHz
Microcontrolador	ATmega 328

Fonte: ANDRIANSYAH Et al., (2017).

O software Arduino usado para programar o Arduino UNO, contém no ambiente de desenvolvimento diversas bibliotecas de módulos prontas para utilizar, assim dispensando que o usuário tenha que desenvolver uma programação específica para cada módulo. Para acessar as bibliotecas é necessário ir na barra de ferramentas do ambiente de desenvolvimento clicar em “*Sketch*” em seguida “incluir biblioteca” assim irá abrir uma lista de bibliotecas bastando apenas escolher a biblioteca adequada para a aplicação em questão. Essas bibliotecas são necessárias para programar e fazer o hardware do módulo operar corretamente via Arduino UNO com base na codificação que é programada por este software, assim o usuário pode desenvolver a lógica de programação sem precisar entender a funcionalidade dos módulos (KAMISAN Et al., 2017).

Embora o Arduino geralmente seja usado para pequenas aplicações amadoras e educacionais, seu conversor analógico para digital integrado pode ser usado para captura e reprodução de som, possibilitando aplicações que empregam maior complexibilidade. Equipado com um poderoso microcontrolador AVR (microcontrolador RISC de chip único com uma arquitetura Harvard e sigla sem significado definido) de 8 bits e um ADC que pode

alcançar taxas de amostragem de até 200 kHz com uma resolução de 10 bits, de acordo com a ficha técnica ATmega328P da Atmel (SILVA Et al., 2015).

2.4.4 Características funcionais

O Arduino apresenta algumas capacidades de processar som, porém possui limitações específicas da plataforma Arduino com relação a qualidade, sua interface pode ser conectada e instalada no software MATLAB. Podemos encontrar uma série de exemplos da interface do Arduino com o MATLAB que são detalhados e mostrados para facilitar o acesso dos usuários do MATLAB e do Arduino. O Atmega328P da Atmel é o núcleo de processamento do Arduino e pode ser programada separadamente. Seu microcontrolador AVR, é equipado com um ADC (conversor analógico para digital) de 10 bits. Um sinal analógico pode ser lido ou amostrado certo número de vezes por segundo, que é referido como frequência de amostragem (fs) que pode ser definida via programação (SILVA Et al., 2015).

Existem diferentes tipos de arquiteturas ADC, mas a maioria dos AVR de 8 bits a 16 bits usa ADC de aproximação sucessiva (modelo de ADC relativamente pequeno com baixo consumo de energia e com alta resolução e precisão). O AVR tem apenas um ADC de 10 bits e usa um multiplexador analógico de 8 canais para amostrar cada um dos pinos analógicos. O AVR tem um *Clock* dedicado que garante a independência da conversão do circuito ADC de outras partes do microcontrolador. O mecanismo de conversão pode ser acionado sob demanda ou automaticamente. As amostras do ADC são armazenadas em dois registradores, o ADC alto e baixo. Para obter os resultados em digital é preciso multiplicar 1023 pela tensão de entrada analógica em seguida dividir pela tensão de referência que no caso é 5 Volts. Por exemplo: se a tensão de entrada (V_{in}) for de 2,5 Volts, o valor convertido será 512 e, se $V_{in} = V_{REF}$ (tensão de referência), o resultado será 1023. O Arduino usa um comando simples para iniciar uma única conversão, o “*analogRead* (pino número)” (onde é feito a leitura analógica de determinado pino). Se nada mais for executado, a conversão estará completa (SILVA Et al., 2015).

2.4.5 Considerações finais

Deve-se ressaltar aqui que existem diversas famílias de Arduino, sendo que o modelo a ser escolhido depende da necessidade, por exemplo, em processos que requerem

altas taxas de processamento uma opção é o Arduino Due que é cerca de 15 vezes mais rápido que o Arduino Uno e possui 2 portas DAC além das PWM normais. Além disso, ele tem um ADC de 12 bits em vez dos atuais 10 bits do Uno, mas infelizmente o custo é o dobro da plataforma Arduino Uno. No entanto, esse preço ainda é muito menor do que as plataformas DSP tradicionais. Tendo isso em vista pode-se dizer que o Arduino uno é uma opção barata para aplicações de desenvolvimento que não exijam tanta exatidão e confiabilidade, sendo ideal para iniciantes e na área educacional (SILVA Et al., 2015).

Por fim, com o intuito de apresentar a variedade de outros modelos da família Arduino, o Quadro 1 mostra os principais tipos de Arduino e suas principais especificações:

QUADRO 1 - Família Arduino:

	UNO	MEGA 2560	LEONARDO	DUE	ADK	NANO	PRO MINI
Portas digitais	14	54	20	54	54	14	14
Portas PWM	6	15	7	12	15	6	6
Portas analógicas	6	16	12	12	16	8	8
Memória	32k	256k	32k	512k	256k	16k	16k
Clock	16MHz	16MHz	16MHz	84MHz	16MHz	16MHz	8MHz 16MHz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial/Módulo USB externo
Conector/alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não
Tensão de operação	5 V	5 V	5 V	3,3 V	5 V	5 V	3,3 ou 5 V
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130mA	40 mA	40 mA	40 mA
Alimentação	7-12 V	7-12 V	7-12 V	7-12 V	7-12 V	7-12 V	3,3-12 V ou 5-12 (dependendo do modelo)

Fonte: FILIPEFLOP (2018).

2.5 PIC

2.5.1 Introdução

De acordo com Rocha (2012) os microcontroladores PIC fabricados pela Microchip são produzidos utilizando a arquitetura Harvard. O PIC possui diversidade de opções, sendo assim possível definir o microcontrolador que se adequa melhor ao projeto a ser desenvolvido. Os microcontroladores PIC's possuem 8 bits, 16 bits e 32 bits. Para aplicações de baixo custo são empregados os PIC's de 8 bits além de ser mais comum no mercado sendo mais fáceis de encontrar e por atender a maioria das necessidades. Para tratamento de ponto flutuante e números muito grandes o mais aconselhável é a utilização de registradores de 16 e 32 bits, mas também para estes casos pode ser utilizado registradores de 8 bits, onde pode-se empregar determinados métodos de programação que possibilitam simular registradores maiores. Os modelos de 8 bits são encontrados na família 10 F, 12 F, 16 F e 18 F. A linha de menor custo é a 10 F, pois é composta por poucos periféricos de tamanho relativamente pequeno contendo 6 terminais. Já a linha 12 F possui encapsulamento com 8 terminais e na família 16 F é diferenciada fisicamente e pode-se encontrar encapsulamentos de diversos tamanhos com intervalo de 14 a 64 terminais. Além disso, estas duas últimas linhas possuem uma enorme quantidade de periféricos integrados, como conversores AD, referência de tensão e módulo PWM. A família 18 F, uma evolução da família 16 F, porém empregam-se aprimoramentos em sua unidade de processamento, com um *set* de instruções maior e oferecendo maior quantidade de periféricos, como ethernet e interface USB (ROCHA, 2012).

A popularidade dos microcontroladores PIC deu-se através do MPLAB IDE se tratando de uma ferramenta de auxílio à desenvolvimento de programas. Além do mais os microcontroladores PIC possuem uma linguagem de programação *Assembly* mais simples comparado a àquelas que outros fabricantes disponibilizam (MARTINS, 2005).

Em concordância com Noor Et al., (2014) o Microcontrolador PIC da Microchip é o microcontrolador eletrônico mais conhecido do mundo por oferecer disponibilidade multitarefa e uma grande quantidade de informação na folha de dados. Destaca-se por ser um CI de baixo custo de aquisição, motivo pela qual foi amplamente aplicado. É caracterizado por ser considerado um microcontrolador flexível para executar tarefas variadas em termos de sensores e tecnologia de automação e controle (PURI; NAYYAR, 2016) como é o caso de automação residencial (BAIG Et al., 2014) e alimentadores automáticos de animais.

Conforme explica Baig Et al., (2014) o PIC foi uma solução simples e uma das maneiras mais fáceis e baratas de implementar em automação residencial em conjunto com Módulos GSM é possível incluir mensagem de resposta (*feedback*) e aceita diversas outras formas de integração, trata-se de um dispositivo pequeno para esse tipo de aplicação. Porém podemos notar que nos últimos anos a facilidade e abrangência ao acesso à internet está gerando redução na aplicação da tecnologia GSM, além do mais as plataformas com módulos já inclusos possuem acesso à internet e *bluetooth* como por exemplo, o ESP32, dessa forma, diminuindo o número de conexões externas, gerando maior confiabilidade e consequentemente diminui os custos de instalação, manutenção e de operação. Portanto atualmente o PIC é mais empregado no controle de dispositivos periféricos, com intuito de aliviar a carga da CPU principal (KALE; BANKAR; JAGTAP 2014).

Ganguly Et al., (2017) aplica o PIC 16F676 para controlar distorções harmônicas em inversor de fonte de tensão monofásica objetivando menores perdas de chaveamento e a remoção de harmônicas. O dispositivo empregado trata-se de um microcontrolador CMOS de 8 bits baseado em flash com 14 pinos. Suas principais características são: oscilador de 0-20 MHz para entrada de *clock*, 12 pinos de I/O, ADC de 10 bits, um temporizador de 8 bits, um comparador, memória de programa de 1024 *words* (1 *word* equivale a 16 bits), SRAM de 64 bytes e 128 bytes EEPROM como memória de dados, programação serial em circuito através de dois pinos. Possui funções de captura/comparação/PWM, a porta serial síncrona pode ser configurada como SPI de 3 fios ou USART (PURI; NAYYAR, 2016). O PIC16FS77A 18f266 há 8 kB de memória e 368 bytes de RAM. Possui 40 pinos, 256 bytes de EEPROM, autoprogramação, CDI e 2 conversores ADC. A Figura 4 apresenta o microcontrolador PIC 16F676.

FIGURA 4 - PIC 16F676



Fonte: Indiamart.

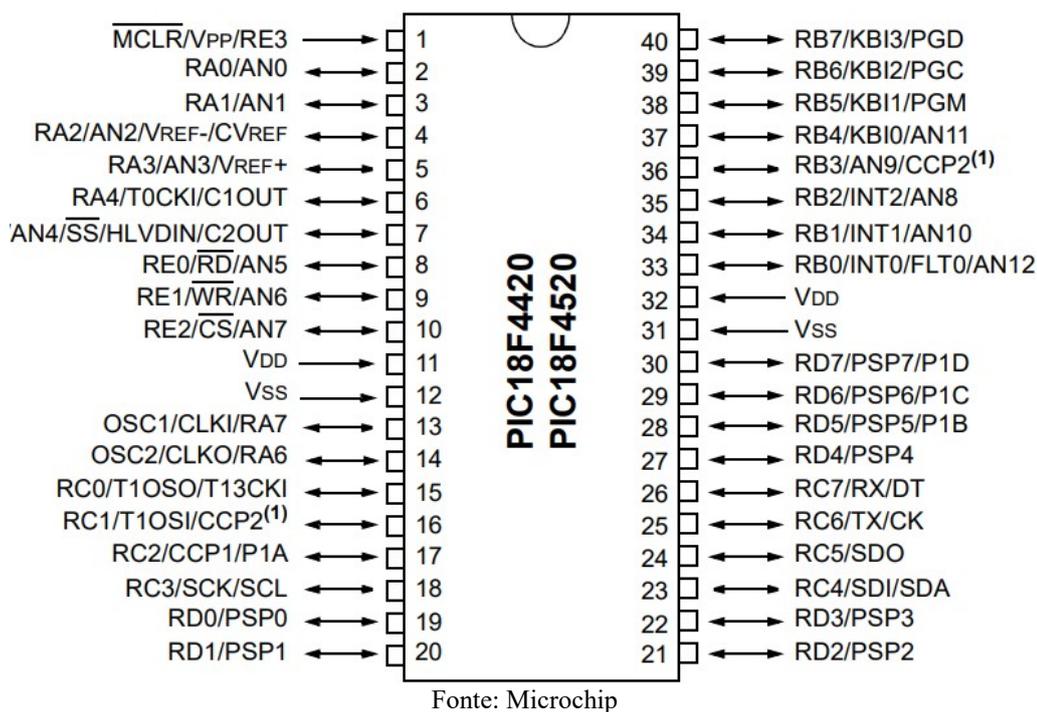
2.5.2 O PIC 18F4520

Foi escolhido para abordar como exemplo neste trabalho o PIC 18F4520, no qual segundo Rocha (2012) é um dos microcontroladores mais modernos da família PIC. O microcontrolador em questão possui uma grande quantidade de periféricos integrados, são suas principais características são apresentadas:

- Possui um microcontrolador com 40 pinos, viabilizando a montagem de um hardware capaz de interagir com diversos recursos e funções simultaneamente;
- Possui 75 instruções, sendo 83 instruções no caso do *set* de instruções estendido;
- Possui 36 portas E/S configuráveis;
- Possui 20 interrupções possíveis;
- Possui memória de programação EEPROM FLASH;
- Possui 32 kbytes de memória de programa, com capacidade de leitura e escrita através do código interno;
- Possui memória interna EEPROM com 256 bytes;
- Possui memória RAM contendo 1536 bytes;
- Possui quatro *timers* (1x8 bits e 3x16 bits);
- Possui comunicações seriais: I²C SPI, e USART;
- Possui conversores analógicos de 10 bits (13x) e comparadores analógicos (2x);
- Possui programação *in-circuit* (alta e baixa tensão);
- Possui *Power-on Reset* (POR) interno;
- Possui *Brown-out Reset* (BOR) interno;
- Possui Grande variedade de recursos adicionais.

O microcontrolador PIC, modelo 18F4520 contém 40 pinos conforme apresentado na Figura 5, sendo que 36 deles são pinos de entrada e saída, sendo que o terminal RE3 (terminal 1) pode ser definido somente como entrada e os demais podem ser definidos como saída ou entrada de dados. Os pinos possuem várias funções multiplexadas correspondente à seus periféricos (ROCHA 2012).

FIGURA 5 - PIC 18F4520



2.5.3 Aplicação

Os microcontroladores PIC podem ser usados em diversas áreas, e podem ser escolhidos através do modelo mais adequado para cada aplicação, como por exemplo, em controle de motores controlado com PIC16F887 e controle robôs PIC16F877A utilizado por (RAJKANNA; MATHANKUMAR; GUNASEKARAN, 2014) comunicação GSM com o PIC16F877A em (MAKWANA; TANDON, 2016). Além disso, podem ser empregados em sistemas que exigem maior complexidade como em terapias que utilizam a técnica de estimulação elétrica nervosa transcutânea, onde o microcontrolador gerará o sinal de pulso único de onda de choque para obter o mesmo efeito terapêutico da terapia tradicional. Podem também ser empregados em controle digital PID em sistema de calefação, como citado por (RUGE, 2009) além e inúmeras outras aplicações.

Conforme Kale (2014) utiliza-se o PIC16F877A para aquisição de dados em motores e outras aplicações de sistemas de controle industrial. Geralmente utiliza-se informações completas sobre o *Data Acquisition System* usando um controlador PIC.

2.6 ESP32

2.6.1 Introdução

Normalmente esse modelo tem sido amplamente utilizado para desenvolver a IoT (internet das coisas) e expandir a área de suas aplicações. Para isso é necessárias soluções poderosas de baixo custo, eficiência e possuir um tamanho reduzido. Cada unidade baseada em IoT compreende um microcontrolador e um módulo de comutação sem fio (geralmente WiFi), ou uma combinação de ambos. Uma grande variedade de módulos e microcontroladores já estão no mercado e são amplamente utilizados para o design e desenvolvimento de dispositivos IoT. Porém, a maioria dos dispositivos oferecidos atualmente possui custo elevado ou são grandes e pesados. Além do mais, poucos módulos são dispositivos de código aberto e não têm restrições no propósito da operação. Embasado nisso, o microcontrolador ESP32, lançado no mercado pela *Espressif Systems* em setembro de 2016 substituto do anterior ESP ESPEC promete ser uma ótima alternativa para aplicação IoT. Seu baixo custo e excelente performance tem chamado atenção de adeptos (ALLAFI; IQBAL, 2017).

O recente produto ESP32 da *Espressif Systems*, projetado para a IoT e projetos relacionados a sistemas embarcados, é um sistema de baixo custo e baixo consumo de energia. Seu *chip* inovador em um único dispositivo com recursos de Wi-Fi e Bluetooth e uma estrutura altamente integrada com um microprocessador Tensilica Xtensa LX6 de arquitetura Harvard de núcleo duplo (MAIER; SHARP; VAGAPOV, 2017), está tendendo no mercado pelo seu baixo custo se comparado a outros microcontroladores que possuem características semelhantes (RAJALAKSHMI; SHAHNASSER, 2017).

Em comparação ao ESP32 pode-se citar o Raspberry Pi. Este último é um microcomputador com capacidades computacionais complexas. Ele é executado na plataforma Linux. O Node-Red, uma ferramenta de programação, permite o acesso direto dos pinos GPIO do Raspberry Pi e esses dados podem ser enviados diretamente para a nuvem. O Raspberry Pi geralmente é empregado em processos que exigem altas taxas de processamento, porém, outras placas em miniatura com custo menor e funcionalidades semelhantes, como por exemplo o ESP8266, (versão anterior do ESP32), e ESP32, estes podem ser conectadas a sensores e atuadores e colocadas em locais com acessibilidade Wi-Fi, de tal modo as entradas do sensor chegam à nuvem usando um protocolo chamado MQTT. Em diversos casos é

possível substituir o Raspberry Pi pelo ESP32 devido suas semelhanças e pelo tamanho reduzido (RAJALAKSHMI; SHAHNASSER, 2017).

Portanto, os dados carregados para a nuvem podem ser baixados em um celular, conectando-se a um servidor local através do Wi-Fi local. Além do mais, o ESP32 é um microcontrolador de código aberto baseado em uma placa de entrada/saída. Ele também possui 36 GPIO, 14 dos quais são Conversor Analógico para Digital (ADC) que podem ser conectados aos sensores. Quando 3,3 volts são aplicados a uma das portas ADC, o valor lido é de 4095, ou seja, seu ADC possui 12 bits de resolução. Esse valor é chamado de tensão de referência para o ESP32. Dessa forma, o ESP32 contém pinos ISP utilizados para conectar o ESP32 com um leitor de cartão SD. O intervalo VCC fornecido ao ESP32 é de 2,2V a 3,6V. O ESP32 se conecta a um micro USB para carregar o programa e a alimentação, ou pode ser conectado a uma bateria de 3,7 V. Para melhor entendimento, algumas características do ESP32 estão apresentadas na tabela 2 (ALLAFI; IQBAL, 2017).

TABELA 2 - Especificações referentes ao ESP32.

Descrição	Atribuição
Tensão de operação	2,2 V até 3,6 V
Portas programáveis	36 portas
ADC	14 portas
DAC	2 portas
Memória Flash	16 Mbytes
SRAM	250 kbytes
<i>Clock</i>	Até 240 MHz
Wi-Fi	2,4 GHz
Corrente modo Sleep	2,5 uA

Fonte: Allafi; Iqbal, (2017).

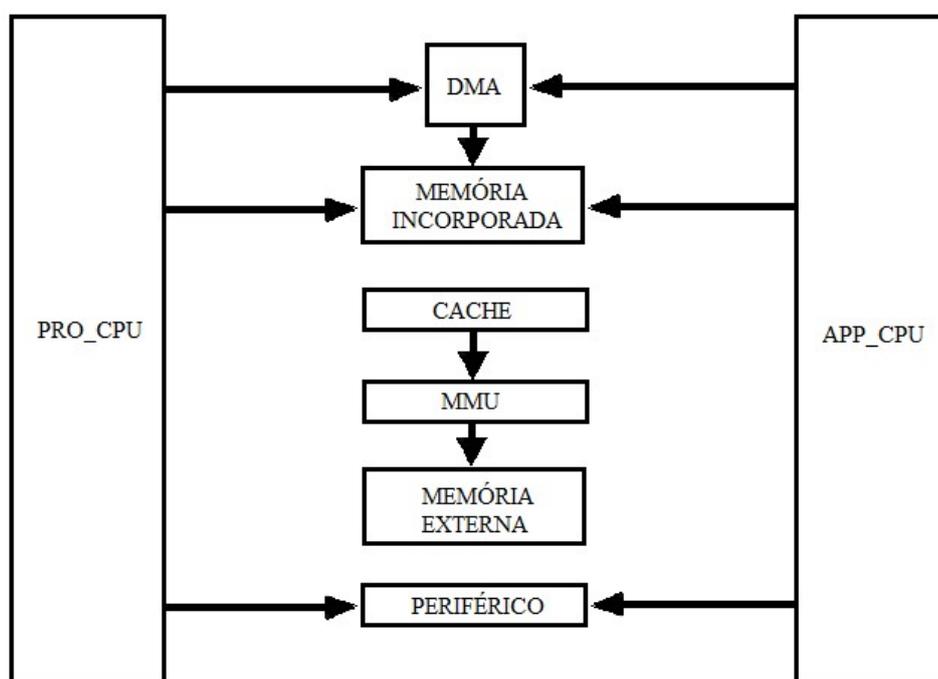
Considera-se ainda que os módulos externos que possuem tensão de operação de 3,3 V podem ser conectados diretamente nos pinos SPI. Também não há a necessidade de conectar os resistores de *pull-up* entre os pinos de um ESP32 para equilibrar a tensão ou proteger o cartão SD contra danos. Além disso, a frequência do transmissor ou receptor é de

cerca de 2,4 GHz, o que é adequado para um servidor web para baixar dados de um cartão SD usando qualquer celular ou tablet com Wi-Fi (ALLAFI; IQBAL, 2017).

Considerando que a tensão de saída de muitos sensores não é padronizada, existem sensores com saída de 0 V a 3 V e outros com níveis de 0 V a 5 V. Nos casos em que a tensão de saída do sensor for entre 0 V e 5 V, isto é, não são compatíveis com as portas do ADC do ESP32. De maneira a resolver este problema, um resistor de *pull-down* é colocado entre a saída do sensor e a entrada do ESP32. O propósito do resistor é diminuir a tensão máxima de 5 V à 3,3 V no máximo (ALLAFI; IQBAL, 2017).

Na Figura 6 apresenta-se a estrutura do sistema e memória do ESP32. O mesmo possui um sistema dual-core de arquitetura Harvard com dois processadores Xtensa LX6. Toda a memória incorporada, memória externa e periférica estão localizados no barramento de dados e/ou no barramento de instruções dessas CPUs. O microcontrolador possui dois núcleos - PRO_CPU para protocolo e APP_CPU para aplicação, porém, os propósitos desses são diversos. O espaço de endereçamento para os dados e o barramento de instruções é de 4 GB e o espaço de endereçamento periférico é de 51 kB. Além disso, as memórias incorporadas são 448 kB ROM, 520 kB SRAM e duas memórias 8 kB RTC, e a memória externa suporta até quatro vezes 16 MB de Flash (MAIER; SHARP; VAGAPOV, 2017).

FIGURA 6 - Estrutura do sistema e memória do ESP32.



Fonte: MAIER; SHARP; VAGAPOV (2017).

Apesar do tamanho relativamente pequeno (25,5 x 18,0 x 2,8 mm) como pode constatar na Figura 7, o ESP32 é um dispositivo de fácil manuseio devido a componentes integrados como oscilador, antena e flash. Diferenciando-se de outras plataformas que necessitam de módulos externos gerando assim um aumento de tamanho e diminuição da confiabilidade nas conexões, como por exemplo se comparado com o Arduino Uno. Módulos semelhantes para outros microcontroladores são frequentemente usados para testes e protótipos ou por amadores. Portanto, há uma vasta variedade de módulos e microcontroladores para IoT, porém a maioria deles tem os mesmos problemas relacionados ao tamanho, desempenho e preço. Caso haja necessidade de módulos externo o circuito compacto do módulo ESP-WROOM-32, facilita a integração do ESP32 a uma PCB de forma a projetar um dispositivo com otimização de espaço. O *kit* ESP32-DevKitC é uma solução pronta para uso e amigável, para testes e propósitos educacionais. O ESP8266, predecessor do ESP32, era extremamente popular em muitos projetos relacionados à IoT, no entanto, o ESP32 é uma alternativa melhorada que pode ser aplicada em projetos mais complexos (ALLAFI; IQBAL, 2017).

FIGURA 7 - *Kit* ESP32



Fonte: ETT

2.6.2 IDE para desenvolvimento de projetos no ESP32

O sistema operacional do ESP32 é o *FreeRTOS* (um sistema operacional de tempo real). Possui código aberto, projetado para sistemas embarcados e fornece funções básicas para aplicativos de alto nível. As principais funções são gerenciamento de memória,

gerenciamento de tarefas e sincronização de APIs. A forma mais usual de programar o ESP32 é usando o *framework* de desenvolvimento ESP-IDF IoT do *Espressif Systems*, (plataforma de desenvolvimento oficial da ESPRESSIF), que está disponível em seu repositório GitHub (que é uma plataforma de hospedagem de código-fonte que permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou de código aberto). O ESP-IDF foi desenvolvido para o Linux, assim, é necessário um terminal Linux para executar os arquivos *bash*. No entanto, é possível desenvolver no Windows usando o MSYS2, sendo que este software disponibiliza um terminal Linux no Windows. Além disso, o ESP-IDF-model é uma ferramenta necessária para iniciar um projeto ESP32, onde inclui todos os arquivos necessários para uma compilação bem-sucedida, que fazem parte de um projeto individual e não estão incluídos no ESP-IDF. O ESP-IDF fornece um manual de configuração visual acessível pelo comando “*make menuconfig*”, sendo o único manual gráfico. Todas as outras operações, como compilar ou verificar, são executadas através de comandos simples como no IDE Arduino. Portanto, o Eclipse IDE de software livre fornece excelente suporte para o projeto *Makefile* (denominação de determinado tipo de arquivo). Um projeto deve ser configurado para usar o compilador *xtensa-esp32-elf-gcc* e referenciar o ESP-IDF para habilitar recursos de depuração, que são essenciais para o desenvolvimento adequado do programa. O Arduino IDE também pode ser usado para gravar o código e comunicar com a placa, basta baixar o pacote compatível com o ESP32, porém por se tratar de uma tecnologia recente, algumas bibliotecas são incompatíveis e necessitam de ajustes para operar corretamente. (ALLAFI; IQBAL, 2017).

A linguagem mais usual para programar o ESP32 é C, assim a maioria das bibliotecas de APIs também é fornecida em C. No entanto, o microcontrolador também pode ser facilmente programado em C ++. Algumas bibliotecas do Arduino podem ser usadas na opção de programação C ++, embora pode ser necessário algumas mudanças. Neil Kolban, engenheiro do Texas, oferece muitas bibliotecas C ++ em seu repositório GitHub para as APIs do ESP32. Como este chip é de código aberto, todos podem desenvolver um “sistema operacional” para o ESP32, assim, também há soluções na Internet para programá-lo em LUA e JavaScript (ALLAFI; IQBAL, 2017).

2.6.3 *Clock e Timer*

O ESP32 pode usar o PLL interno de 320 MHz ou um cristal externo. Também é possível usar um circuito oscilante como uma fonte de *clock* a 2-40 MHz para gerar o *clock*

mestre CPU_CLK para ambos os núcleos da CPU. Este *clock* pode chegar a 160 MHz para aplicações de alto desempenho ou menor para reduzir o consumo de energia. Todos os outros *clock*, como o APB_CLK para periféricos, são acionados pelo oscilador principal. Além disso, há vários osciladores de baixa potência, como o RTC_CLK interno, com uma frequência padrão de 150 kHz e a opção de ajustá-lo para modos de economia de energia. Existem quatro temporizadores de 64 bits para propósitos genéricos com *prescalers* de 16 bits com um intervalo de 2 a 65536. Cada *timer* usa o *clock* do APB, geralmente a 80 MHz. Esses temporizadores podem ser crescentes ou decrescentes, ser congelado e acionar eventos. Além disso, há 4 temporizadores genéricos e também temporizadores para acionar o controlador PWM. Além de 8 canais PWM de baixa velocidade existem canais 8 de alta velocidade, cada um acionado por quatro *timers* (ALLAFI; IQBAL, 2017).

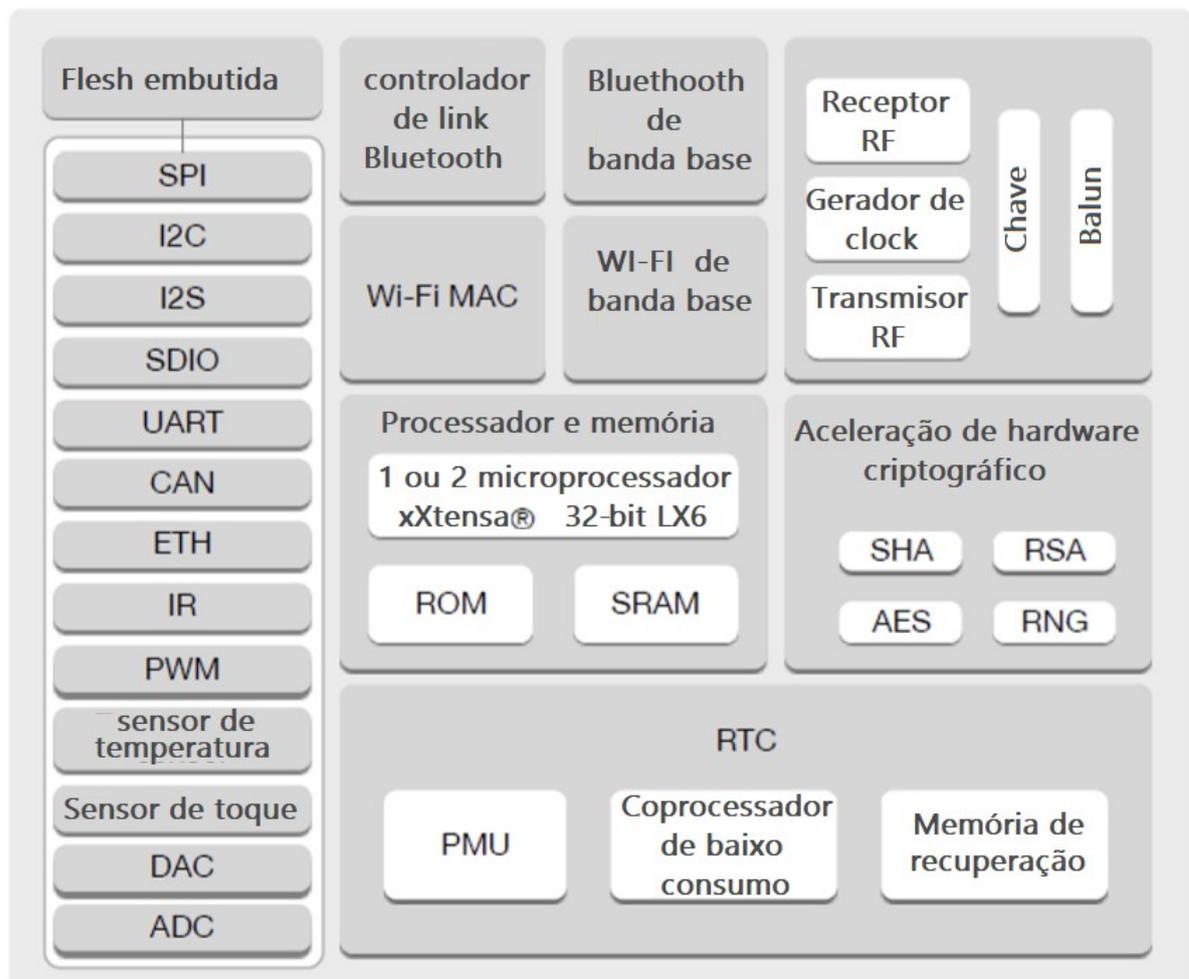
2.6.4 Diagrama de blocos e funções

A estrutura do microcontrolador ESP32 foi projetada para operar sob os seguintes protocolos - TCP/IP, WLAN 802.11 b/g/n/e/i completo e especificação Wi-Fi *Direct*. O microcontrolador pode oferecer operações Conjunto de Serviço Básico sob o protocolo DCF. Também é compatível com a operação do grupo P2P e compatível com o mais recente protocolo Wi-Fi P2P. Assim, ele pode operar como uma estação, estar conectado à internet ou servidor, e ponto de acesso, com a finalidade de fornecer uma interface de usuário, por exemplo, o smartphone executando um aplicativo móvel (ALLAFI; IQBAL, 2017). A Figura 8 ilustra o diagrama de blocos funcionais.

Além disso o microcontrolador possui capacidade de suportar versão 4.2 do BR/EDR e BLE, o Bluetooth que se ajusta ao padrão atual sendo capaz de operar a uma velocidade de até 4 Mbps. O ESP32 pode trabalhar sob diversas configurações de energia como modo ativo (o chip de rádio está funcionando), e modo de inatividade (a CPU está totalmente operacional, mas o Wi-Fi e o *Bluetooth* estão desligados). Além disso, existem modos de sono profundo (modo *sleep*), em que a CPU opera com desempenho inferior, para isso é diminuído a frequência de *clock*. As portas configuráveis são compostas por dois ADCs de 12 bits com 18 canais, sendo possível configura-los para resoluções de 9 bits, 10 bits e 12 bits com uma atenuação de - 0 dB, - 6 dB ou - 11 dB para diferentes intervalos de entrada. Um canal ADC é conectado ao sensor de *hall* integrado para detectar campos magnéticos e outro para o sensor de temperatura com faixa de - 40 °C a 125 °C dedicado ao monitoramento da temperatura do *chip*. Além dos ADCs, há também dois DACs de 8 bits para converter os

sinais digitais em saídas de sinal de tensão analógica. Dez das portas configuráveis são sensíveis a variações capacitivas, podem ser usados para sensores de toque e possui alta sensibilidade. Além disso, o ESP32 fornece diversas interfaces: uma Interface MAC Ethernet, três interfaces UART até 5 Mbps, um Controlador Host SD/SDIO/MMC, duas interfaces I2C com uma frequência de 10 kHz até 10 MHz, duas interfaces de barramento I2C com modo padrão rápido, um controlador remoto infravermelho de 8 canais e um contador de pulsos de 8 canais. O controlador PWM pode ser usado para controlar conversores, acionar motores digitais ou gerar formas de onda digitais. Três SPIs podem ser usados no modo mestre escravo com um *clock* de até 80 MHz (ALLAFI; IQBAL, 2017).

FIGURA 8 - Diagrama de Blocos Funcionais



Fonte: Allafi; Iqbal, (2017).

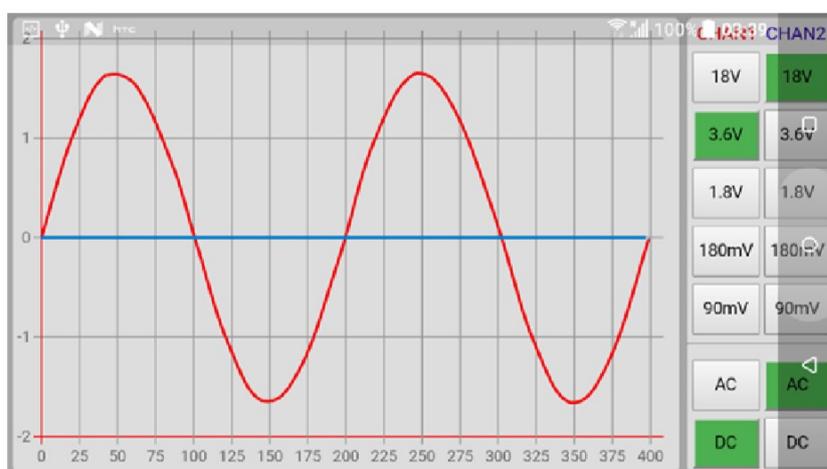
2.6.5 Exemplo de aplicação do ESP32

Segundo Maier (2017) o ESP32 pode ser aplicado em projetos comum de IoT,

como controlar soquetes, luzes remotamente e construir um osciloscópio baseado em smartphone. Maier (2017) apresenta de um protótipo de um osciloscópio sem fio portátil baseado no ESP-WROOM-32 como o núcleo de hardware e um aplicativo de smartphone como a unidade de exibição e controle.

Os resultados do osciloscópio sem fio são vistos através do aplicativo na tela do *smartphone* e é ilustrado na Figura 9.

FIGURA 9 - Resultados vistos através na tela do smartphone.



Fonte: MAIER; SHARP; VAGAPOV (2017).

Maier (2017) testa o experimentalmente as entradas analógicas do ESP32, sendo um ponto fundamental quando aplicado a controle em malha fechada de conversores. O ADC integrado foi submetido a testes para verificar a precisão da conversão. Foi aplicada uma tensão de 0,0 V a 3,5 V à entrada do ADC em intervalos de 0,1 V e os valores medidos foram registrados. Este teste foi repetido para diferentes configurações, o melhor resultado foi entregue em resolução de 12 bits com uma atenuação de -6 dB. Através desses testes foi possível constatar que o intervalo de entrada do ADC não começa em 0,0 V, mas em 0,17 V. Assim, optou-se por utilizar o intervalo de 0,2 V a 1,9 V para este projeto. Além disso, a frequência média foi de aproximadamente 44,5 kHz com resolução de 10 bits e 12 bits. Foi ligeiramente superior a uma resolução de 9 bits, aproximadamente 45,9 kHz, porém esse ganho de frequência não foi significativo a ponto de levar à perda de resolução. Isto deve ser levado em consideração para o projeto da instrumentação de medição de um conversor estático, pois a atenuação de melhor desempenho é limitada em aproximadamente 2 V em quanto a atenuação de 12 dB é capaz de suportar leitura de até 3 V, ou seja, digitalmente igual a 4096.

Apesar do ESP32 possuir sensor de efeito *hall*, bons resultados aos testes e tamanho reduzido como vantagens para aplicação em conversores estáticos, não se pode dizer que é uma das melhores opções de microcontroladores para projetos de alta confiabilidade, pois se trata de uma tecnologia nova, dessa forma se desconhece a sua confiabilidade no geral.

2.6.6 Considerações finais

O ESP32 é sistema novo no mercado, sendo uma excelente opção devido às propriedades de desempenho e preço, assim pode-se empregá-lo em projetos que vão do nível básico a intermediário. O microcontrolador está disponível em vários fatores de forma. A versão mais amigável ESP32-DevKitC é uma solução perfeita para fins amadores e educativos, o módulo ESP-WROOM-32 fornece uma aparência atraente de solda pequena enquanto que o ESP32 QFN48 é a opção para manufaturas industriais e soluções de pequeno porte (ALLAFI; IQBAL, 2017).

O excelente desempenho do microcontrolador é alcançado devido à estrutura dual core e uma extensão significativa dos recursos operacionais. O sistema operacional do microcontrolador *FreeRTOS* é um software de código aberto que oferece um ótimo suporte para aplicações em tempo real. Se destacando entre os microcontroladores de baixo custo, por possuir Wi-Fi, Bluetooth e dois núcleos de processamento em um único dispositivo compacto, atendendo a grande maioria das aplicações, um pouco mais caro que o Arduino, porém é facilmente compensado pelas vantagens. Assim, espera-se que o ESP32 desempenhe um papel importante no design de futuros sistemas de IoT e projetos incorporados (ALLAFI; IQBAL, 2017).

Por fim, para elucidar este microcontrolador o Quadro 2 faz uma comparação entre alguns microcontroladores que se assemelham com ESP32.

QUADRO 2 - Comparação entre alguns microcontroladores.

Chip (Módulo)	ESP32 (ESP-WROOM-32)	ESP8266 (ESP8266-12E)	CC32 (CC3220MODSF)	Xbee (XB2B-WFPS-001)
Detalhes:				
CPU	Tensilica Xtensa LX6 32 bit Dual-Core e 160/240 MHz	Tensilica Xtensa LX106 32 bit e 80 MHz (máx 160 MHz)	ARM Cortex- M4 e 80 MHz	N/A
SRAM	520 KB	36 KB acessível	256 KB	N/A
FLASH	2MB (máx. 64MB)	4 MB (máx. 16MB)	1MB (máx. 32MB)	N/A
Tensão	2,2V à 3,6V	3,3V à 3,6 V	2,3V à 3,6 V	3,14 V à 3,46 V
Corrente de operação	80 mA average	80 mA average	N/A	N/A
Programação	Free (C, C++, Lua, etc.)	Free (C, C++, Lua, etc.)	C (SimpleLink SDK)	AT e API comandos
Código aberto	Sim	Sim	Não	Não
Conectividade:				
Wi-Fi	802,11 b/g/n	802,11 b/g/n	802,11 b/g/n	802,11 b/g/n
Bluetooth	4.2 BR/EDR + BLE	-	-	-
UART	3	2	2	1
E/S:				
GPIO	32	17	21	10
SPI	4	2	1	1
I2C	2	1	1	-
PWM	8	-	6	-
ADC	18 (12-bit)	1 (10-bit)	4 (12-bit)	4 (12-bit)
DAC	2 (8-bit)	-	-	-
Tamanho				
	25,5 x 18,0 x 2,8 mm	24,0 x 16,0 x 3,0 mm	20,5 x 17,5 x 2,5 mm	24,0 x 22,0 x 3,0 mm

Fonte: MAIER; SHARP; VAGAPOV (2017).

2.7 DSP

2.7.1 Introdução

De acordo com Brito de Lima (2011) o DSP trata-se de um processador de sinais digital sendo composto por conversor analógico para digital, conversor digital para analógico. Seu alto desempenho aliado ao baixo custo são seus principais atributos que o torna muito atrativo e o principal concorrente do FPGA. Conforme Brito de Lima (2011) e Cardoso (2006), as características do DSP permitem a implementação de algoritmos de controle (fuzzy, adaptativo, robusto, clássico, *sensorless* etc) com flexibilidade de alteração dos parâmetros de controle via *software* dispensando necessidade de parâmetros físicos como, por exemplo, filtros dessa maneira os custos e tamanho de *hardware* decrescem.

O DSP é um tipo de microprocessador otimizado para processos aritméticos, já os demais processadores são otimizados para gerenciamento e manipulação de dados. Essas características o tornam apto a operar em aplicações que não admitem atrasos significativos

no processo de aquisição, processamento e transferência de dados (HOLDEFER, 2004). Sendo assim a estrutura interna do DSP é fundamentada em arquitetura *Harvard* modificada com barramentos internos de programa e dados independentes, permitindo acesso simultâneo a instruções de dados e programas. Os DSP's podem ser classificados pelo formato de dados como ponto flutuante ou ponto fixo. No caso de processador de pontos flutuantes são caracterizados os números pelo expoente e a mantissa. Já o processador de ponto fixo é caracterizado por palavras com tamanho fixo (8 bits, 16 bits). A definição do formato numérico depende do custo e as restrições sinal ruído requeridas para a aplicação almejada (BRITO de LIMA, 2011).

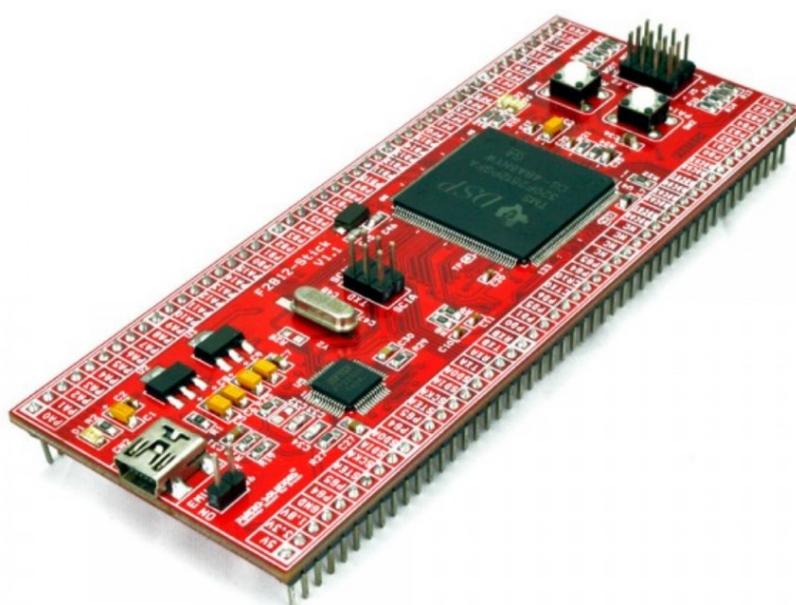
De acordo com Wolter (2017), o MAC 4 foi o primeiro DSP lançado no mercado no ano de 1979, produzido pela Bell Labs, devido a necessidade de processadores dedicado a processamento digital de sinal. Sua popularidade se deu com o lançamento do TMS32010 pela *Texas Instruments* em 1983, devido sua ferramenta RTDX. O mesmo é baseado na arquitetura Harvard, porém os dados de instrução são separados dos dados de memória. Além disso, possuía um conjunto de funções como instruções carregar e acumular ou multiplicar e acumular. Operava com números de 16 bits e necessitava apenas de 390 ns para calcular uma operação de multiplicação. Desde então a *Texas Instruments* é líder no mercado de DSP's.

Após a Motorola também se tornou importante no mercado de desenvolvimento de DSP's com o lançamento do Motorola 56000. Aproximadamente cinco anos mais tarde surgiu a segunda geração de DSP's. Com três memórias para armazenar simultaneamente dois cálculos também possuíam hardware para acelerar sistemas em malha fechada e alguns já trabalhavam com 24 bits. Na segunda geração, as operações como multiplicar-acumular durava aproximadamente 21 ns. O DSP's da terceira geração se destacou pelo aumento da capacidade de processamento, permitindo assim a solução de problemas matemáticos complexos e específicos como operações com matrizes e transformada de Fourier. Alguns chips desenvolvidos por volta de 1995 incluíam mais de um processador operando em paralelo. A quarta geração é marcada pelas instruções de codificar/decodificar. Além disso, foram adicionadas as extensões MMX VLIW e SIMD e surgiram as arquiteturas superescalar. Nessa geração a velocidade de *clock* aumentou e uma única operação de acumular e multiplicar passou a ser feita em 3 ns. Atualmente os processadores de sinais digitais possuem um desempenho muito maior graças ao avanço constante da tecnologia e arquitetura com regras de designe, *fast-access two-level cache*, circuito DMA e um sistema *bus* maior (diversos blocos compartilham uma única linha de dados). É importante ressaltar que nem

todos os DSP's possui a mesma velocidade e cada aplicação requer um determinado DSP. Hoje, um C600 da *Texas Instruments*, contém 6 *clocks* e 1 GHz sendo capaz de executar até 8000 MIPS (milhões de instruções por segundo). Atualmente os principais fabricantes de DPS são a *Motorola*, a *Analog Devices*, e a *Texas Instruments*, (WOLTER, 2017).

Na Figura 10 está apresentado o kit DSP TMS320F28x da Texas Instruments.

FIGURA 10 - DSP - TMS320F28x



Fonte: PANTECHSOLUTIONS (2018)

2.7.2 IDE para desenvolvimento de projetos.

Se tratando de um dispositivo programável, o DSP requer um IDE específico para o desenvolvimento dos códigos e programação do DSP. Considerando que a maioria dos DSP's trabalha com linguagem de programação C, C++ e Assembly, juntamente com o software IDE de desenvolvimento, os fabricantes fornecem as funções pré-definidas para cada linguagem que é feita para facilitar a implementação nos DSPs. O IDE disponibiliza o ambiente de trabalho necessário para desenvolvimento de um projeto. Este se trata de um compilador que gerencia todas as execuções, depurações e simulações do código. Como exemplo, a Texas Instruments que possui o IDE Code Composer Studio (CCP), e a Analog possui o IDE VisualDSP++, (WOLTER, 2017).

Primordialmente, considera-se que a realização de um projeto envolve três etapas básicas, sendo estas a pesquisa, a simulação e a emulação. De forma sucinta a pesquisa trata-se basicamente da percepção do problema, onde é definido se há necessidade de cálculo com

ponto flutuante ou ponto fixo. A etapa de simulação é caracterizada pela modelagem do projeto, onde é realizada a compilação do código, testes e as modificações necessárias. Finalmente, são efetuados os testes do projeto no seu ambiente de uso, onde é feita a análise detalhada e é corrigido os erros do programa, visando de forma a adequar o equipamento da melhor forma possível para atender as necessidades de projeto e de forma que não apresente falhas. O DSP possui a vantagem perante outros processadores, que é a possibilidade de melhorias, sem que haja a troca de equipamentos. Pode-se modificar a qualquer momento o projeto através da interface JTAG (Joint Test Action Group) (WOLTER, 2017).

2.7.3 Características físicas e estruturais.

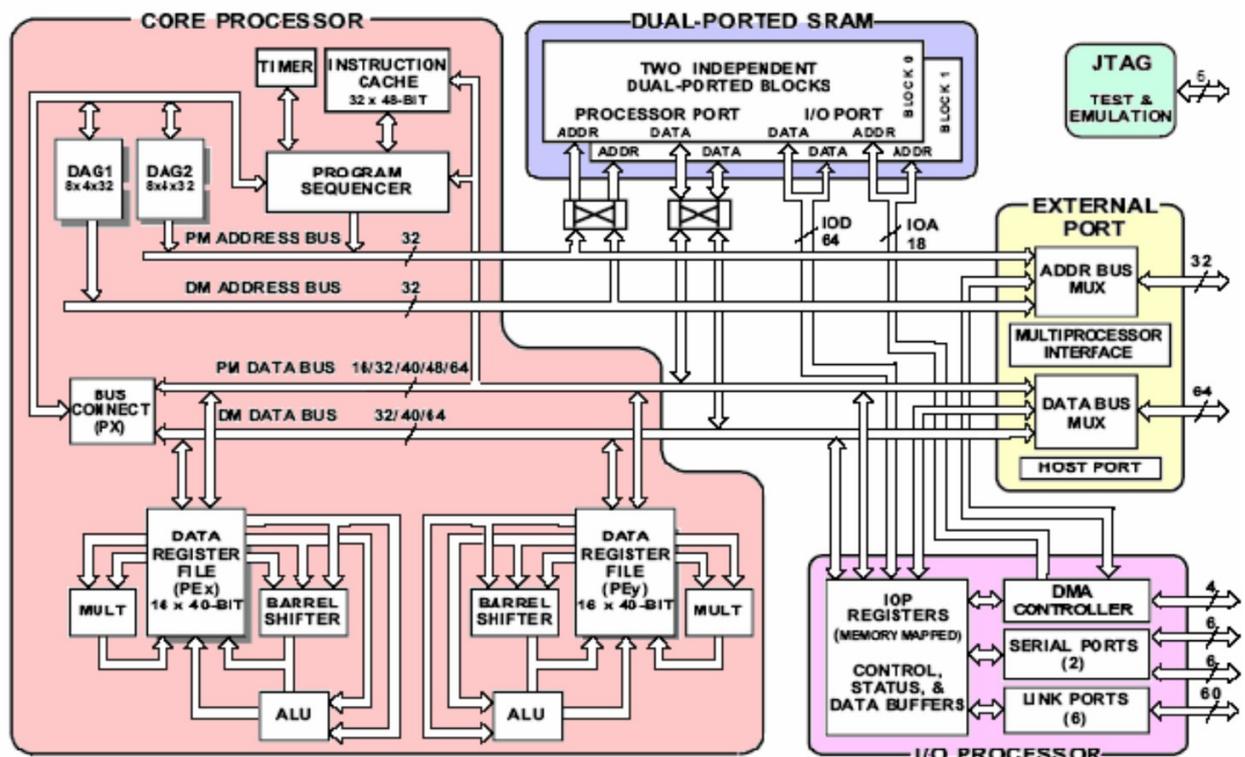
Graças ao desenvolvimento dos novos e poderosos algoritmos e a evolução das tecnologias de fabricação dos circuitos tem possibilitado o processamento em tempo real nos DSPs. Para que os dados sejam processados em tempo real e necessário que os sinais sejam processados de forma contínua e instantânea, a partir do momento que chegam ao DSP. Como exemplo, na aplicação em conversores estáticos, o sinal analógico deve ser imediatamente convertido para digital e processado instantaneamente, por fim deve ser aplicado na atividade em questão com o mínimo de perda de tempo possível (HOLDEFER, 2004).

Além de possuírem processamento em tempo real, os DSPs possuem características próprias. Quando comparados à microcontroladores genéricos possuem velocidade de processamento muito superior. A fim de aperfeiçoar e dinamizar os programas alguns DSP possuem a capacidade de se conectar e interagir com outros softwares como, por exemplo, o DSP TMS320C64x+ da Texas Instruments detêm uma interface de transferência de dados bidirecional com o programa Matlab. O DSP é um dispositivo programável que possui seu próprio código de instruções. Assim, cada fabricante de processador cria seu próprio IDE para seu modelo de chip, dessa forma a manipulação do microprocessador torna-se muito mais simples e rápida. Os DSP's são preparados para atender as necessidades mais comuns de operação, para isso eles possuem o recurso de instruções de repetição, que possibilita a execução de operações adição, multiplicação e transferências de memória subsequentes em apenas um único ciclo de memória na maioria das vezes. Operações como a transformada rápida de Fourier, permite a implementação de filtros digitais que é feita de forma fácil pela capacidade que o DSP tem de repetir (com uma instrução prévia, por exemplo, "RPT") em extrema velocidade uma instrução complexa (como por exemplo, a "MPYA": operação de multiplicação acumulada "*Multiply and Accumulate Previous Product*") (WOLTER, 2017).

Um exemplo de aplicação do DSP importante foi no emprego de cancelamento de ruídos. Para tal operação é captado o ruído ambiente e é gerado um “antirruído” com as ondas simétricas e sentido oposto, sendo a cada vale captado é gerado um pico e também reciprocamente. Esse fenômeno é conhecido como interferência destrutiva e pode ser usado como, por exemplo, no interior de automóveis e salas de aula. Este processo, coleta, processamento e geração de sinal é detalhado no item 2.7.4 (WOLTER, 2017).

Ao se estudar o processamento digital de sinal analisa-se o DSP, assim como as ferramentas que trabalham juntamente com o processador. São recursos como ULA's, registradores, geradores de endereço e memória. O conjunto desses mecanismos com chip DSP na placa também pode ser simplesmente chamado de DSP. No diagrama da Figura 11 está representada a composição de uma placa DSP. Em sua maioria os DSP possuem a mesma arquitetura, sendo constituída por cinco blocos principais: a porta SRAM, o processador principal, o emulador JTAG, a porta para comunicação externa e um processador do tipo entradas e saídas, além de seus componentes internos (WOLTER, 2017).

FIGURA 11 - Composição de um DSP, representado via Diagrama de blocos.



Fonte: WOLTER (2017).

Analisando o núcleo *core processor* encontram-se importantes subgrupos. Como os geradores de endereço imediato entre os registradores e a memória, estes são conhecidos

como geradores de endereço (DGA). As ULA's (unidade lógica aritmética) são responsáveis por realizarem operações lógicas e aritméticas, estas podem ser em ponto flutuante ou ponto fixo. Geralmente as placas de DSP contém um *timer* programável, usados para gerar interrupções periódicas, como também podem ativar funções ou rotinas a cada novo ciclo. As unidades de DMA (Direct Memory Access) tem a capacidade de operar independentemente e de forma imperceptível ao processador principal podendo ser empregado na realização de outras rotinas quando o processador principal estiver ocupado. Além disso, é importante para a comunicação entre as portas seriais ou *link ports* e a memória interna (WOLTER, 2017).

Antes de especificar um DSP deve ser considerado alguns itens importantes como os tipos de sinais a serem monitorados pelo DSP, frequência de operação do conversor, circuito de condicionamento de sinal, resolução, tempo de conversão analógica para digital, números de entradas ou saídas analógicas, período de execução de instruções, capacidade de memória e arquitetura básica (BRITO de LIMA, 2011), portanto é apresentado algumas características importantes do DSP TMS320F28379D na Tabela 3.

TABELA 3 - Especificações DSP TMS320F28379D

Descrição	Atribuição
CPU	2 x C28 x e 2 x CLA
Frequência (MHz)	200
Flash (KB)	1024
RAM (KB)	204
Resolução ADC	16 bits/12 bits
DAC	3
Saídas PWM	24
PWM de alta resolução	16
SPI	3
UART	4
I2C	2
Entradas ADC	4
E/S	97 - 169

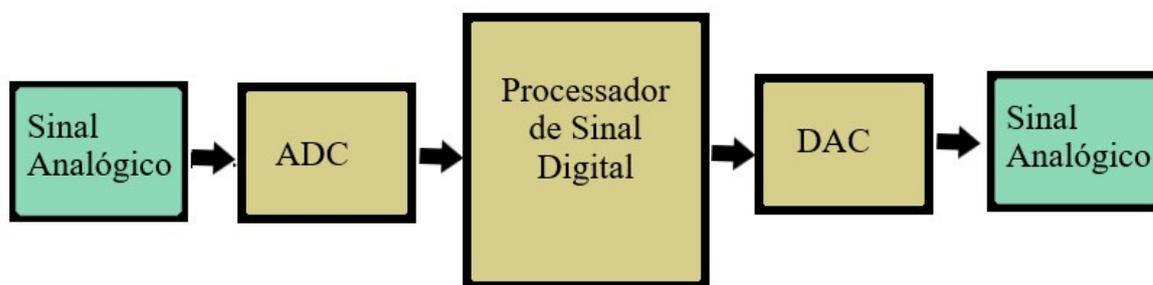
Fonte: DSP TMS320F28379D

2.7.4 Aplicações

Os DSP podem ser empregados em inúmeras aplicações, como por exemplo, na área militar, médica, industrial, telecomunicações, espacial, científica e comercial (BRITO de LIMA, 2011) em uma infinidade de componentes como computadores multimídia, celulares, controladores de disco rígido, modems, gravadores de vídeo e televisores (WOLTER, 2017).

Por possuir uma alta velocidade de processamento os DSP se popularizaram na eletrônica de consumo, como em aparelhos que sintetizam os sons de diversos instrumentos, como órgão de tubos, teclado o piano e o violão. Existem dispositivos DSP específicos preparados para processamento de áudio e vídeo, para isso possuem integrado codificador/decodificador que se encarregam de fazer as conversões necessárias. A Figura 12 ilustra o processamento de recebimento de dados e saída de dados em aplicação "antirruído" (WOLTER, 2017).

FIGURA 12 - Exemplo simples de aplicação de processamento de sinal



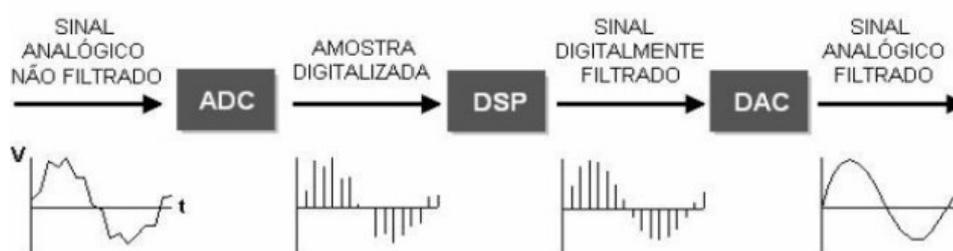
Fonte: WOLTER (2017).

2.7.5 Filtros digitais

De acordo com (WOLTER, 2017), os filtros se dividem em analógico e digital e tem a finalidade de remover sinais indesejáveis, como ruídos ou apenas extrair partes importantes do mesmo, como por exemplo, componentes de determinada banda de frequência. Nos filtros analógicos utilizam-se circuitos eletrônicos analógicos feitos basicamente de capacitores, resistores e amplificadores operacionais. Já nos filtros digitais como é demonstrado um esquema de funcionamento na Figura 13, é ajustado via programação. Muitos destes filtros são amplamente empregados em aplicações como: equalização gráfica em sistemas Wi-Fi, diminuição de ruído, realce de sinal de vídeo entre outros. Os filtros digitais apresentam grande precisão. Estes filtros fazem uso de um processador digital para realizar cálculos numéricos com base em amostras de sinais que são amostrados por meio de

entrada de um sinal analógico, onde, um ADC realiza a conversão deste sinal para ser lido pelo DSP. Após, processa-se os dados e realizam-se os cálculos para fazer a filtragem, através de operações de soma subtração multiplicação e divisão que são definidos via programação. Além disso, também se pode utilizar técnicas matemáticas complexas, como Hilbert e Transformadas de Fourier. Em situações que se necessite dos valores que representam o sinal analógico filtrado, podem ser convertidos para uso através de um DAC.

FIGURA 13 - Filtro Digital, etapas de sua configuração.



Fonte: WOLTER (2017).

Conforme Delfino (2005) e Wolter (2017) os filtros digitais possuem inúmeras vantagens que os tornam muitos mais versáteis e compactos no processamento de sinais se comparado aos filtros analógicos tais como:

- Facilidade de adaptação automática a características de sinais distintos;
- Facilidade de controlar sistemas complexos multiníveis lineares ou não lineares compostos por diversos sensores e atuadores;
- Maior estabilidade devida sua Imunidade a interferências causadas por variação de temperatura e desgaste dos componentes;
- Facilidade de implementação de métodos avançados tais como otimização, identificação, geração de estimativas, adaptação e adequação a variações em tempo real;
- Formulação de algoritmos de controle diretamente em código computacional ao invés da implementação em *hardwares* analógicos;
- Tolerância a grandes escalas de trabalho em sinais de baixa frequência;
- Possibilidade de definir pontos de operação pré-definidos;
- Desenvolvimento de projeto mais ágil e barato;

- Maior facilidade de manutenção e monitoramento.

Dessa forma, tem-se otimizado a de filtragens digitais em alta frequência, principalmente em ondas de rádio, no qual anteriormente eram utilizados os filtros analógicos. Obtendo assim melhora na qualidade dos sinais através de equipamentos mais versáteis (WOLTER, 2017).

2.8 FPGA

2.8.1 Introdução

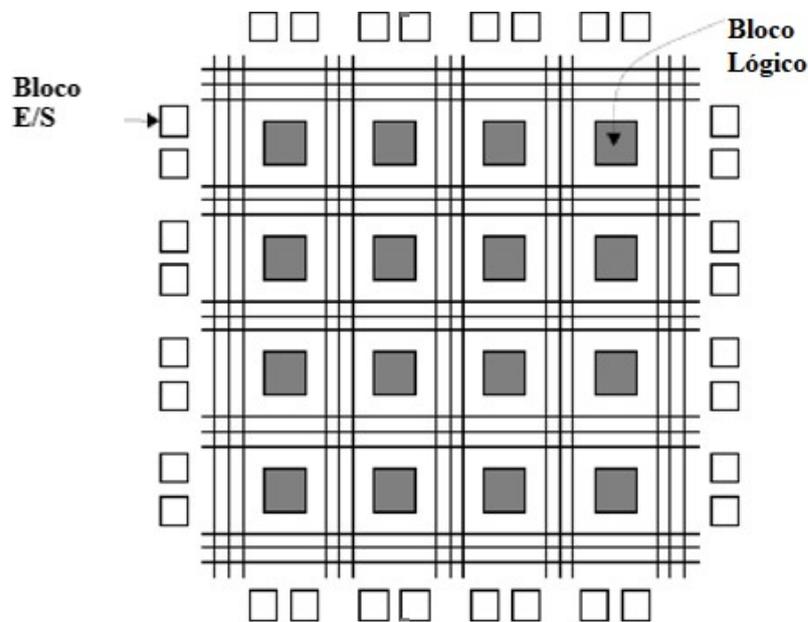
Considerando que os algoritmos de controle podem ser efetivamente implementados em plataformas digitais, obtém-se a facilidade de integração com sistemas digitais. Plataformas digitais são compostas por controladores que podem receber informações, processar e disponibiliza-las de acordo com a necessidade. Microcontroladores, microprocessadores, processador de sinais digitais (DSP), redes neurais, lógica difusa e FPGA baseado em VLSI são diferentes plataformas digitais disponíveis para controle. Uma desvantagem, das técnicas de controle baseadas no microcontrolador e DSP é a necessidade de processamento sequencial. Assim, para a utilização de um recurso de processamento simultâneo, dispõe-se de um controlador baseado em FPGA (PANDIT, 2017), (JAIN; AWARE; JUNGHARE, 2016).

Os FPGAs podem ser programados para ter uma estrutura paralela que o torna mais rápido, também pode ser configurado para seguir lógicas sequenciais, oferecendo uma funcionalidade para aplicações complexas e consumindo menos energia. Eles são facilmente programados e podem ser testados em tempo real e monitorado através de sua IDE. Além disso, possuem vantagens como precisão, capacidade de reprogramação, confiabilidade, resposta rápida em tempo real e ocupa pouco espaço (PANDIT, 2017), (JAIN; AWARE; JUNGHARE, 2016). Ainda, alguns possuem ADC integrado no *chip* FPGA como é o caso do *kit* BeMicro Max 10, ou, externo como por exemplo o DE0-Nano que são abordados neste trabalho.

Os FPGA são compostos por dispositivos lógicos que podem ser programados. O núcleo é cercado por blocos de entrada de saída programáveis que são constituídas por vários

blocos lógicos dispostos em uma matriz que é conectada através de uma interconexão programável, estas podem ser memórias RAM, aceleradores de hardware, núcleos de processadores rápidos e lentos. A vantagem do projeto do FPGA é que ele fornece uma lógica cocorrente. A lógica sequencial pode ser implementada juntamente com uma arquitetura paralela, aumentando assim a velocidade do desempenho, ou seja, tem a possibilidade de agrupar diversos circuitos independentes em um mesmo dispositivo FPGA. Outra grande vantagem é o recurso de operações simultâneas caracterizada pela associação de circuitos lógicos, este recurso não está disponível na plataforma DSP ou microcontroladores. A Figura 14 mostra uma estrutura típica de um FPGA (PANDIT, 2017).

FIGURA 14 - Estrutura de um FPGA.



Fonte: PANDIT (2017).

2.8.2 IDE software Quartus II

Dentre os diversos modelos de FPGAs, o software de projeto lógico programável do modelo escolhido, o Quartus II, é usado para desenvolvermos a lógica de programa onde é empregada a descrição de hardware dos FPGAs, no caso da fabricante Altera (atualmente Intel). O analisador *Signal Tap II* e o visualizador RTL são usados para analisar e depurar o design. O analisador “*Time Quest*” é usado para restrição adequada de tempo e disposição do relógio (JAIN; AWARE; JUNGHARE, 2016).

Para se determinar a exatidão de um circuito lógico simula-se seu comportamento.

Para o modelo Quartus II da Altera (Intel), desenvolveu-se o código VHDL para o módulo PID e PWM. A ferramenta *Megawizard Plug-in Manager* pode ser usada para inserir desenhos esquemáticos. O diagrama esquemático inserido é processado por várias ferramentas do Quartus II, como o analisador *Time Quest*, o analisador *Signal Tap*, para analisar o arquivo, sintetizar o circuito e gerar uma implementação dele para o dispositivo alvo, assim é possível sintetizar o mesmo circuito analógico testado no PSIM. O analisador *Time Quest* analisa os caminhos de temporização no projeto, calcula o atraso de propagação ao longo de cada caminho e verifica violações nas restrições de tempo. Já o analisador *Signal Tap II* captura dados de sinal armazenado na memória do dispositivo, dispensando sondas e equipamentos externos para analisar o projeto e capturar o estado das portas de entradas e saídas internas e disponibiliza uma visualização das atividades lógicas no monitor. Após a síntese inicial, é gerada uma visualização RTL do circuito projetado como o mostrado no Apêndice D, onde existe um circuito lógicos interno à cada bloco, como pode ser visto em Apêndice E. Isso pode ser visto, utilizando o *RTL Viewer* (JAIN; AWARE; JUNGHARE, 2016).

De acordo com Pandit (2017), o ambiente Matlab-Simulink com a ferramenta geradora de sistema também é usado para gerar um código HDL que pode ser carregado em um FPGA. O desenvolvimento de todo o sistema em FPGA é feito usando os blocos reutilizáveis para prototipagem rápida e implementação eficiente. Uma vez que o algoritmo HDL é sintetizado em um FPGA, ele pode ser implementado em um chip CI para aplicações personalizados de um sistema.

Os FPGAs são altamente confiáveis e precisos, porém alguns deles não possuem integrados os ADCs. Nesses casos a desvantagem pode ser superada fazendo interface com um ADC externo na placa FPGA, para isso o *clock* precisa ser levado em consideração. Nesse caso, a parte desafiadora é ter o bloco completo que é um ADC juntamente com a geração de PWM em um FPGA com o tempo de conversão e o tempo de processamento de todos os componentes levados em consideração.

2.8.3 Modulação de pulso PWM

A implantação de modulação PWM em conversores estáticos é obtida gerando uma forma de onda repetitiva denominada portadora que é comparado com o sinal modulador gerado pelo controle PI. A forma de onda repetitiva terá uma frequência igual à frequência de

comutação. A implementação no FPGA envolve contador e comparador. A vantagem do PWM é que ele opera em alta frequência constante de comutação.

Outra característica do FPGA é o *delay-locked loops* (DLL) atraso bloqueado de loops. O PWM digital possui uma lógica síncrona que é implementada por um contador. O bloco assíncrono é implementado pela DLL usando o recurso de deslocamento de fase disponível nos FPGAs. Assim, o bloco PWM pode operar em alta frequência de comutação e a lógica restante pode operar em uma frequência externa mais baixa. A vantagem é o baixo consumo de energia e também a redução de atrasos. A fim de se obter frequências mais altas um multiplicador é usado para multiplicar o relógio externo mais lento para obter uma frequência de comutação mais elevada para o PWM, auxiliando o programador de forma que possa escolher a frequência mais adequada para seu projeto e possibilitando também a sincronização de diferentes processos que possam ser programados (PANDIT, 2017).

2.8.4 Aplicações

O FPGA é mais indicado para aplicações em que exigem processamento paralelo e altas velocidades de processamento, pode ser utilizado em controlar a saída de conversores estáticos empregando a tecnologia de comutação, na qual um PWM é gerado para controlar o acionamento do comutador, para se atingir uma tensão desejada visando diminuir as variações de saída (PANDIT, 2017).

Algoritmos de alta velocidade podem ser implementados eficientemente em um FPGA. As metodologias de projeto do FPGA podem ser classificadas em: 1) Nível do sistema: Os parâmetros do projeto são especificados. 2) Nível de comportamento: O comportamento do design é especificado. 3) Registrador do nível de transferência: O sistema é especificado com componentes combinacionais e sequenciais. 4) Nível Físico: O sistema é descrito em termos do hardware alvo, por exemplo: nível do transistor (Pandit, 2017).

O controle baseado em FPGA é implementado levando em consideração três regras principais para o projeto: 1) Refinamento: simplificação do algoritmo de controle 2) Modularidade: simplificação da utilização lógica 3) Compatibilidade entre o projeto e o hardware (PANDIT, 2017).

Os FPGAs são programados usando (HDL), estes são: *VHSIC hardware description Language*; (VHDL) e Verilog. O ambiente HDL fornece o resumo da utilização lógica do controlador. A simulação pode ser feita em um Simulador *Modelsim*. As vantagens

dessa arquitetura são que o controlador pode ser usado sem qualquer adição de componentes passivos. Por exemplo, um controlador proporcional PI, podemos ajustar a saída alterando os valores das constantes de ganho, dessa forma é possível obter respostas rápidas e configurável utilizando menos espaço (PANDIT, 2017).

Já o *Look up table* (LUT), consiste em um bloco lógico de 4 bits (*carry look ahead*) e um Flip Flop tipo D. O LUT tem entrada de 4 bits que tem 16 combinações. Além disso, possui uma memória RAM 16X1 ou ROM, e uma lógica combinacional de 4 bits, podendo ser implementada no LUT (PANDIT, 2017).

A vantagem de uma abordagem de projeto baseada em LUT é que ela consome menos recursos do FPGA, economizando área. Também possui baixo consumo de energia. Portanto, em aplicações, em que a variação do erro é pequena, é aconselhável utilizar uma abordagem baseada em LUT. Por outro lado, aplicações envolvendo grandes variações de erros levam a um grande número de entradas, nos projetos baseados em LUT, aumentando assim sua complexidade (PANDIT, 2017).

O esquema baseado no controle PID é amplamente preferido nas indústrias por causa de sua estrutura simples e facilidade de realização. O controlador PID, juntamente com o módulo PWM, é usado para controle de velocidade de motores elétricos, controle de conversores estáticos de potência e acionamentos elétricos. Para implementação digital, os microcontroladores também podem ser utilizados, mas o FPGA tem flexibilidade, eficiência de energia, além de cada vez com mais baixo custo, essas vantagens o tornam a cada dia mais competitivo no mercado.

De acordo com (JAIN; AWARE; JUNGHARE, 2016) e (DE0-Nano) os FPGAs são flexíveis. Com o FPGA é possível o usuário adicionar ou subtrair a funcionalidade conforme necessário de maneira fácil. Já controladores analógicos, por exemplo, trabalham em baixa frequência e ocupam muito espaço. No projeto do controlador analógico, ao alterar os parâmetros do controlador, temos que programar todo o circuito novamente.

No FPGA, não há estrutura fixa de hardware, por isso é definida pelo usuário. Para exemplificar esta vantagem a Figura 16 apresenta o modelo escolhido, o *kit* BeMicro Max 10 que se diferencia do outro modelo estudado pois possui conversor AD integrado no mesmo encapsulamento. Além disso, a taxa de amostragem máxima é feita a 1 MHz, dividida pelo número de entradas analógicas em uso. Já o DE0-Nano, apresentado na Figura 15, possui taxa de amostragem máxima de 200 kHz e a conversão AD é feita externamente pelo periférico

aumentando o tamanho do *hardwar*. Devido as vantagens foi definido para aplicação pratica o *kit* BeMicro Max 10.

Na Figura 16 pode-se observar o FPGA da Altera BeMicro Max10, onde constata-se a ausência do conversor AD externo, otimizando assim o espaço fisico do FPGA.

FIGURA 15 - Kit DE0-Nano

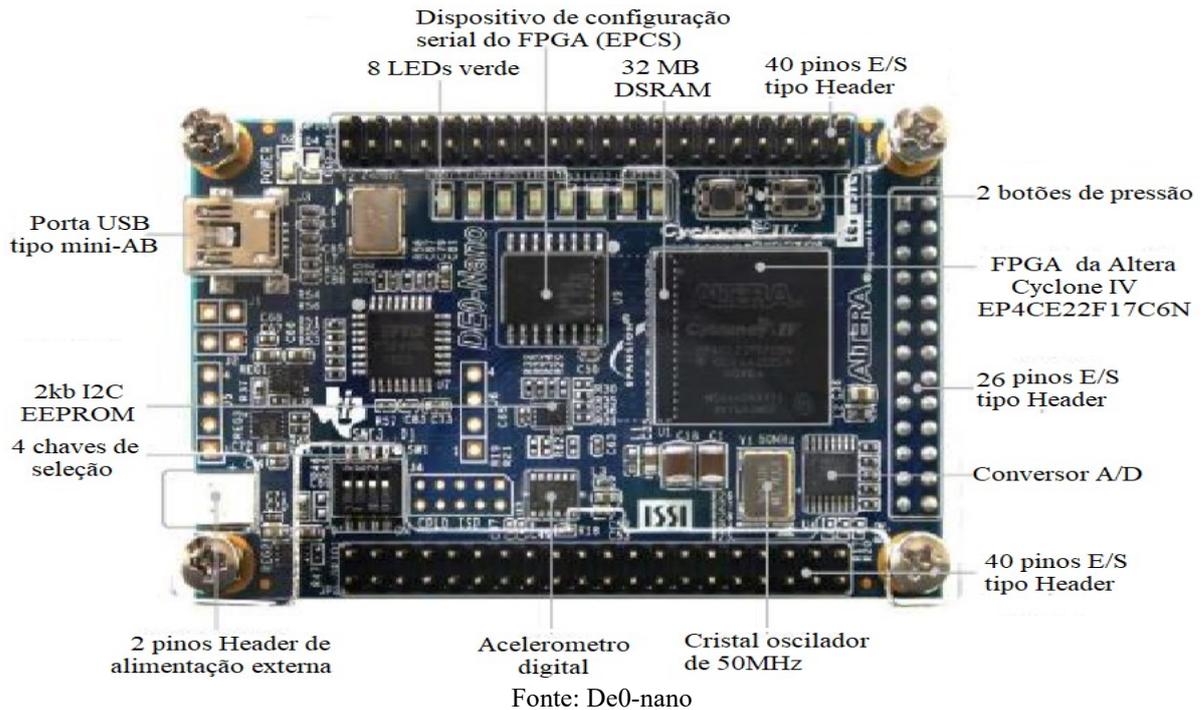
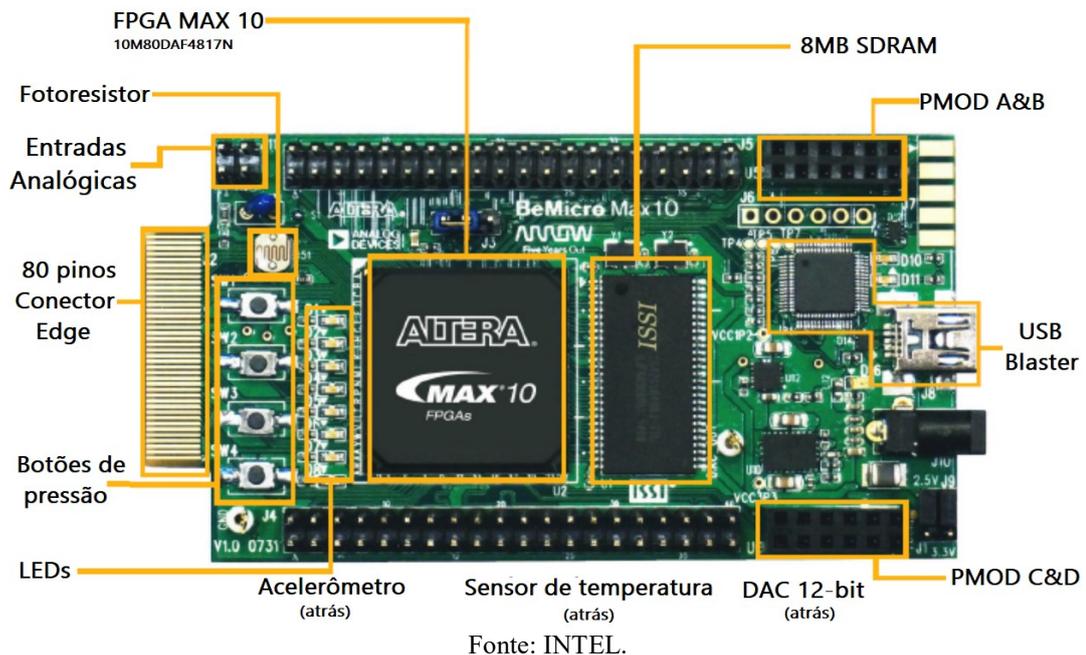


FIGURA 16 - Imagem kit Max 10 utilizado para realização dos experimentos



Devido a confiabilidade e processamento em tempo real o FPGA é muito empregado em conversores estáticos, sendo a melhor opção para gerenciamento de tarefas em malha fechada pelo fato da possibilidade de dividir os processos programados em blocos que podem ser processados de forma paralela ou sequencial, assim eliminando as dificuldades encontradas em processos sequenciais.

Como exemplo de aplicação do FPGA, pode-se citar a aplicação em conversores estáticos multiníveis. Alguns conversores multiníveis possuem a capacidade atenuar a ondulações de tensão de saída e a regulação de tensão pode ser melhorada. Dessa forma, as topologias multiníveis oferecem melhor flexibilidade no projeto do sistema modular, menor esforço nos semicondutores e na reconfiguração do sistema, como se pode observar em (SÁ, 2013). Geralmente as topologias multiníveis necessitam de sinais de modulação PWM distintas, então podemos concluir que pelo fato das vantagens de processamento paralelo o FPGA se destaca entre os demais processadores digitais, pois possibilita controlar dezenas e até centenas de sinais simultaneamente.

2.9 Comparação entre os microcontroladores

Fundamentalmente os microcontroladores são diferenciados principalmente pela sua arquitetura de processamento, capacidade da memória de armazenamento de dados, capacidade de processamento e instruções de programas (memória de programa e memória de dados), além da quantidade de pinos de E/S, quantidades e os tipos de periféricos, forma de alimentação, arquitetura e conjunto de instruções disponibilizadas nos circuitos internos (MARTINS, 2005).

Assim, ao comparar os microcontroladores estudados percebe-se que não é possível afirmar qual o melhor tipo de dispositivo, pois cada aplicação exige um determinado processador, método de processamento paralelo ou sequencial e velocidade de processar instruções que atenda a demanda do projeto. Além disso os dispositivos microcontroladores possuem recursos diferenciados, como ADC, DAC, comunicação com outros dispositivos e uma infinidade de recursos que devem ser analisados antes da aquisição de um microcontrolador. Dessa forma se adotarmos a aplicação em questão como exemplo foi necessário apenas dois canais ADCs, sendo que todos os microcontroladores analisados possuem este recurso, portanto o FPGA foi definido pela sua alta velocidade de processamento e recurso de processamento paralelo. Portanto, deve-se analisar as

particularidades de cada microcontrolador, sendo então possível entender como funcionam os microcontroladores, suas atribuições e restrições. Conhecendo suas especificações e demais atribuições é possível aplicar o microcontrolador nas mais diversas áreas.

Nesse sentido, uma das afirmações que pode ser feita é que o Arduino é o mais simples entre os microcontroladores estudados no presente trabalho, pois possui baixa capacidade de processamento, e não é possível utilizá-lo industrialmente por ter sido desenvolvido com intuito de ser uma ferramenta para fins didáticos, além de possuir baixa confiabilidade, pouca robustez e incompatibilidade com a maioria dos sensores e atuadores que trabalham com 24 V. Porém, destaca-se que o microcontrolador do Arduino Atmega328, assim como o PIC, são microcontroladores de baixo custo e pode ser utilizado industrialmente embarcado em diversos dispositivos.

Dessa maneira, entre os microcontroladores estudados o ESP32 se encontra em uma escala acima dos dois microcontroladores recém-salientado devido sua maior capacidade de processamento sendo que seu *clock* pode chegar a 160 MHz e pode ser multiplicado ou dividido de acordo com a necessidade, além de possuir diversos recursos como comunicação sem fio e dois núcleos de processamento e diversos módulos integrados.

O DSP é um dos microcontroladores mais apropriados para aplicação com altas densidades de processamento, especializado em processamento de sinais digitais, desempenha tarefas em tempo real se assemelhando ao FPGA, sendo muito aplicado em controle de conversores estáticos e processamento de som e imagem, porém não possui processamento paralelo.

Por fim, o FPGA não é considerado um microcontrolador porém pode assumir as mesmas funções, possui custo mais elevado entre os processadores abordados se comparado a modelos de mesmo nível (nível básicos ou completo), porém pode ser configurado para processar dados em paralelo ou de forma sequencial para atender a necessidade do usuário, além de possuir a blocos lógicos de fácil acesso. Assim através do código o usuário pode construir um circuito real internamente no chip. O mesmo também é capaz de substituir qualquer um dos microcontroladores citados, no entanto é necessária atenção com números reais, pois para sua grande maioria há a necessidade de conversão para ponto flutuante.

Com relação ao FPGA é importante ressaltar a possibilidade da programação ser feita com base em descrição de hardware. No momento em que é sintetizado o código é gerado um circuito interno no chip que se comporta como um hardware real que pode

processar informações de forma paralela ou sequencial. Dessa forma, é possível definir qualquer pino, desde que não seja reservado, como por exemplo, saída ou entrada PWM. É possível também sintetizar um circuito de comunicação UART e uma infinidade de circuitos.

Com o intuito de comparar os microcontroladores estudados, o Quadro 3, faz uma breve análise entre os microcontroladores abordados no trabalho.

QUADRO 3 - Análise comparativa entre os microcontroladores abordados.

Chip (Módulo)	Arduino UNO	PIC 18F4520	ESP32 (ESP-WROOM-32)	Kit DSP - TMS320F28379D	FPGA-BeMicro Max 10
DETALHES:					
Arquitetura de processamento – CPU	Atmel AVR – 8Bit – Harvard	Harvard	Tensilica X tensa LX6 32 bit Dual-Core e 160/240 MHz – Harvard	2x C28x 2x CLA Harvard - 32Bit	Processamento paralelo/ Sequencial
SRAM	2 kb	1,5 kb	520 KB	–	–
FLASH	32 kb	32 kb	2MB (máx. 64MB)	1024 kb	256 kb
Clock máx. (MHz)	16	40	160/240	200	50 (pode-se multiplicar ou dividir)
Tensão	5 V	3,3 V	2,2 V à 3,6 V	3,3 V	3,3 V
Corrente de operação	40 mA	20 mA	80 mA	20 mA	20 mA
Programação	C, C++ e Wiring	Assembly	Free (C, C++, Lua, etc.)	C, C++ e Assembly	VHDL, HDL, verilog, binária
Código aberto	Sim	–	Sim	–	–
CONECTIVIDADE:					
Wi-Fi	–	–	802,11 b/g/n	–	–
Bluetooth	–	–	4.2 BR/EDR + BLE	–	–
UART	Sim	Sim	3	Sim	–
E/S					
GPIO	14	36	32	97/169	250
SPI	Sim	Sim	4	3	6
I2C	Sim	Sim	2	2	4
PWM	6	Sim	8	24	–
ADC	6 (10-bit)	13(10-bit)	18 (12-bit)	4 (12/16-bit)	18 (12-bits)
Frequência Max. de conversão ADC	0,2 MSPS	-	2 MSPS	3,5 MSPS	1 MSPS
DAC	2	Sim	2 (8-bit)	3	Sim (8-bits)
Tamanho(mm)	68x53x10	53x16x5	25,5x18,0x2,8	130x59x11	90x48x11,6

Fonte: Autoria própria.

3. CONVERSOR CC-CC BOOST CLÁSSICO

3.1 Introdução

Conforme (SÁ, 2014) os conversores CC-CC podem ser empregados em uma infinidade de aplicações, como por exemplo, em sistemas de geração de energia renovável, veículos elétricos ou híbridos, células a combustível, UPS, sistemas diversos alimentados por baterias que necessitem de estágio de conversão e sistemas de alimentação bidirecional.

Pensando nas inúmeras aplicações dos conversores CC-CC, tornam-se importantes pesquisas relacionadas na área, principalmente quando diz respeito a crescente demanda em energia elétrica que por consequência gera o aumento na produção de energia além do maior número de pesquisas relacionadas a geração de energias alternativas. Quando se fala em novas fontes de energia não se pode deixar de citar a geração fotovoltaica, que vem a cada dia ganhando maior notoriedade perante as demais formas de geração.

Dentro desse contexto, o conversor Boost torna-se um importante tema de estudo, pois está diretamente relacionado a geração de energia fotovoltaica conectada à rede. Assim, pensando em futuras aplicações neste trabalho será estudado o conversor Boost e seu controle através do FPGA escolhido BeMicro Max 10.

Neste capítulo será realizado um estudo teórico sobre o conversor Boost onde é apresentado as equações que regem seu funcionamento. Serão também obtidos os resultados de simulações para posteriormente ser implementado na prática. Primeiramente será estudado o princípio de funcionamento do conversor Boost, as etapas de operação juntamente com suas respectivas formas de onda, assim com o ganho estático também são apresentadas neste capítulo.

3.1.1 Aplicações

Dentre as aplicações do conversor Boost podemos citar:

- **Energia Fotovoltaica:** (SANTOS, 2018) e (SÁ, 2014) explicam que para obter uma tensão elevada na saída de painéis fotovoltaicos, é preciso

realizar associação em série desses painéis. Entretanto, essa associação diminui a possibilidade de os painéis serem controlados para operar no ponto de máxima potência, e podem gerar ineficiência caso haja deficiência em um dos dispositivos em série. Ou seja, quanto menor a associação em série, maior é a eficiência do sistema para este caso. A desvantagem da conexão em série é a alta corrente devido a baixa tensão. Esse problema pode ser contornado se amplificarmos a tensão de saída para diminuir as perdas, assim o projetista pode arranjar os painéis de forma a gerar maior eficiência sem se preocupar com a tensão de saída.

- **Células a Combustível:** De modo teórico uma célula a combustível pode disponibilizar aproximadamente um volt entre seus terminais. Todavia, na prática este valor é bem inferior, apresentando ainda um rápido decréscimo com o aumento da corrente drenada. A associação de diversas células em série resulta em um valor de tensão de saída elevado. Entretanto, quanto maior a potência na saída da associação em série, menor é a tensão de saída. Essa característica mostra que é necessário um estágio elevador de tensão, e controle de potência onde a máxima potência consiga ser associada com uma tensão elevada de saída (SÁ, 2014).
- **Veículos Elétricos ou Híbridos:** Atualmente, os carros híbridos que eram considerados apenas uma possibilidade remota, estão sendo fortemente estudados em núcleos de pesquisa e já estão sendo comercializados no mercado automotivo. O conceito de veículo elétrico é descrito por um veículo que se move por meio de um ou mais motores elétricos. Exemplos de veículos elétricos são carros, trens, caminhões, barcos, aviões, entre outros. Esse tipo de veículo possui a vantagem de consumir energia elétrica primária gerada por uma grande variedade de fontes renováveis, e não apenas de combustíveis fósseis. A alimentação desses veículos pode ser feita por baterias, painéis solares ou mesmo por condutores específicos posicionados na rota deste veículo, como no caso de trens, metros e ônibus elétricos urbanos. O emprego de conversores CC-CC é amplamente utilizado nesses veículos, pois na maioria dos casos emprega-se baterias para alimentação do sistema (SÁ, 2014).
- **Nobreaks:** Os Nobreaks são também conhecidos como UPS, ou seja, fonte

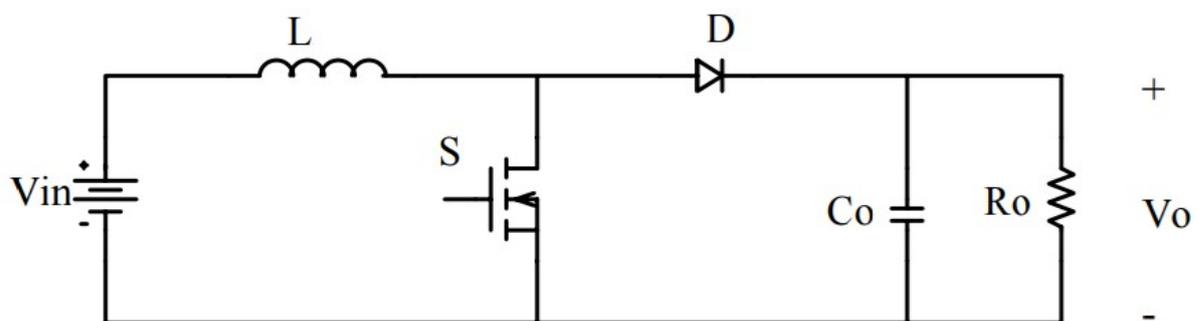
ininterrupta de energia. Seu funcionamento consiste de um equipamento elétrico que tem como objetivo principal suprir instantaneamente, ou quase instantaneamente, uma carga crítica, em caso de queda inesperada de energia. No momento que ocorre a falta, o nobreak é desconectado da rede e a bateria assume a alimentação da carga. Os nobreaks representam um ramo promissor de aplicação de conversores CC-CC, com a possibilidade da criação de uma nova tecnologia ou da adaptação de uma das existentes (SÁ, 2014).

3.2 Princípio de funcionamento do conversor Boost

O conversor Boost é um conversor elevador de tensão. Conforme Brockveld (2017), o mesmo é aplicado em sistemas que demandam tensão elevada, porém por ser um conversor não isolado, emprega-se onde não se necessita de isolação entre a fonte e a carga. Dentre suas principais vantagens destacam-se o alto rendimento e a simplicidade da topologia, pois possui apenas um interruptor, além de um capacitor e um indutor como elementos armazenadores de energia, como pode ser observado na Figura 17.

Porém, para aplicações que exigem altas potências e tensões elevadas normalmente é utilizado topologias quadráticas ou ainda conversores multiníveis, pois esta última é originada pela associação de conversores em série ou paralelo. Entretanto, ao se utilizar topologias mais complexas devem-se levar em consideração as perdas nos semicondutores, pois a quantidade de semicondutores no circuito pode influenciar na eficiência do conversor, (BROCKVELD, 2017) e (SÁ, 2014).

FIGURA 17 - Conversor CC-CC Boost clássico



Fonte: BASTSCHAUER; NETO; e PETRY (2000).

Conforme descrito por Erickson (1997) e também em Brockveld (2017), o ganho estático ideal do conversor Boost tende ao infinito quando a razão cíclica se aproxima da unidade. Considerando que d é a razão cíclica aplicada ao interruptor S , após a análise do conversor Boost a Equação que define o ganho estático ideal do conversor é dado pela Equação 1:

$$G_{boost} = \frac{1}{1-d} \quad (1)$$

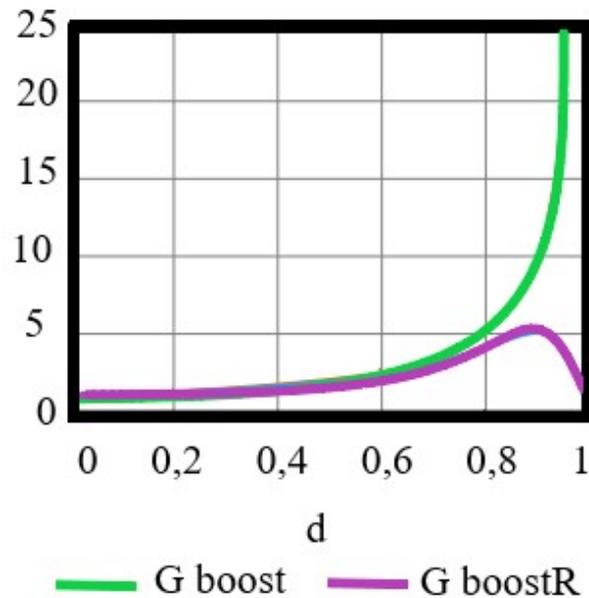
Segundo Erickson (1997) e Brockveld (2017), sabe-se que na prática, os elementos parasitas limitam o ganho estático do conversor. Conforme Erickson (1997) e também apresentado em Brockveld (2017), considerando como exemplo, uma resistência (R_L) em série com o indutor, onde R é a resistência da carga. No modo de condução contínua o ganho estático passara a ser definido pela Equação 2:

$$G_{boost_R} = \frac{1-d}{(1-d)^2 + \frac{R.L}{R}} \quad (2)$$

Conforme explicado por Erickson (1997) e Brockveld (2017), devido as perdas série do indutor e nos componentes, o conversor Boost se limita em aplicações que demandam elevado ganho estático. Outro item importante a ser considerado neste modelo de conversor para altos ganhos é a razão cíclica a ser adotada, esta deve permanecer aproximadamente 90% do período em condução, porém, isso e dificulta a implementação prática. Para este caso é empregado outras topologias de conversores quadráticos, onde basicamente são originados através da associação de dois ou mais conversores, em série como pode ser visto um exemplo em (BROCKVELD, 2017) e em paralelo exemplificado por (SÁ, 2014).

Na Figura 18 apresenta-se uma comparação entre o ganho estático do conversor Boost com perdas e sem perdas. A característica estática do conversor Boost sem perda é demonstrado na curva em verde (GBoost), e a característica estática em roxo (GBoostR) considera em modo de condução contínua a resistência série do indutor, dessa forma comprava-se o real funcionamento do conversor (BROCKVELD, 2017).

FIGURA 18 - Curva do ganho estática do conversor Boost.



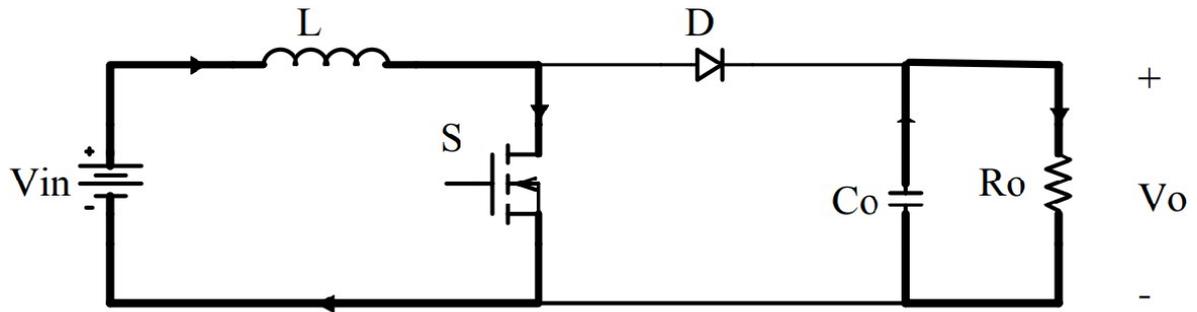
Fonte: BROCKVELD (2017)

3.2.1 Princípio de funcionamento do conversor Boost operando em condução contínua

Para implementação do conversor Boost é importante entender o seu funcionamento, portanto, é apresentado aqui as etapas de operação e suas respectivas formas de onda e as equações que regem o funcionamento do conversor.

Na primeira etapa de operação, no instante T_0 , o interruptor S está aberto, nesse período o indutor L armazena energia fornecida pela fonte V_{in} . A corrente no mesmo cresce linearmente até atingir o seu valor de pico em T_1 . No instante T_1 o interruptor S é aberto, o que caracteriza o final da primeira etapa de operação. Nesta etapa a corrente de alimentação da carga é fornecida pelo capacitor C_0 . A Figura 19 demonstra o circuito do conversor para a primeira etapa de operação em condução contínua (BASTSCHAUER; NETO; PETRY, 2000).

FIGURA 19 - Primeira etapa de operação do conversor em condução contínua.



Fonte: BASTSCHAUER; NETO; e PETRY (2000).

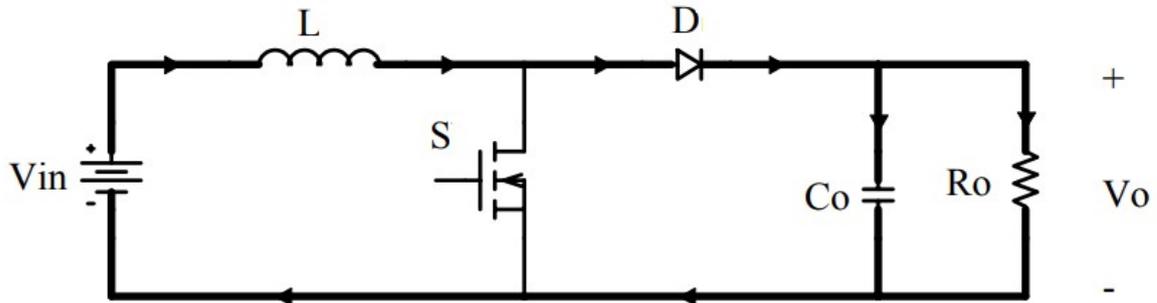
De acordo com Bastschauer, Neto e Petry (2000) o período em que o interruptor permanece em condução é definida pelo comando recebido do FPGA que mantém o interruptor ligado por um determinado período. Quanto maior for o tempo em condução, dado pela razão cíclica, maior será a tensão de saída. Ao término desta, etapa a corrente do indutor atinge seu pico máximo. Durante a primeira etapa de operação a corrente no indutor é definida pela Equação 3:

$$I_{in}(t) = I_{0(min)} + \frac{V_{in}}{L} \cdot t \quad (3)$$

Conforme Bastschauer, Neto e Petry (2000), a segunda etapa de operação é marcada pelo interruptor S aberto, para $T=T_1$, onde o diodo D entra em condução. Durante esta etapa, a fonte V_{in} e o indutor L disponibilizam energia para a carga, desmagnetizado parcialmente o indutor L . A corrente no indutor decresce linearmente até que no instante T_2 , caracterizando o final da segunda etapa de operação. Na Figura 20 mostra-se o circuito do conversor para a segunda etapa de operação. Já a Figura 21 apresenta as principais formas de ondas do conversor Boost após a análise das etapas de operação do conversor. Durante essa etapa a corrente no indutor Boost é dada pela Equação 4:

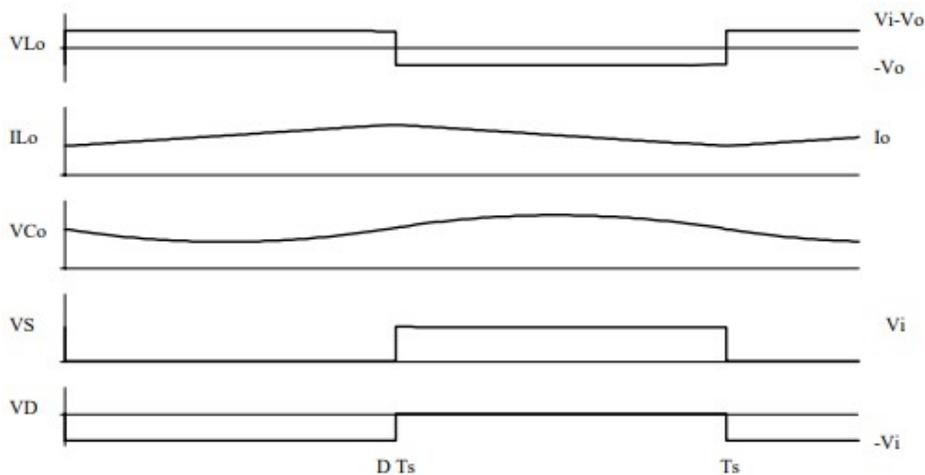
$$I_{in}(t) = I_{p-} + \frac{V_0 - V_{in}}{L} \cdot t \quad (4)$$

FIGURA 20 - Segunda etapa de funcionamento do conversor em condução contínua.



Fonte: BASTSCHAUER; NETO; e PETRY (2000).

FIGURA 21 - Principais formas de onda do conversor Boost.



Fonte: BASTSCHAUER; NETO; e PETRY (2000).

3.2.2 Simulação do conversor Boost em malha aberta:

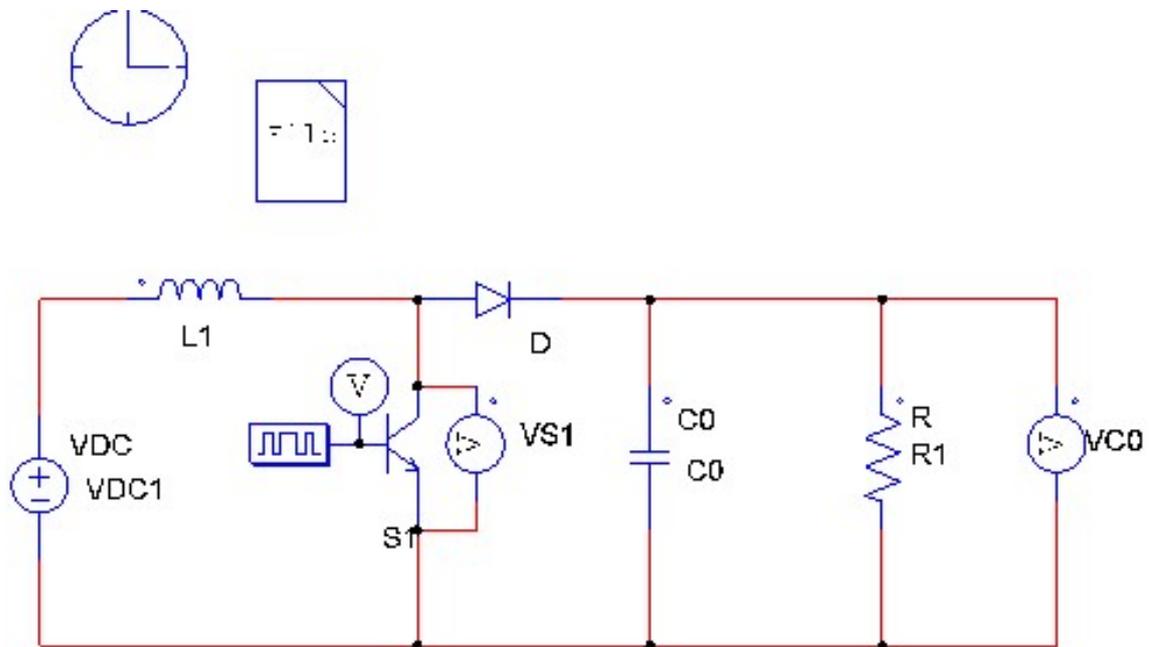
Nesse item serão apresentados os resultados de simulação do conversor Boost no modo de condução contínua. Os valores utilizados para a simulação do conversor no software PSIM, foram os mesmos apresentados no capítulo seguinte, conforme mostrados na Tabela 4 para os resultados experimentais, com o intuito de comparar os resultados simulados e práticos obtidos inicialmente em malha aberta. O projeto do conversor foi realizado no software Mathcad, e se encontra no Apêndice B.

A Figura 22 ilustra o esquemático do conversor simulado no PSIM.

Na Figura 23 são apresentados os resultados de simulação das principais formas de

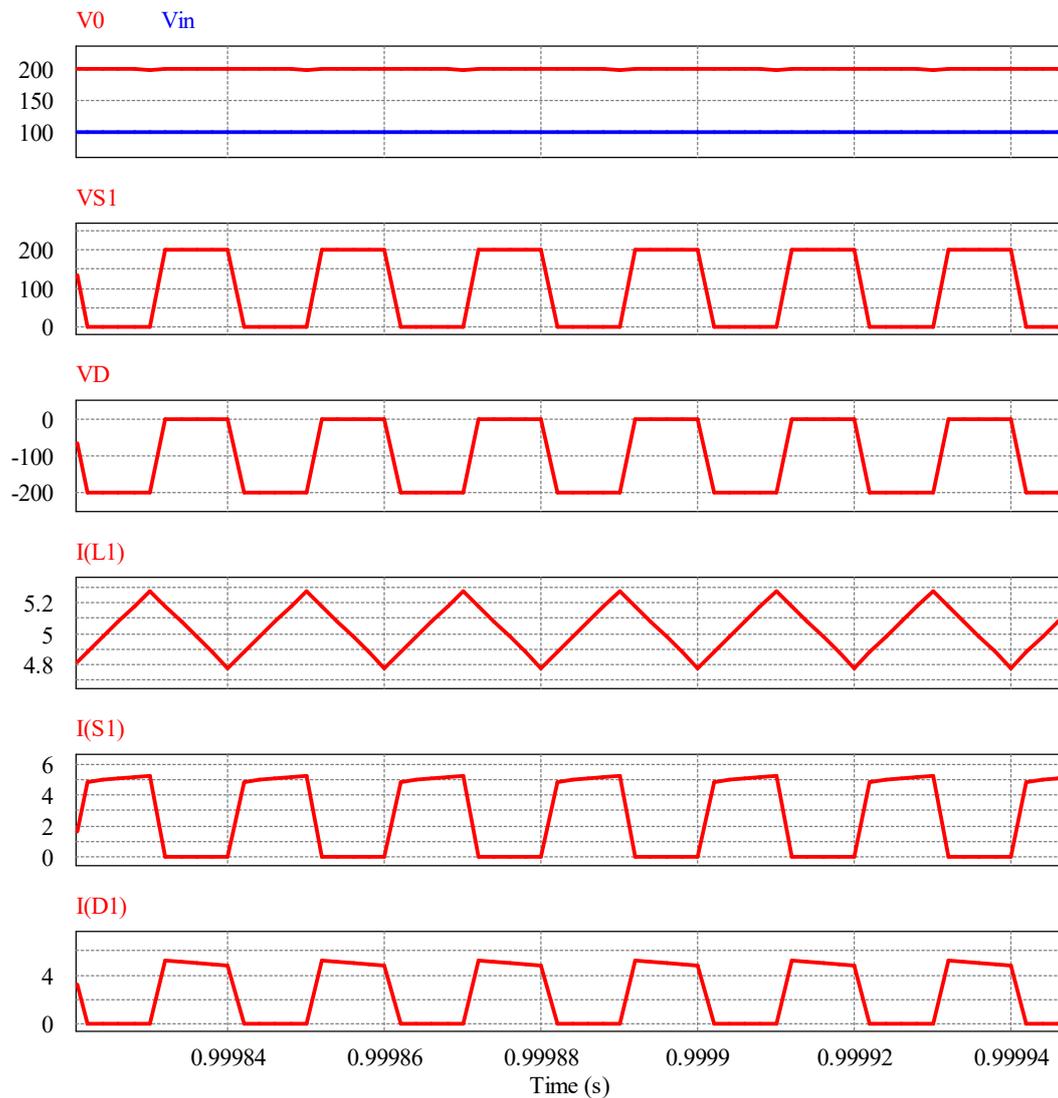
onda do conversor Boost, onde pode-se observar que para uma razão cíclica utilizada de 0,5 o ganho estático do conversor será dois, ou seja, a tensão na saída será o dobro da tensão de entrada. Além disso, conforme comentado no estudo teórico a corrente no indutor de entrada varia de um valor mínimo para um valor máximo na primeira etapa de operação, conforme apresentado na Equação 3, e varia de um valor máximo para um valor mínimo na segunda etapa de operação, conforme mostrado na Equação 4. O valor dessa variação segue o especificado no projeto, conforme o Apêndice B. Já a tensão no interruptor S possui um valor nulo na primeira etapa de operação, quando o interruptor está conduzindo e o mesmo valor da tensão na saída na segunda etapa de operação, quando o interruptor está bloqueado conforme pode ser visto na figura 23. Também pode-se concluir que os resultados da simulação estão de acordo com os valores calculados e também leva em conta a oscilação de tensão e corrente, portanto é possível afirmar que a corrente e tensão máxima se aproximam respectivamente de 5.25 A e 202 V.

FIGURA 22 - Esquemático conversor Boost utilizado na simulação, em MA.



Fonte: Autoria própria (2018).

FIGURA 23 - Principais formas de onda obtidas na simulação do conversor Boost.

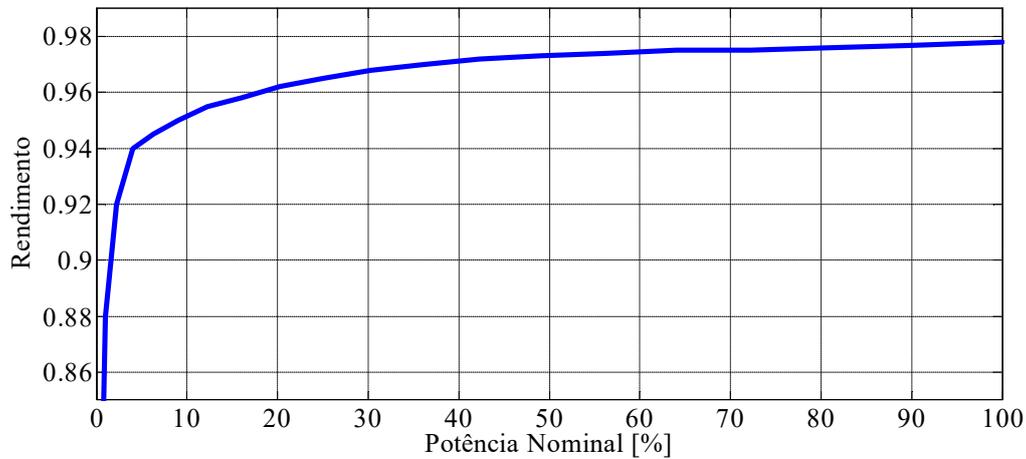


Fonte: Autoria própria (2018).

Outro aspecto importante a ser observado nos conversores estáticos é seu rendimento. Sabe-se que o valor ideal seria 100% conforme Equação (1), mas isto é impossível pelas leis da física. Com isto espera-se que os conversores possuam um elevado rendimento, pois dessa forma mantém-se a eficiência de energia, diminuindo assim as perdas.

Para a obtenção do rendimento através da simulação do conversor Boost foi utilizado o software PSIM, onde através da inserção do modelo real dos semicondutores utilizados, através da ferramenta Device Database Editor do PSIM, utilizando um interruptor Mosfet Cool MOS e um Diodo Schottky. A Figura 24 apresenta o rendimento do conversor Boost por meio de simulação.

FIGURA 24 - Rendimento do conversor Boost simulado no PSIM.

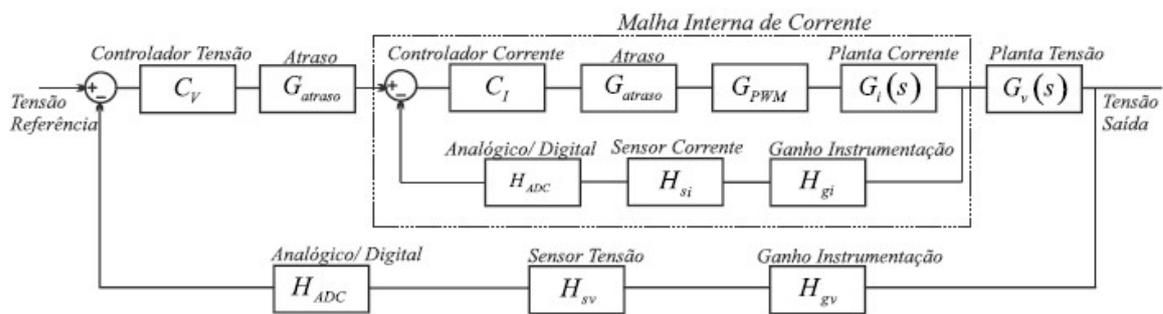


Fonte: Autoria própria (2018).

3.3 Conversor Boost operando em Malha Fechada:

Para se aplicar o controle no conversor Boost é importante primeiramente se obter a modelagem das plantas para as malhas de corrente e tensão. Considerando que a tensão de saída seja controlada através do valor médio da corrente do conversor, aplicam-se dois controladores, um para cada malha de controle respectivamente. A Figura 25 apresenta o diagrama de blocos utilizado na implementação do controle de tensão com malha interna de corrente (SÁ, 2014).

FIGURA 25 - Diagrama de blocos da malha de corrente interna à malha de tensão.



Fonte: SÁ (2014).

Assim, considerando primeiramente a malha interna de corrente, adota-se o modelo matemático simplificado para a planta de corrente, obtida a partir da análise de pequenos sinais, conforme apresentado em Erickson (1997) e Sá (2014), e mostrado na Equação (5):

$$G_{iL} = \frac{\hat{i}_{L1}(s)}{\hat{d}(s)} = \frac{V_0}{L \cdot s} \quad (5)$$

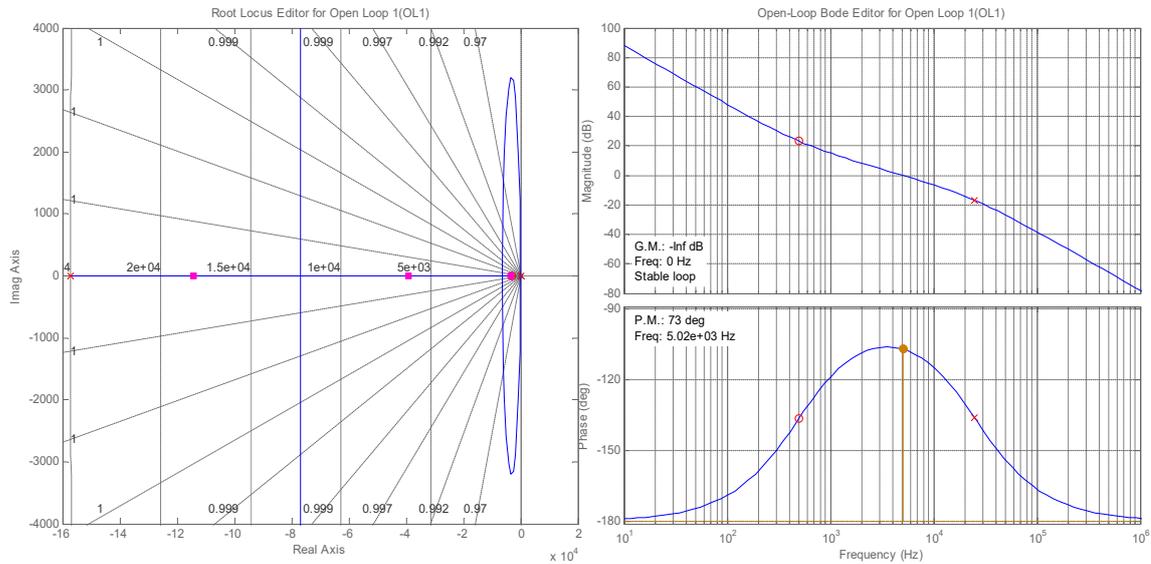
Para o projeto do controlador PI considera-se primeiramente que um valor alto do zero do compensador para tornar a resposta rápida; além disso, o pólo do compensador deve ser inserido acima da frequência de comutação do conversor de forma atenuar os ruídos da malha de realimentação. Dessa forma, a Equação (6), apresenta o compensador de corrente, assim como seu projeto, segundo Sá (2014) e Batschauer (2012), como seguem:

- ✓ P_c – pólo do controlador deve ser posicionado acima da frequência de comutação;
- ✓ Z_c – zero do controlador deve ser posicionado uma década abaixo da frequência de cruzamento;
- ✓ K_{pi} – o ganho do compensador deve ser projetado para que o sistema apresente uma margem de fase baixa (entre 45° e 90°), e uma frequência de cruzamento igual a um décimo da frequência de chaveamento.

$$C_i(s) = k_{pi} \cdot \frac{s + z_c}{s \cdot (s + p_c)} \quad (6)$$

A Figura 26 apresenta a Função de Transferência de laço aberto (FTLA), para o projeto do controlador de corrente, desenvolvido no software Matlab, no SISOTOOL:

FIGURA 26 - FTLA da malha interna de corrente.



Fonte: Autoria própria (2018).

Após o projeto da malha interna de corrente projeta-se a malha externa de tensão:

A função de transferência que apresenta o modelo de pequenos sinais para a planta de tensão está apresentada na Equação 7. Este modelo simplificado considera apenas a razão cíclica, além da capacitância e da resistência de saída do conversor (SÁ, 2014).

$$G_{V_0} = \frac{\hat{v}_0(s)}{\hat{i}_L(s)} = R \cdot \frac{2}{\pi} \cdot D \cdot \frac{1}{R \cdot C \cdot s + 1} \quad (7)$$

Com o controlador PI de tensão inserido, a tensão de saída deve se manter inalterada independente da carga, ou seja, independente da corrente na entrada do conversor. De posse do modelo da planta de tensão, projeta-se o controlador de tensão.

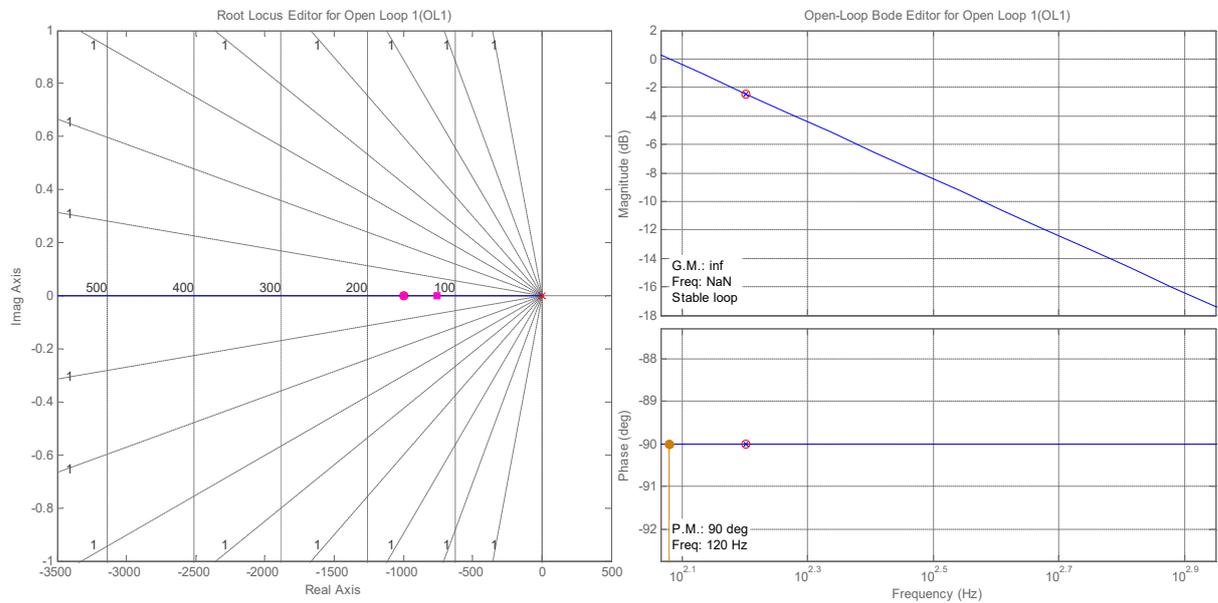
No projeto do controlador PI de tensão considera-se um sistema de primeira ordem, onde o pólo na origem é responsável pelo erro nulo em regime permanente, e o zero do compensador é posicionado sobre o pólo da planta, de forma a cancelá-lo. O ganho do compensador é determinado com o intuito de garantir a frequência de cruzamento por zero estabelecida, que neste caso será de 120 Hz, proporcional a frequência da rede de 60Hz. Assim, na Equação (8) mostra-se o compensador de tensão utilizado (SÁ, 2014).

$$C_v(s) = k_{pi} \cdot \frac{s + z_{cv}}{s} \quad (8)$$

A Figura 27 apresenta a Função de Transferência de laço aberto (FTLA), para o

projeto do controlador de tensão, desenvolvido no software Matlab, através da ferramenta SISOTOOL:

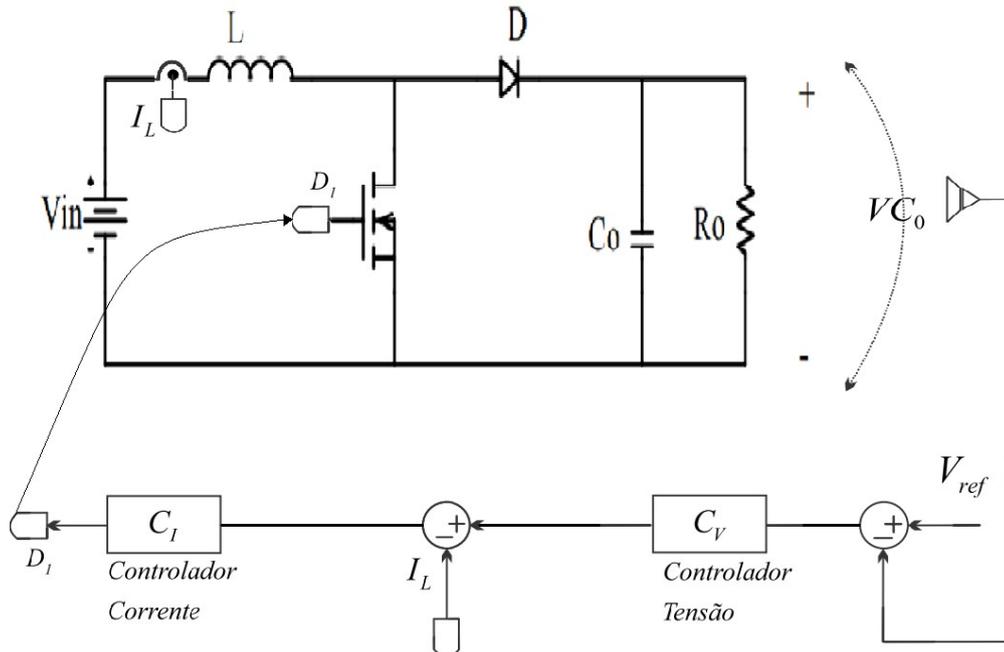
FIGURA 27 - FTLA da malha externa de tensão.



Fonte: Aatoria própria (2018).

Na ilustração da Figura 28 é demonstrado o esquemático do controle de tensão e corrente em cascata. É possível observar que o controle de tensão se localiza na malha externa e passa pelo controle de corrente interno ao PI de tensão. Dessa forma, a malha de corrente é mais rápida, por ser a malha mais interna e por ser sensível à corrente de entrada do conversor, agindo antes que a variação seja percebida pela malha de tensão.

FIGURA 28 - Esquemático do conversor Boost, controle da corrente e tensão.



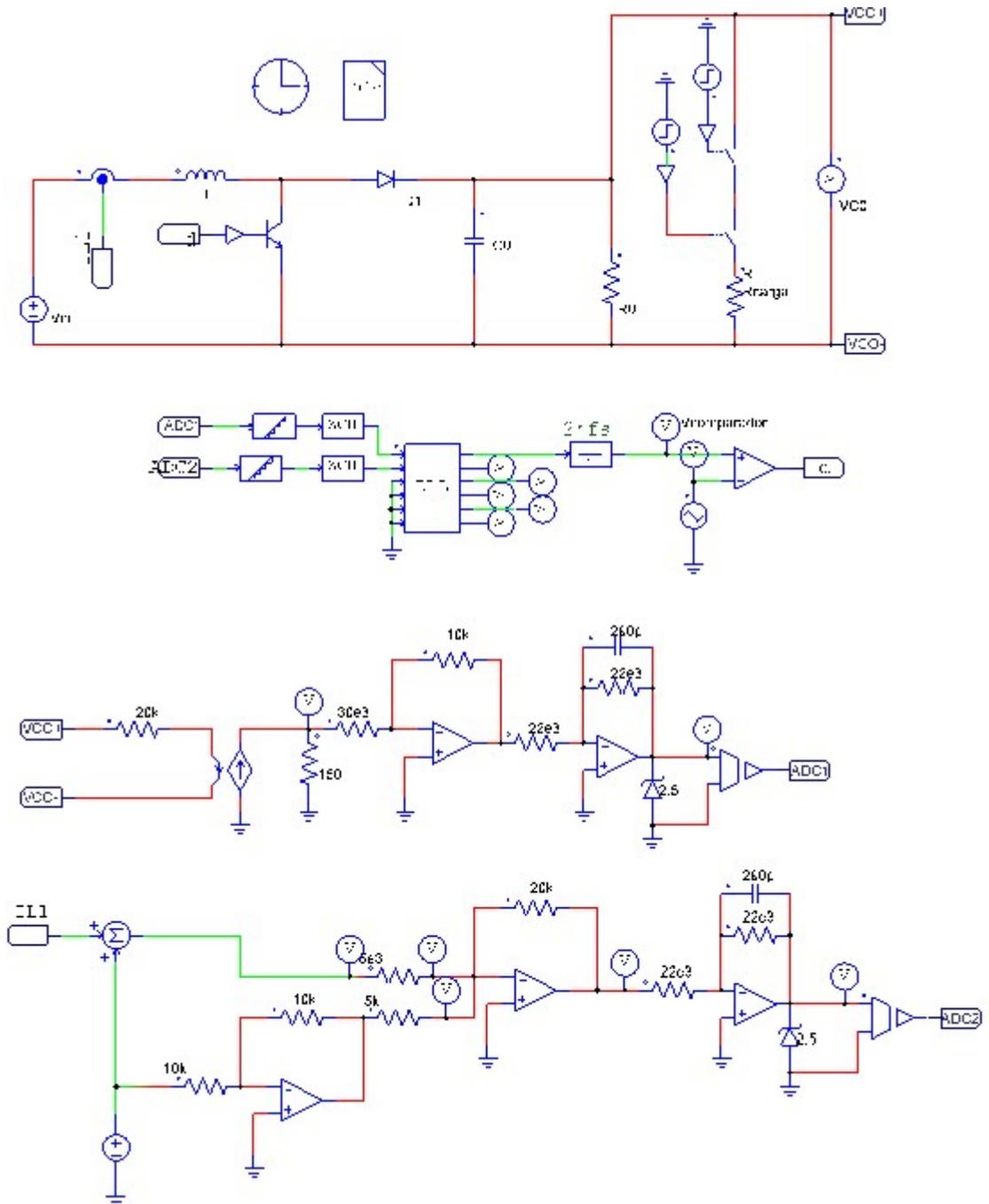
Fonte: Autoria própria (2018).

3.3.1 Simulação do conversor Boost em malha fechada:

Na Figura 29 está ilustrado o esquemático do conversor Boost em malha fechada (MF), simulado no PSIM.

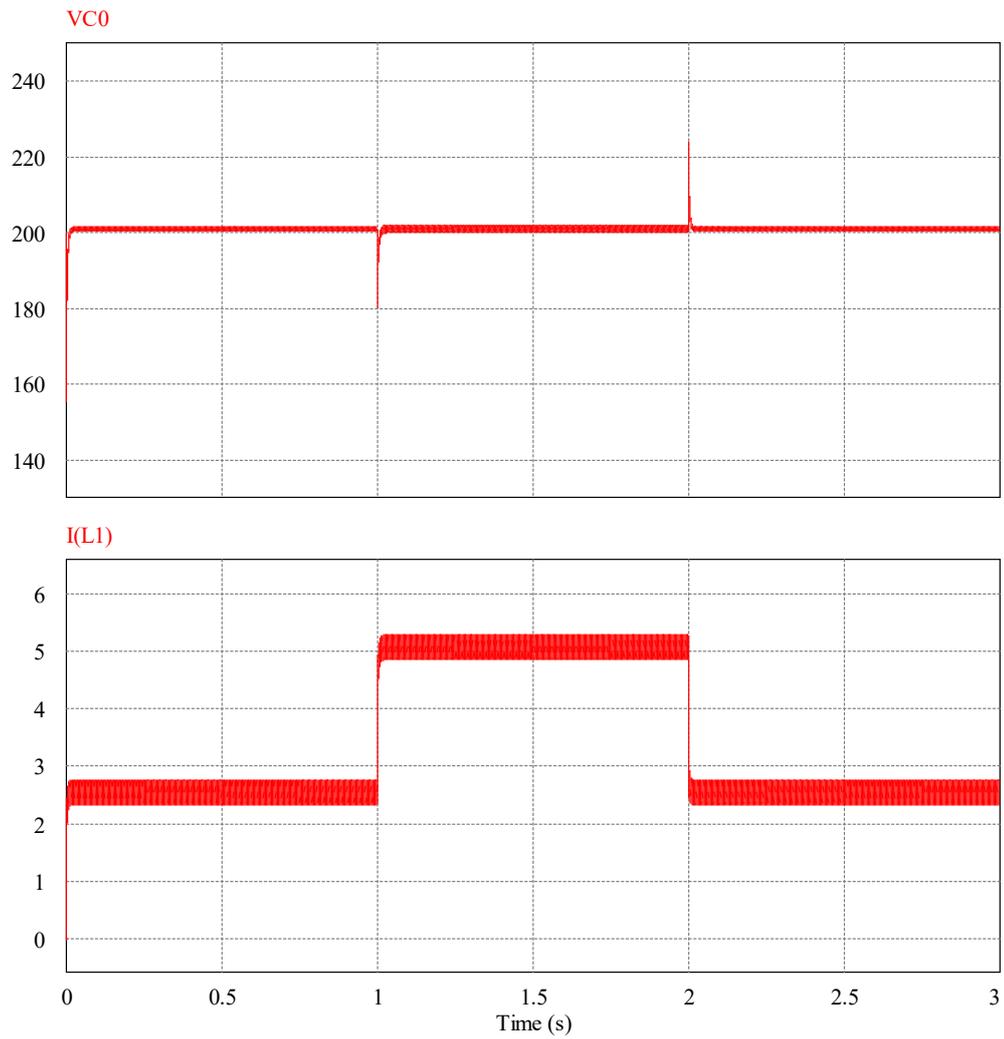
Já na Figura 30 são apresentados os principais resultados de simulação do conversor em malha fechada, onde se pode observar que apesar da variação de carga ser de 50% para 100% do seu valor nominal, isto comprovado através do valor da corrente de entrada (I_{L1}), o valor da tensão de saída não varia, a não ser apresentando pequenos picos nestes instantes. Dessa forma pode-se observar que o controlador PI empregado na simulação e projetado para o conversor, atende plenamente as necessidades requeridas.

FIGURA 29 - Esquemático conversor Boost utilizado na simulação, em MF.



Fonte: Aatoria própria (2018).

FIGURA 30 - Principais formas de onda obtidas na simulação do conversor em MF.



Fonte: Autoria própria (2018).

4. RESULTADOS E DISCUSSÕES

4.1 Introdução

Nesse capítulo é apresentado as especificações, discussão e os resultados experimentais do conversor estudado. O mesmo trata-se de um conversor elevador de tensão Boost projetado para operar com tensão nominal de entrada $V_{in} = 100V$, e com resistência de carga de $R = 80 \Omega$, quando submetido a potência nominal é capaz de fornecer $500 W$ em seus terminais. Para isso foi definido frequência de comutação de $50 kHz$. Este valor foi escolhido por ser uma frequência intermediária, desta forma há um equilíbrio entre o aquecimento gerado pela comutação do interruptor e a dimensão do indutor. O FPGA possui facilidade de gerar diversas frequências, bastando apenas multiplicar ou dividir o *clock* interno.

Para os testes em malha fechada, ou seja, quando o sistema faz a leitura da saída e realimenta o controlador, é empregado inicialmente baixa carga, posteriormente é aplicado o degrau de carga onde se tem um acréscimo na potência de aproximadamente 40%. Qualquer outra grandeza aplicada no degrau de carga será válida desde que sejam respeitadas as restrições do conversor Boost, não sendo possível exceder a potência nominal para não danificar os componentes do conversor, para isto o valor da carga não deve ser um valor maior do que o projetado, conforme o Apêndice B.

Para realizar o projeto do Conversor Boost foram definidas as condições iniciais de forma à operar com potência nominal de $500 W$ e $200 V$ na saída tendo como frequência de comutação de $50 kHz$ e ganho estático igual a dois. As grandezas apresentadas na tabela 4, oriundas dos cálculos disponíveis em Apêndice B foram baseados em Martins; Barbi (2006), e calculados de forma a atender as condições iniciais. O projeto do indutor disponível em Apêndice B foi baseado em Lange (2012) sendo consultado a tabela de fio esmaltado em Barbi (2000) e as informações do núcleo encontram-se em disponível em Thornton (2018).

TABELA 4 - Parâmetros experimentais utilizados para o conversor Boost.

Descrição	Atribuição
Tensão de entrada	$V_{in} = 100 V$
Corrente de saída	$I_0 = 2,5 A$
Tensão de saída	$V_0 = 200 V$
Indutância de entrada	$L = 2 mH$
Capacitor de saída	$C_0 = 0,125 \mu F$
Resistência da carga	$R_0 = 80 \Omega$
Frequência de chaveamento	$f_s = 50 kHz$
Razão cíclica dos interruptores	$D = 0,5$
Potência de saída	$P_0 = 500 W$

Fonte: Autoria própria (2018)

Primeiramente o conversor foi projetado no Mathcad de acordo com alguns parâmetros pré-estabelecidos como V_{in} , V_0 , f_s , e potência desejada, os demais valores são obtidos através dos cálculos. Após a execução dos cálculos disponível no Apêndice B pode-se determinar as características físicas dos componentes a serem utilizados, a partir de então pode ser desenvolvido o esquemático no Altium Designer.

Para a construção do indutor optou-se por um núcleo toroidal para aperfeiçoar o tamanho do componente. Caso houvesse a necessidade de otimizar o tamanho do indutor, optar-se-ia por aumentar a frequência de comutação, porém neste caso não foi necessário.

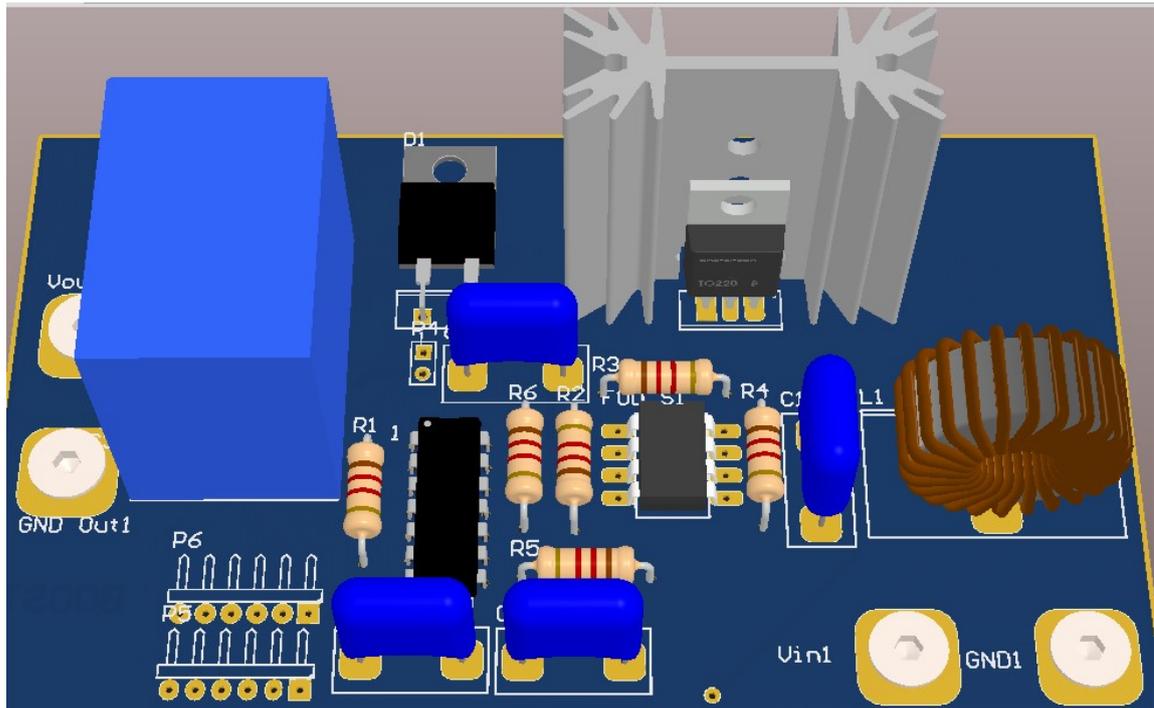
4.2 Desenvolvimento do protótipo

Para o desenvolvimento do protótipo foi utilizado o software *Altium Designer*, onde primeiramente foi desenhado o esquemático apresentado em Apêndice C. Em seguida foi feito o roteamento da placa, ou seja, conectar os componentes através das trilhas, podendo ser feito de maneira manual, utilizando o *Interactively Route Connections*, ou então automaticamente utilizando o recurso *Auto Route do software*.

Posteriormente foram seguidos os passos para geração de dos arquivos .GTL, .GBL, .GKO e .TXT, que são necessário para fazer a fresagem com a Fresadora CNC

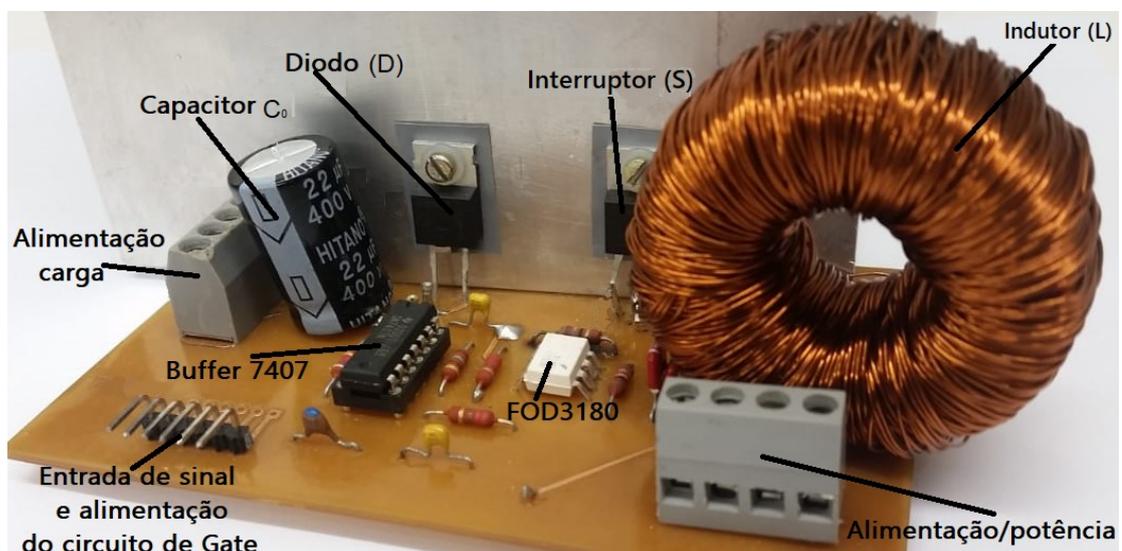
(comando numérico computacional). Ao término da fresagem deve-se remover o cobre que não interessa para o circuito que é feito de forma manual ou na Fresadora. A Figura 31 é uma representação do Conversor Boost em 3D, em seguida na Figura 32 pode-se observar o conversor após a finalização do processo.

FIGURA 31 - Imagem em 3D do conversor Boost, com $P_0 = 500 W$.



Fonte: A autoria própria (2018).

FIGURA 32 - Vista Lateral do protótipo do conversor Boost implementado, $P_0 = 500W$.



Fonte: A autoria própria (2018).

4.2.1 Especificação dos semicondutores

Adotando como base os resultados da simulação, que podem ser observados na Figura 23, é levando em conta a corrente e tensão de pico, dessa forma os semicondutores devem suportar grandezas maiores que 200 V e 5 A. Porém por questão de disponibilidade, utilizou-se interruptor Mosfet Cool MOS modelo 24N60C3, com capacidade de drenar uma corrente de 24,3 A e tensão de 650 V entre seus terminais *drain* e *source*, e um diodo *Schottky* modelo SDT10S30 com capacidade para suportar 300 V e 10. Utilizou-se para os dois componentes encapsulamento TO-220.

4.2.2 Especificação do capacitor

Levando em conta a capacitância e os esforços de tensão e corrente conforme Apêndice B, optou-se pelo capacitor de Film pela sua alta capacidade de drenar a corrente, dessa forma não há a necessidade de utilizar um banco de capacitores otimizando-se o espaço.

O capacitor definido, da fabricante Epcos, código B32794D2156 possui 15 μ F, suporta 630 V e corrente de 13 A eficaz. Outra opção é a aplicação do capacitor de código MKP1848615454P da Vishay que suporta 10 A e 450 V porém ocupa uma área um pouco maior em relação ao capacitor da Epcos.

No entanto por questão de custo e também pelo conversor não ser o foco principal do trabalho, foi utilizado um capacitor eletrolítico da fabricante Hitano de 22 μ F. Não sendo indicado devido a baixa capacidade de corrente, porém atendeu as necessidades, pois foi submetido a testes por períodos muito curtos.

4.2.3 FPGA utilizado

O processador escolhido para a implementação prática foi o FPGA da altera, retratado na Figura 16, modelo BeMicro Max10 10M08DAF484. Conforme pode-se constatar em (INTEL, 2018), o kit possui as seguintes especificações:

- Bloco ADC, diodo sensor de temperatura, RAM-onchip, memória flash de usuário e autoconfiguração não volátil. Permite expansão adicional de dois cabeçalhos de prototipagem de 40 pinos totalizando 80 pinos;
- Acelerómetro de 3- eixos da Analog Devices;

- Conversor Digital-para-Analógico de 12 bits da *Analog Devices*;
- Sensor de temperatura digital da *Analog Devices*;
- 8 MB de SDRAM;
- 8 LEs;
- Memória de 414 kbit (kb) M9K;
- Memória flash do usuário de 256 kbit (kb);
- *Loops* bloqueados de 2 fases (PLLs);
- 24 multiplicadores de 18x18 bits;
- 1 bloco ADC – 1 MSPS com 12 bits, 18 canais;
- 17 entradas analógicas;
- 1 sensor de temperatura;
- 250 entradas/saída para fins gerais (GPIO);
- Autoconfiguração não volátil com suporte a inicialização dupla;
- USB-Blaster™ embutido para uso com o programa Quartus® II. Este USB-Blaster também pode ser usado para programar dispositivos externos altera.

4.3 Geração de PWM implementado no FPGA

Para controlar o conversor de potência é necessário gerar pulsos com variação do período de tempo ligando e desligando e uma frequência de chaveamento constante. Já para a implementação do controle do conversor pode-se optar por um controle analógico ou digital. No caso em questão o utilizou-se controle digital, pela vantagem de baixo consumo e imunidade a ruído (PANDIT, 2017).

Conforme Pandit (2017) o PWM depende de um temporizador. Isto é, um relógio (mais conhecido como *clock*) interno e simples, que conta até certo número e depois volta para zero. Esse temporizador é definido via programação e gera o PWM com frequência definida pelo tempo que o pino fica em estado lógico alto até o momento em que o

temporizador chega a zero, ao terminar a contagem a saída lógica vai para zero, e um próximo período se inicia. Dessa forma, é possível ter um pino externo em nível lógico baixo, em seguida alto por uma quantidade de tempo específica via programação, repetindo-se o processo ciclicamente. Dessa maneira, é gerada uma forma de onda quadrada, onde o ciclo completo é denominado PWM. O mesmo princípio é válido para a geração da portadora, que no caso foi aplicada a portadora tipo dente de serra, onde é feito a contagem até um valor estabelecido ao mesmo tempo que é comparado com a moduladora. No Momento que os valores se igualam o pino de saída vai para estado lógico alto permanecendo até o término da contagem. O propósito é criar um sinal de controle com frequência e razão cíclica determinadas. A razão cíclica é a fração de tempo em que a onda fica em estado ativo.

O contador responsável pela geração da portadora e do sinal PWM possui um “tamanho” (e, portanto, a resolução do ciclo de serviço), de 10 bits, o que é igual para 1024 valores. Portanto o *clock* de 51,2 MHz (definindo-se os parâmetros de multiplicação de *clock* no *software Quartus II*) é dividido por 1024, pois um ciclo do contador leva 1024 pulsos de *clock*, nessas condições é gerado uma frequência de 50 kHz. Dentre esses 1024 pulsos pode-se definir por quanto tempo a saída ficara em estado logico alto estabelecendo a quantidade de pulos limite, esse valor “limite” é comparando com o valor do contador, no momento que o valor comparado for maior que a contagem a saída vai para “zero”. A fim de obter maiores frequências, podemos multiplicar o *clock* por um fator, conhecido como multiplicação do *clock*, da mesma forma pode ser feita a divisão de *clock* dividindo o *clock* por outro fator. Isso também pode ser chamado de pré-dimensionamento, porque precede o contador. A frequência de *clock* pode ser facilmente alterada, além disso, há a possibilidade de gerar diversas frequências distintas simultaneamente através da alteração dos parâmetros no *software Quartus II* seguindo os passos descritos em Intel FPGA (2018).

A nível de *hardware*, ao descrever um código em VHDL o FPGA sintetiza registradores e comparadores semelhantes aos dos microcontroladores, que são utilizados em diversas atividades como definir e criar a saída pulsante com razão cíclica. Assim o registrador define o período que a saída permanece “alta”. Por exemplo, para uma taxa de 50% (0,5) do ciclo, o registrador é definido para $((0,5-1) * 1024) = 512$, sendo assim este valor resultante será o valor “limite”. Portanto o comparador compara o valor do registrador com o valor atual do contador, se o valor atual for menor que o valor estabelecido a saída recebe um valor de nível lógico “alto” caso contrário a saída recebe nível logico “baixo”.

Em Apêndice D é apresentado uma visão geral após a sintetização. Cada bloco

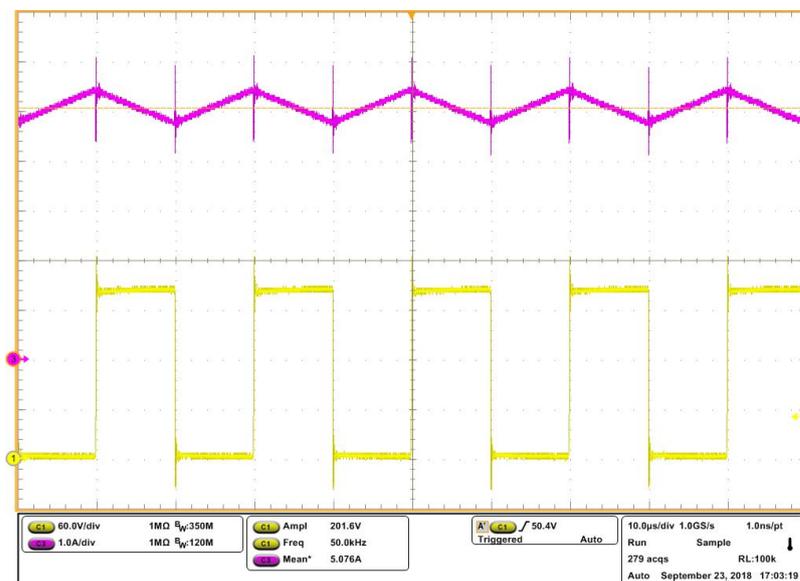
gerado simboliza que internamente existe um *hardware* real referente ao programa sintetizado. Os blocos são utilizados para melhorar a compreensão e organização e são interligados através de uma janela de código mestre, onde é descrito todas as interligações. Também é possível notar que em uma das extremidades encontra-se a entrada de clock e na outra a saída (GPIOA) que comanda o interruptor.

4.4 Conversor Boost operando em malha aberta

Inicialmente o conversor Boost foi implementado em malha aberta para análise de funcionamento. Foi estabelecida a razão cíclica de 50%, porém na prática há um retardo característico do interruptor no momento em que ela é aberta de aproximadamente 3%, portanto para razão cíclica de 50% no *gate*, o período em que o interruptor ficara aberta será de aproximadamente 53%. Há também um pequeno atraso no circuito do *gate*, sendo que esta característica deve ser observada em aplicações onde o conversor é projetado para operar em malha aberta. A defasagem varia de acordo com o interruptor escolhido e é especificada no manual do fabricante.

Na Figura 33 mostra-se a tensão no interruptor comprovando a teoria de que o período em que o interruptor se encontra fechada a tensão sobre o interruptor é zero e a corrente no indutor aumenta gradativamente, já no instante que o interruptor é aberto, a corrente no indutor diminui gradativamente até o próximo período.

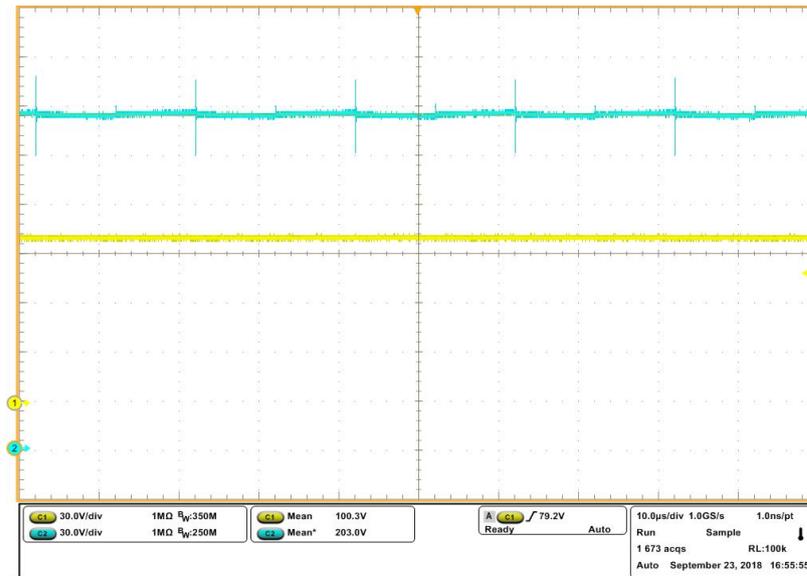
FIGURA 33 - Tensão no interruptor (amarelo) e corrente no indutor (roxo)



Fonte: Autoria própria (2018).

Na Figura 34 o sinal em amarelo representa a tensão de entrada e o sinal azul demonstra a tensão de saída, onde observa-se semelhança nos resultados de simulação assim como na análise teórica no qual afirma-se que o conversor Boost é um conversor elevador de tensão e que a tensão de saída deve ser o dobro da entrada quando razão cíclica for igual a 0,5.

FIGURA 34 - Tensão de entrada V_{in} (amarelo) e tensão de saída V_o (azul).



Fonte: Autoria própria (2018).

4.5 Conversor Boost operando em malha fechada

Para o controle da malha de corrente foi utilizado o sensor de corrente da fabricante LEM, modelos LTSR 25 NP. Com o valor de corrente lido pelo sensor, projeta-se o controlador PI de corrente, conforme detalhado no capítulo anterior.

Já o controlador PI de tensão foi inserido para que a tensão de saída se mantenha inalterada independente da carga, ou seja, independente da corrente na entrada do conversor. O sensor de tensão utilizado foi da fabricante LEM, modelo LV 25 NP.

Com os compensadores de corrente e tensão projetados, pode-se implementar os compensadores no FPGA.

O código do controle de corrente interno a malha de tensão representado na Figura 35 leva em conta a leitura de tensão e corrente do conversor Boost conforme ilustração da Figura 28, e tem como sinal de referência a saída do controle da malha de tensão, onde o erro é dado pela diferença entre a referência e leitura do AD de corrente.

Para digitalizar o controlador foi empregado o software Matlab, com a função “c2d” e o método de discretização “Tustin”. As diferenças implementada no FPGA foi obtida a partir da função de transferência discreta dada pela Equação (9) (SÁ, 2014):

$$FT_{discreta} = \frac{B0.erro(k) + B1.erro(k-1) + B2.erro(k-2)}{A0.pwm(k) + A1.pwm(k-1) + A2.pwm(k-2)} \quad (9)$$

Fazendo então o coeficiente $A0=1$, a saída dos controladores, conforme segue na Equação (10) (SÁ, 2014):

$$pwm_{atual} = B0.erro(k) + B1.erro(k-1) + B2.erro(k-2) - A1.pwm(k-1) - A2.pwm(k-2) \quad (10)$$

Os valores atribuídos as variáveis do controle PI de corrente discretizadas B0C, B1C, B2C, A1C e A2C, são respectivamente: 0,3918; 0,02385; - 0,3679; 1; - 0,5986 e 0,1706, de forma a serem determinados em razão da margem de fase que varia entre 45° e 90° como é apresentado por (Sá, 2014).

FIGURA 35 - Código implementado no controle de corrente do conversor.

```
refcorrente=pwm_atual_tensao;
erro_antigo_corrente=erro_anterior_corrente;
erro_anterior_corrente=erro_atual_corrente;
erro_atual_corrente=refcorrente - adcorrente;
pwm_antigo_corrente=pwm_anterior_corrente;
pwm_anterior_corrente=pwm_atual_corrente;
aux = (B0C*erro_atual_corrente) + (B1C*erro_anterior_corrente) ;
aux1= (B2C*erro_antigo_corrente) + (-A1C*pwm_anterior_corrente) -(A2C*pwm_antigo_corrente);
pwm_atual_corrente = aux + aux1;
```

Fonte: Autoria própria (2018).

Na Figura 36 é representado o código referente ao controlador PI de tensão. A leitura de tensão é feita na saída do conversor, sendo coletado o sinal analógico através do pino AIN1 do FPGA, em seguida é convertido para digital com resolução de 12 bits, armazenado em (AD tensão) e subtraídas com a tensão de referência, previamente definida pelo usuário, dessa forma se obtém o erro entre a tensão desejada e a tensão atual. Se o erro for diferente de zero haverá uma ação de controle, sendo armazenado em (pwm_atual_tensao) posteriormente esse valor é transmitido para outro bloco de programação responsável pela modulação onde é definido a saída com nível lógico alto ou baixo, que por sua vez comanda o interruptor.

Para digitalizar o controlador foi empregado o software Matlab, com a função “c2d” e

o método de discretização “Tustin”. A Equação a diferenças implementada no FPGA foi obtida a partir da função de transferência discreta dada pela Equação (11):

$$FT_{discreta} = \frac{B0.erro(k) + B1.erro(k-1)}{A0.pwm(k) + A1.pwm(k-1)} \quad (11)$$

Fazendo então o coeficiente $A0=1$, a saída dos controladores, conforme segue na Equação (12):

$$pwm_{atual} = B0.erro(k) + B1.erro(k-1) - A1.pwm(k-1) \quad (12)$$

No caso do controle de tensão as variáveis $B0$, $B1$ e $A1$ assumem as nomenclaturas $B0T$, $B1T$ e $A1T$, e neste caso recebem os respectivos valores 0,251, 0,248 e -1.

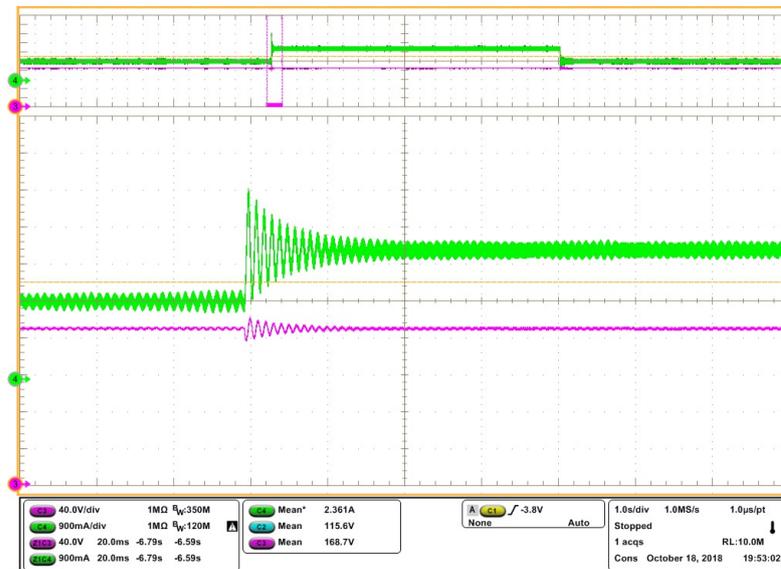
FIGURA 36 - Código implementado no controle de tensão do conversor.

```
erro_anterior_tensao=erro_atual_tensao;
erro_atual_tensao=reftensao - adtensao;
pwm_anterior_tensao=pwm_atual_tensao;
pwm_atual_tensao=(B0T*erro_atual_tensao)+(B1T*erro_anterior_tensao)-(A1T*pwm_anterior_tensao);
```

Fonte: Autoria própria (2018).

Dessa forma, a Figura 37 apresenta o resultado experimental no instante do degrau de carga, primeiramente apenas com a malha de tensão inserida. Nota-se neste caso, uma rápida oscilação de tensão (em roxo) causada pela ação de controle conhecida como sistema subamortecido. A oscilação pode ser facilmente controlada de acordo com a necessidade bastando apenas alterar os valores da constante proporcional e/ou integral. Na Figura 37 o sinal em verde representa a corrente no indutor e o sinal em roxo representa a tensão na carga.

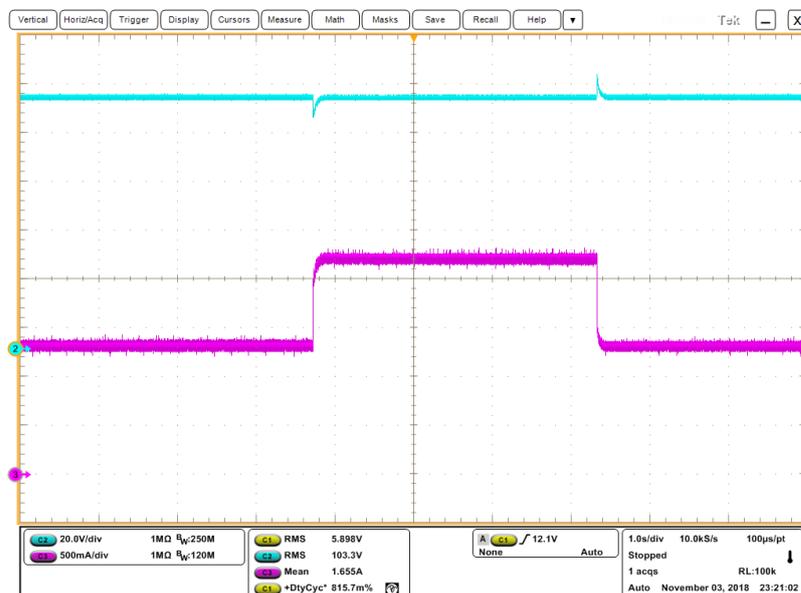
FIGURA 37 - Formas de onda experimentais do controle de tensão PI.



Fonte: Autoria própria (2018).

Posteriormente as malhas de tensão e corrente em cascata são implementadas, temos então neste caso um sistema criticamente amortecido conforme podemos constatar na Figura 38. As principais vantagens de termos uma malha de corrente num conversor Boost é o aumento da percepção de variação de carga, pelo fato do sensor de corrente ter maior agilidade na percepção de oscilações de carga, por não sofrer influência de componentes internos como do período de descarga do capacitor. Outra grande vantagem é a ausência de oscilação na corrente e tensão nos instantes de variação de carga.

FIGURA 38 - Corrente de entrada e tensão de saída com o controlador PI.

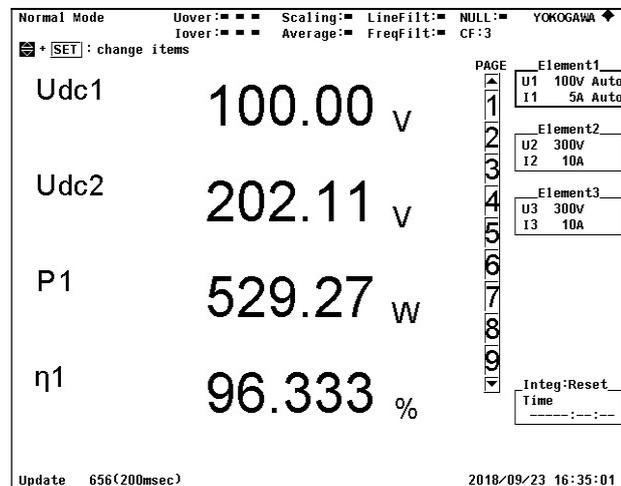


Fonte: Autoria própria (2018).

4.6 Análise de rendimento do conversor Boost

Para a realização da análise de rendimento utilizou-se o analisador de potência *yokogawa wt500* com intuito de comprovar a eficiência do conversor. Adotou-se a razão cíclica de 50% na base do transistor MOSFET modelo 24N60C3. Na Figura 39 pode-se observar o rendimento do conversor em potência nominal, obtendo-se eficiência máxima de 96,3%. A análise detalhada dos testes do rendimento nos diferentes pontos amostrados pode ser observada no Apêndice A.

FIGURA 39 - Tela do analisador de potência para os testes de rendimento.

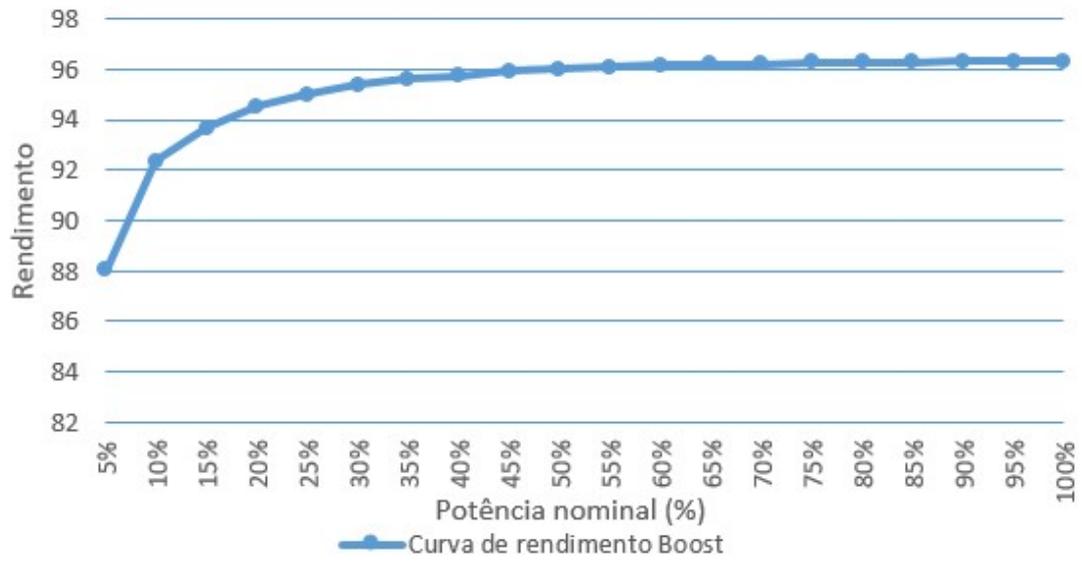


Fonte: Autoria própria (2018).

A Figura 40, apresenta a curva de rendimento real, sendo possível notar eficiência inferior ao resultado da simulação, onde foi simulado apenas as perdas nos semicondutores, dessa foram comprovando que além das perdas nos semicondutores existe as perdas no indutor conforme rege a Equação (2).

Na realização dos testes de rendimento foram coletadas amostras com intervalos de 5% da potência até chegar na potência nominal, e com o auxílio do software Matlab foi possível levantar a curva de rendimento.

FIGURA 40 - Rendimento experimental do protótipo do conversor Boost construído.



5. CONCLUSÃO

Por meio deste trabalho pode-se estudar os microcontroladores e suas aplicações. Dos microcontroladores estudados pode-se dividi-los por meio dos métodos e capacidade de processamento, sendo que cada modelo abordado no trabalho possui particularidades específicas que atendem uma infinidade de funções. Assim destaca-se a necessidade do conhecimento dos diferentes microcontroladores para evitar desperdício de recurso.

Dentre os microcontroladores escolhidos, pode-se perceber que o Arduino é indicado para propósitos didáticos, pois se trata de uma plataforma de fácil entendimento, onde o usuário pode desenvolver projetos sem se preocupar com hardware. Porém, o mesmo possui baixa confiabilidade devido ao seu tamanho relativamente grande e suas conexões podem apresentar falhas. O Arduino foi desenvolvido para fins didáticos, portanto de forma geral não é indicado para aplicações industriais, porém seu microcontrolador Atmega328 pode ser empregado industrialmente.

Já os microcontroladores da família PIC são microcontroladores semelhantes a família Atmega, onde são muito utilizados em aplicações com baixo nível de complexidade. Existem atualmente no mercado diversos modelos de PIC no qual podem ser adquiridos conforme a necessidade de uma específica finalidade. A principal diferença entre o Arduino e o PIC é a construção física, sendo que o Arduino é uma plataforma didática e o PIC é um microcontrolador que pode ser embarcado em diversos equipamentos eletrônicos.

Portanto, os microcontroladores mais simples como PIC e o ESP32 são usados para controle de processos mais simples e para diminuir a sobrecarga de microcontroladores principais. O ESP32 possui a vantagem de ser dual core sendo uma boa alternativa para aplicação na eletrônica de potência além de possuir integrado *bluetooth* e *WI-FI*, que pode ser uma grande vantagem para aplicações como a geração fotovoltaica residencial, além de possuir desempenho relativamente alto.

O DSP e o FPGA requerem maior conhecimento em eletrônica e engenharia, pois são processadores de alto desempenho, e são muito utilizados para processamento de som, imagem e dados em tempo real. O FPGA está no topo da hierarquia, pois ele pode ser arranjado de forma a assumir funções de qualquer outro processador, além de possuir capacidade de processar grande volume de dados paralelamente.

O estudo dos microcontroladores é de grande valia, pois os modernos conversores

de potência, sistemas de comunicação, reconhecimento de padrões, processamento e sistemas de transmissão de som e imagem, e demais equipamento eletrônicos possuem alta densidade de informações à serem tratadas, exigindo cada vez mais dos dispositivos de processamento para suas funções sejam desempenhadas da melhor forma possível. Além disso o custo dos modernos processadores decresce em passos acelerados, e constantemente novas tecnologias são lançadas. Por esta razão, existe a necessidade da migração para dispositivos modernos que possuam maior capacidade de processamento.

É importante ressaltar que existem diversos microcontroladores com os mais variados tipos de arquitetura, topologias empregadas, modos de acesso, hardware, modos de programação, entre outras características. Portanto cada caso exige um determinado microcontrolador. Como por exemplo, o trabalho em questão, onde poderia ter sido utilizado qualquer outro microcontrolador mais em conta que o FPGA para o controle do conversor Boost, tendo apenas uma pequena variação no tempo de resposta e na resolução da leitura analógica, sendo quase imperceptível. Porém, o FPGA é o controlador mais indicado para processamento de grande quantidade de sinais que exigem processamento em tempo real, processamento paralelo e também é uma das mais nova e poderosa tecnologias no mercado. Por este motivo optou-se por demonstrar a aplicação do FPGA no controle do conversor Boost com intuito de ampliar os conhecimentos e futuramente aplicar o FPGA no gerenciamento de diversas atividades simultâneas na área de geração e conversão de energia.

Quanto ao estudo, análise e implementação do controle do conversor Boost tornou-se fundamental na aplicação prática do FPGA escolhido. Assim, após a análise teórica e simulação do conversor foi possível implementá-lo experimentalmente. Os resultados em malha aberta obtidos nos testes práticos foram semelhantes aos dados teóricos e a simulação, validando assim o protótipo construído. Já os controles de corrente na entrada e da tensão na saída do conversor obtiveram bons resultados, conforme esperado, principalmente com as malhas de controle de corrente e tensão em cascata.

Finalmente, considera-se o conversor Boost é uma peça fundamental para aplicações futuras, além de ter contribuído de forma significativa para aprimoramento dos conhecimentos na implementação do controle e descrição de hardware, certamente os conhecimentos adquiridos poderão ser empregados nas mais diversas aplicações envolvendo conversores estáticos, inversor de frequência, acionamentos elétricos, carros elétricos e uma infinidade de dispositivos que necessitam de controle microcontrolado sendo uma área extremamente vasta.

REFERÊNCIAS

ALLAFI I.; Iqbal T., **Design and Implementation of a Low Cost Web Server Using ESP32 for Real-Time Photovoltaic System Monitoring**, IEEE Electrical Power and Energy Conference (EPEC), pg. 1 – 5, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8286184>>. Acesso em out. 2018.

ANDRIANSYAH M., SUBALI M., PURWANTO I., S. A. Irianto, R. A. Pramono **e-KTP as the basis of home security system using arduino UNO**, IEEE, 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), pg. 1– 5, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8320693>>. Acesso em fev. 2018.

ARDUINO. **Arduino**. Disponível em: <<https://store.arduino.cc/usa/arduino-uno-rev3>> Acesso em fev. 2018.

BAIG M. Q., MAQSOOD J., ALVI H., KHAN K, **A Comparative Analysis on Home Automation Techniques**, IEEE, International Conference on Artificial Intelligence, Modelling and Simulation, pg. 109 – 114, 2014. Disponível em: <<https://ieeexplore.ieee.org/document/7102444>>. Acesso em fev. 2018.

BARBI, Ivo. **Eletrônica de potência: projetos de fontes chaveadas**. 3ª ed. Ed. do autor, 2001.

BASTSCHAUER, Alessandro Luiz; NETO, Anis cézar Chehab; PETRY, Clóvis Antônio **Conversor Boost operando em condução contínua aplicado à correção de fator de potência empregado controle de corrente de entrada por valores médios instantâneos**, pg. 13-15, 2000 Disponível em: <http://www.joinville.udesc.br/portal/professores/mezaroba/materiais/Boost_600W.pdf>. Acesso em out. 2018.

BASTSCHAUER, Alessandro Luiz. **Apostila da Disciplina de Controle de Conversores Estáticos**, pg. 50 – 225, 2012. Disponível em: <http://www.joinville.udesc.br/portal/professores/batschauer/materiais/Apostila_CCE_v7__si_mobologia_.pdf>. Acesso em set. 2018.

BEGHINI, Lucas Bragazza, **Automação residencial de baixo custo por meio de dispositivos móveis com sistema operacional Android** pg. 22, 2013. Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-04022014-152853/?&lang=br>>.

Acesso em ago. 2018.

BROCKVELD, Sergio Luis Junior. **Conversor cc-cc Boost-flyback com divisão de esforços de corrente no estágio de entrada**, Dissertação, pg. 43 – 45, 2017 Disponível em: <https://repositorio.ufsc.br/handle/123456789/176745> Acesso em out. 2018.

DE0-NANO, **User manual**. Disponível em: <<http://www.ti.com/lit/ug/tidu737/tidu737.pdf>>. Acesso em: out. 2018.

DSP TMS320F28379D, **Texas Instruments**, Disponível em: <<http://www.ti.com/product/TMS320F28379D>>. Acesso em nov. 2018.

EPCOS. **Film Capacitors**, Metallized Polypropylene Film Capacitors (MKP), datasheet, Disponível em: <<https://media.digikey.com/pdf/Data%20Sheets/Epcos%20PDFs/B32794-B32798.pdf>>. Acesso em nov. 2018.

ERICKSON, R. Fundamentals of Power Electronics. [S.1]: USA: Chapman & Hall, 1997. (Lectures on Power Electronics).

FILIFELOP. **Qual Arduino Comprar? Conheça os Tipos de Arduino**, 2018 Disponível em: <www.filieflop.com/blog/tipos-de-arduino-qual-comprar>. Acesso em fev. 2018.

GANGULY S., A. SADHUKHAN, P. K. GAYEN, S. DOLUI, **Hardware Implementation of Single-Phase Fullbridge VSI Switched by PSO Based SHEPWM Signals Using Embedded PIC Microcontrolle**, IEEE Calcutta Conference (CALCON), pg. 341 – 346, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8280752>>. Acesso em fev. 2018.

HOLDEFER, Antônio Eliseu, **Controle digital de retificador trifásico utilizando o controlador tms320lf2407**, Dissertação, 2004. Disponível em: <<https://repositorio.ufsc.br/bitstream/handle/123456789/86876/211748.pdf?sequence=1&isAllowed=y>> Acesso em nov. 2018

INDIAMART, **PIC16F676 Microcontroller**. Disponível em: <<https://www.indiamart.com/proddetail/pic16f676-microcontroller-16153273162.html>>.

Acesso em: Nov. 2018.

INTEL, FPGA **BeMicro Max 10**, manual descritivo. Disponível em: <https://fpgawiki.intel.com/wiki/BeMicro_Max_10>. Acesso em Nov 2018.

JAIN R. V.; M. V. AWARE; A. S. JUNGHARE, **Implementation of a PID control PWM Module on Altera DE0 Kit Using FPGA**, First International Conference on Control, Measurement and Instrumentation (CMI), pg. 341 – 345, 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7413767>>. Acesso em fev. 2018.

KALE A. V.; Bankar S. A.; Jagtap S. R. **Design of PIC Microcontroller-Based Data Acquisition Module with Lab VIEW Interfacing**, International Conference on Communication and Signal Processing, pg. 858 – 861, 2014. Disponível em: <<https://ieeexplore.ieee.org/document/6949965>>. Acesso em: fev. 2018.

KAMISAN M, A. A. Aziz, AHMAD W. R. W. , KHAIRUDIN N., **UiTM campus bus tracking system using Arduino based and smartphone application**, IEEE 15^a Student Conference on Research and Development , pg. 137-141 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8305406>>. Acesso em fev. 2018.

LANGE, André De Bastiani, **Retificador pfc monofásico pwm bridgeless três-níveis de alto desempenho**, Tese mestrado, 2012. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/100858>>. Acesso em agt. 2018.

MAIER A.; SHARP A.; VAGAPOV Y. **Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things**, Internet Technologies and Applications (ITA), pg. 143 – 148, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8101926>>. Acesso em: Jan 2018.

MAKWANA S. D., TANDON A. G., **Touch Screen Based Wireless Multifunctional Wheelchair Using ARM and PIC Microcontroller**, IEEE, International Conference on Microelectronics, Computing and Communications (MicroCom), pg. 1 – 4, 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7522524>>. Acesso em fev. 2018.

MARTINS, Denizar Cruz; BARBI, Ivo. **Eletrônica de potência: Conversores CC-CC básicos não isolados**. Florianópolis, pg. 67 – 110, 2^a ed. Ed. dos autores, 2006.

MARTINS, Nardênio Almeida. **Sistemas Microcontrolados**, Uma Abordagem com o

Microcontrolador PIC 16F84, pg. 14-20, 2005. Disponível em: <www.martinsfontespaulista.com.br/anexos/produtos/capitulos/203289.pdf>. Acesso em Out. 2018.

MICROCHIP **PIC18F2420/2520/4420/4520 Data Sheet** Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/39631a.pdf>> Acesso em fev. 2018.

NOOR M. Z. H., BAHARUDDIN M. S, RAHIMAN M. H. F., TAIB M. N., MD DOM M. S. **Design and Development of Automatic Rabbit Feeding System (ARFES) using PIC Microcontroller** IEEE Symposium on Industrial Electronics & Applications (ISIEA), pg. 153 – 156, 2014. Disponível em: <<https://ieeexplore.ieee.org/document/8049889>>. Acesso em fev. 2018.

PANDIT S., SHET V. N. **Review of FPGA based Control for Switch Mode Converters**, Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), pg. 1 – 5, 2017.

PANTECHSOLUTIONS **TMS320F2812 DSP starter kit**, 2018 Disponível em: <www.pantechsolutions.net/dsp-dsc-boards/tms320f2812-dsp-starter-kit>. Acesso em fev. 2018.

PENIDO É. C. C.; TRINDADE R. S., **Microcontroladores**, Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, 2013. Disponível em: <http://estudio01.proj.ufsm.br/cadernos/ifmg/tecnico_automacao_industrial/microcontroladores.pdf> Acesso em: nov. 2018.

PURI V., NAYYAR A., **Real time smart home automation based on PIC microcontroller, Bluetooth and Android technology**, IEEE, 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pg. 1478 – 1484, 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7724513>>. Acesso em abr. 2018.

RAIHAN S. R. S.; RAHIM N. A. **Modeling of FPGA-based Pulse-Width Modulation for Parallel Three-phase AC/DC Converters** International Conference for Technical Postgraduates (TECHPOS), pg. 1 – 3, 2009. Disponível em: <<https://ieeexplore.ieee.org/document/5412108>>. Acesso em fev. 2018.

RAJALAKSHMI A.; SHAHNASSER H. **Internet of Things using Node-Red and Alexa**,

17th International Symposium on Communications and Information Technologies (ISCIT), pg. 1 – 4, 2017. Disponível em <<https://ieeexplore.ieee.org/document/8261194>>. Acesso em: fev. 2018.

RAJKANNA U., MATHANKUMAR M., GUNASEKARANK., **Hand Gesture Based Mobile Robot Control Using PIC Microcontrolle**, International Conference on Circuits, Power and Computing Technologies, pg. 1687 – 1691, 2014. Disponível em: <<https://ieeexplore.ieee.org/document/8117996>>. Acesso em abr. 2018.

RAJU Y. , MOHAMMED S., PAUL S., JOHN G. A., NAIR D. S., **Development and implementation of arduino microcontroller based dualmode fire extinguishing robot**, IEEE, International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), pg. 1-4 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8303141>>. Acesso em out. 2018.

ROCHA R. C., **Projeto de controlador de temperatura para mantas termoeletricas**, Trabalho de conclusão de curso, pg. 7 – 15, 2012. Disponível em: <<http://lyceumonline.usf.edu.br/salavirtual/documentos/2339.pdf>>. Acesso em fev. 2018.

RUGE I. A., **Metodo basico para implementar un controlador digital pid en un microcontrolador pic para desarrollo de aplicaciones a bajo costo**, pg. 4, 2009. Disponível em: <http://www.edutecne.utn.edu.ar/microcontrol_congr/industria/MTODOB~1.PDF>. Acesso em abr. 2018.

SÁ, Francieli Lima de **Conversores CC-CC com elevada taxa de conversão estática**, Tese, 2014. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/74645>> Acesso em mar. 2018.

SÁ, Francieli L. de; Caio V. B. Eiterer; Domingo Ruiz-Caballero; Samir A. Mussa **Double Quadratic Buck Converter** Brazilian Power Electronics Conference, pg. 36 – 43, 2013. Disponível em: <<https://ieeexplore.ieee.org/document/6785092>>. Acesso em out. 2018.

SANTOS, Israel Joaquim dos. **Aplicação de um sistema fotovoltaico conectado à rede**, um estudo de caso, pg. 40 – 49, 2018.

SILVA S.; SOARES S.; VALENTE A.; SYLVAIN T. M., **digital sound processing using arduino and matlab**, Science and Information Conference (SAI), pg. 1184 – 1191, 2015.

Disponível em: <<https://ieeexplore.ieee.org/document/7237295>>. Acesso em fev. 2018.

SOH, M. Z. H. C. ISMAIL H., OTTHAMAN F. H., SAFIE M. K., ZUKRI M. A. A., ABDULLAH S. A. C., **Development of automatic chicken feeder using Arduino Uno**, IEEE, International Conference on Electrical, Electronics and System Engineering (ICEESE), pg. 120 – 124, 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8298402>>. Acesso em abr. 2018.

THORNTON. **Parâmetros Efetivos do Núcleo**. Disponível em: <http://www.thornton.com.br/pdf/nt_C.pdf> Acesso em agt. 2018.

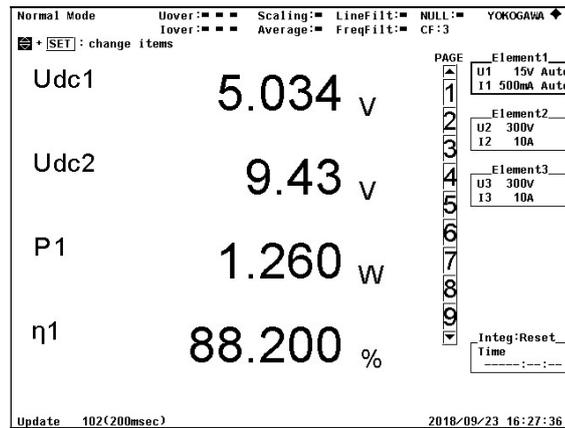
VISHAY. **Metallized Polypropylene Film Capacitors DC-Link Capacitor**, Datasheet. Disponível em: <<https://www.vishay.com/docs/28164/mkp1848dcl.pdf>>. Acesso em nov. 2018.

WOLTER, S. K. **PROCESSADORES DE SINAIS DIGITAIS – DSP**, Universidade Federal do Paraná Departamento de Engenharia Elétrica Processamento Digital de Sinais, 2017. Disponível em: <<http://www.eletrica.ufpr.br/marcelo/TE810/012007/Stefan-DSP>>. Acesso em mar. 2018.

WT500. **WT500 Power Analyzer**. Disponível em: <http://www.testequipmenthq.com/datasheets/YOKOGAWA-WT500-Datasheet.pdf>. Acesso em: set.2018.

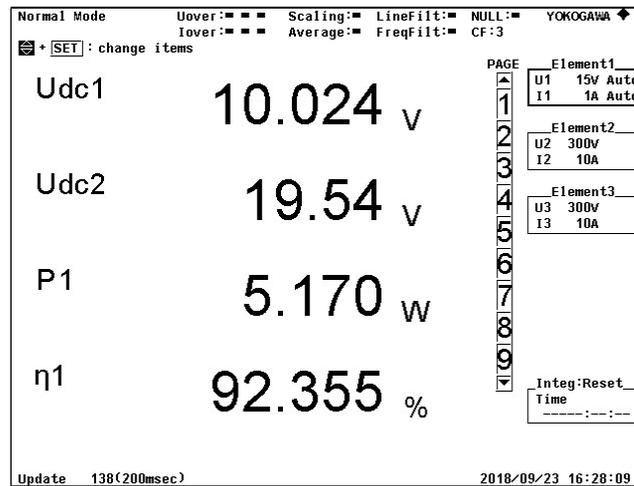
APÊNDICE A - ANÁLISE DE RENDIMENTO DO CONVERSOR BOOST

FIGURA 41 - Análise rendimento com 5% da potência nominal



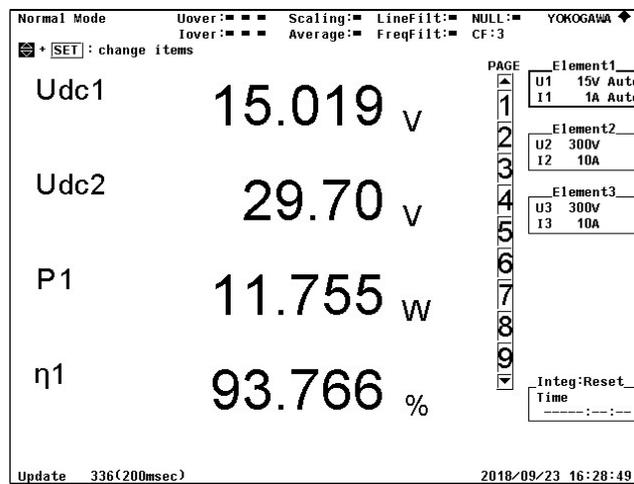
Fonte: Autoria própria (2018).

FIGURA 42 - Análise rendimento com 10% da potência nominal



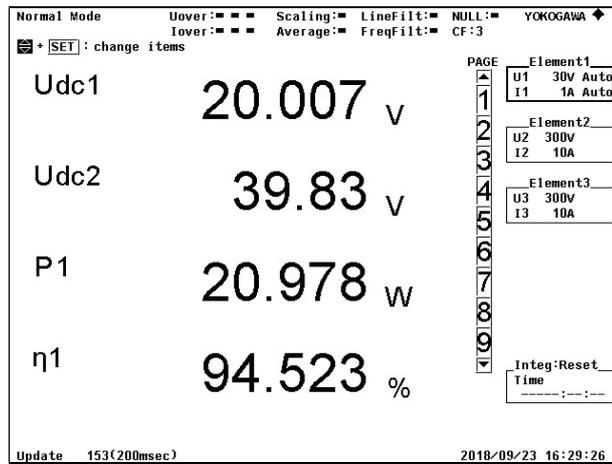
Fonte: Autoria própria (2018).

FIGURA 43 - Análise rendimento com 15% da potência nominal



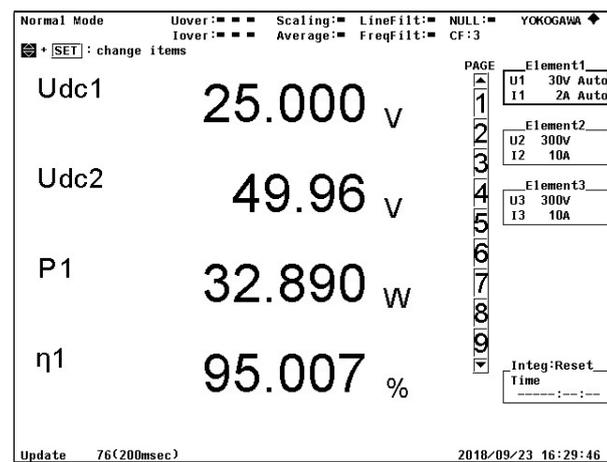
Fonte: Autoria própria (2018).

FIGURA 44 - Análise rendimento com 20% da potência nominal



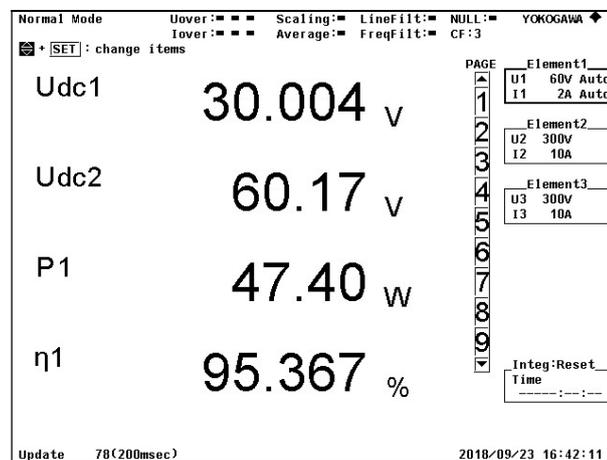
Fonte: Autoria própria (2018).

FIGURA 45 - Análise rendimento com 25% da potência nominal



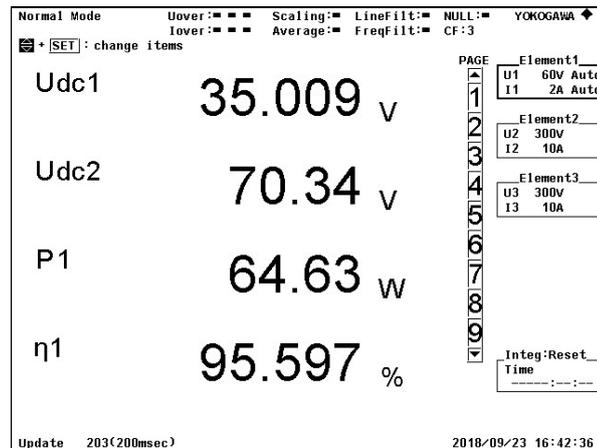
Fonte: Autoria própria (2018).

FIGURA 46 - Análise rendimento com 30% da potência nominal



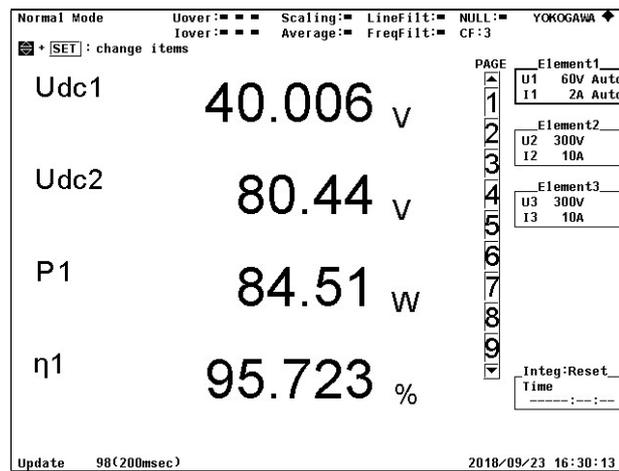
Fonte: Autoria própria (2018).

FIGURA 47 - Análise rendimento com 35% da potência nominal



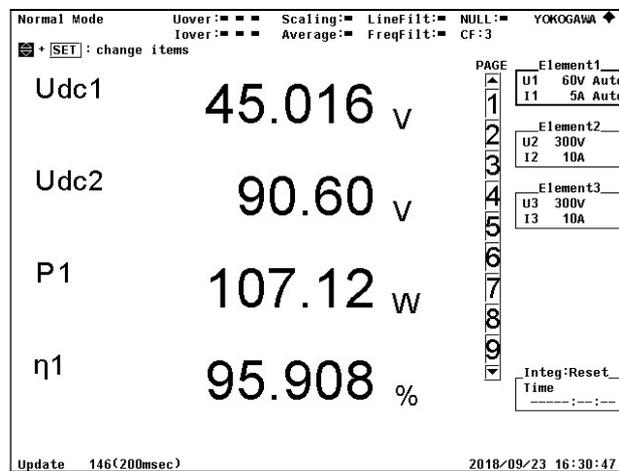
Fonte: Autoria própria (2018).

FIGURA 48 - Análise rendimento com 40% da potência nominal



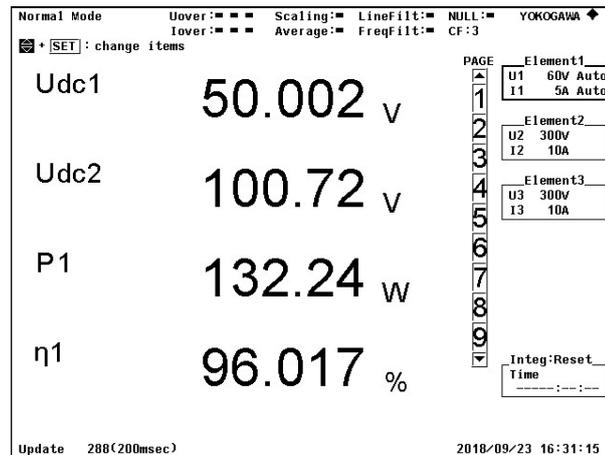
Fonte: Autoria própria (2018).

FIGURA 49 - Análise rendimento a 45% da potência nominal



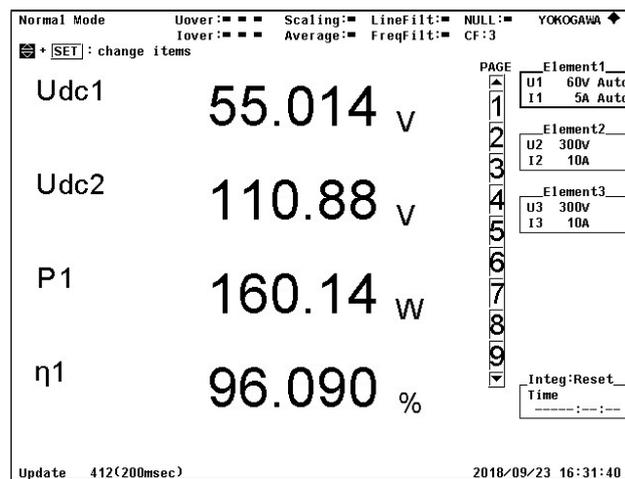
Fonte: Autoria própria (2018).

FIGURA 50 - Análise rendimento com 50% da potência nominal



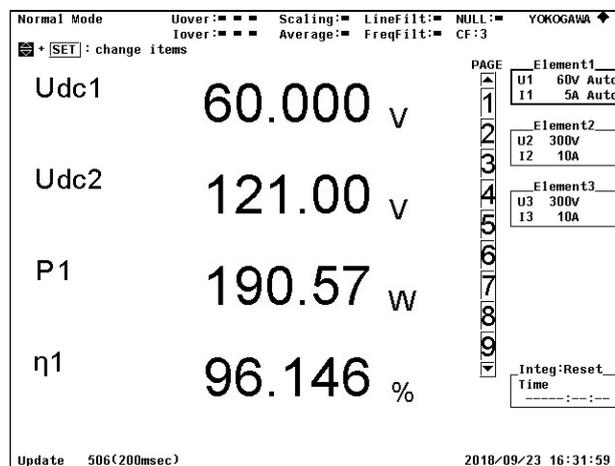
Fonte: Autoria própria (2018).

FIGURA 51 - Análise rendimento com 55% da potência nominal



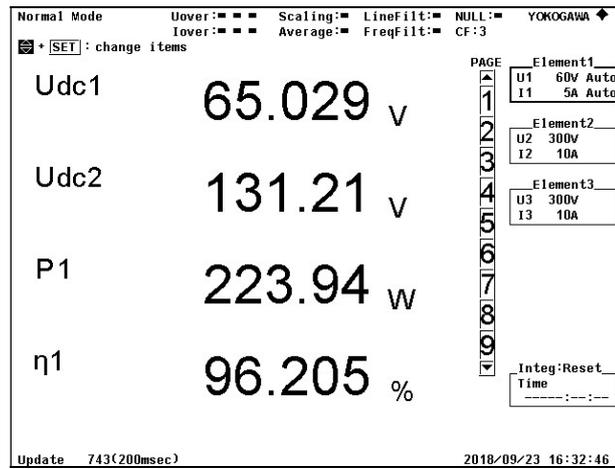
Fonte: Autoria própria (2018).

FIGURA 52 - Análise rendimento com 60% da potência nominal



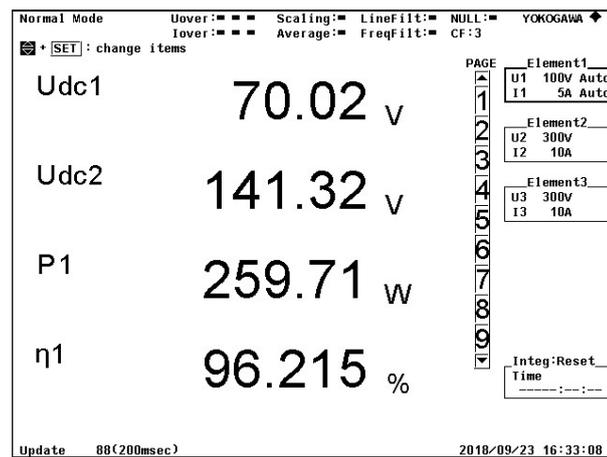
Fonte: Autoria própria (2018).

FIGURA 53 - Análise rendimento com 65% da potência nominal



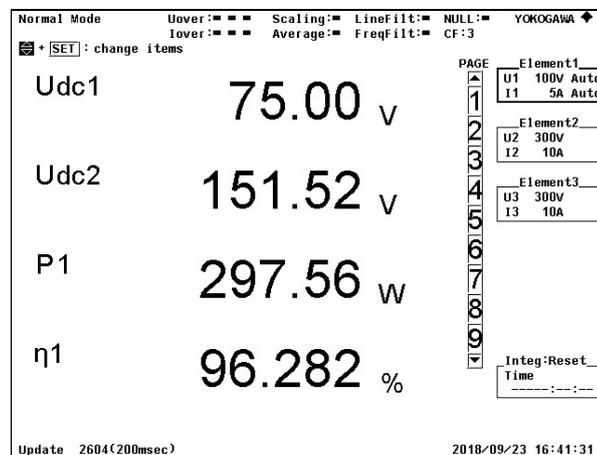
Fonte: Autoria própria (2018).

FIGURA 54 - Análise rendimento com 70% da potência nominal



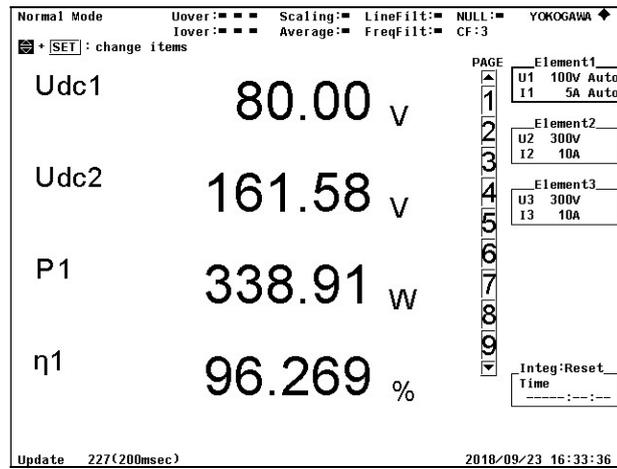
Fonte: Autoria própria (2018).

FIGURA 55 - Análise rendimento com 75% da potência nominal



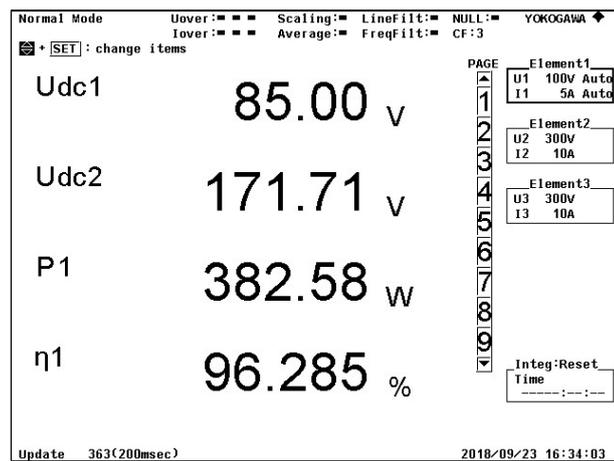
Fonte: Autoria própria (2018).

FIGURA 56 - Análise rendimento com 80% da potência nominal



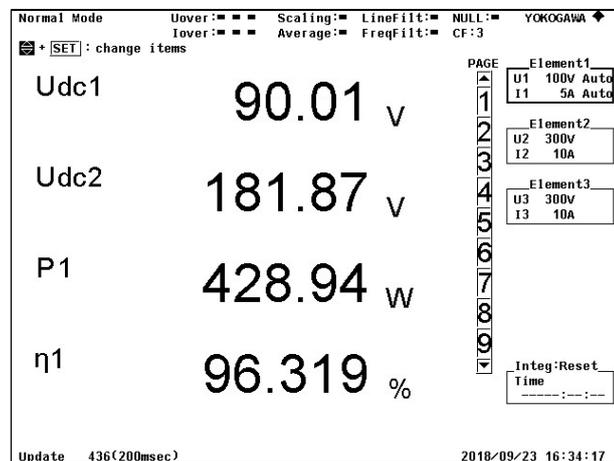
Fonte: Aatoria própria (2018).

FIGURA 57 - Análise rendimento com 85% da potência nominal



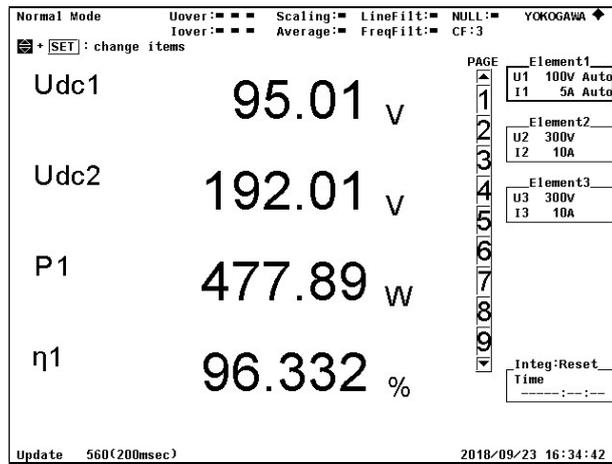
Fonte: Aatoria própria (2018).

FIGURA 58 - Análise rendimento com 90% da potência nominal



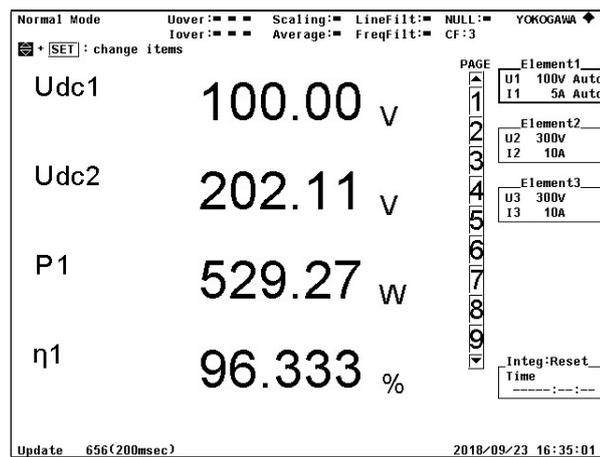
Fonte: Aatoria própria (2018).

FIGURA 59 - Análise rendimento com 95% da potência nominal



Fonte: Autoria própria (2018).

FIGURA 60 - Análise rendimento com 100% da potência nominal



Fonte: Autoria própria (2018).

APÊNDICE B - PROJETO DOS PARÂMETROS DO CONVERSOR BOOST



Centro Universitário Facvest Projeto de Eletrônica de Potência

Professora: Franciéli Lima de Sá, Dra.
 Aluno: Cleiton Dal Agnol
 Disciplina: TCC 2
 Curso: Engenharia Elétrica

Conversor Boost

Projeto dos Parâmetros do Conversor Boost (MARTINS; BARBI, 2006)

Lages -SC - Setembro de 2018

Definição dos principais parâmetros:

$$V_{in} := 100V$$

tensão de entrada total

$$D := 0.5$$

razão cíclica

$$P_{in} := 500W$$

potência de entrada total

$$I_{in} := \frac{P_{in}}{V_{in}}$$

$$I_{in} = 5 A$$

corrente de entrada

$$f_{chav} := 50kHz$$

frequência de operação do conversor

$$T_s := \frac{1}{f_{chav}}$$

$$T_s = 2 \times 10^{-5} s$$

período de operação do conversor

$G := \frac{1}{1-D}$	$G = 2$	ganho estático
$V_0 := G \cdot V_{in}$	$V_0 = 200 \text{ V}$	tensão no capacitor 1
$P_0 := P_{in}$	$P_0 = 500 \text{ W}$	potência de saída
$I_0 := \frac{P_0}{V_0}$	$I_0 = 2.5 \text{ A}$	corrente de saída
$R_0 := \frac{V_0}{I_0}$	$R_0 = 80 \Omega$	resistência de carga

Cálculo dos indutores a partir das ondulações de corrente

Cálculo de L_1 :

$\Delta I_{L1} := 10\% \cdot I_{in}$	$\Delta I_{L1} = 0.5 \text{ A}$	ondulação de corrente em L1
$L_1 := \frac{V_{in} \cdot D \cdot T_s}{\Delta I_{L1}}$	$L_1 = 2 \times 10^{-3} \text{ H}$	valor do indutor L1
		valor do indutor L4

Cálculo do capacitor de saída

Cálculo de C_0 :

$\Delta V_0 := 1\% \cdot V_0$	$\Delta V_0 = 2 \text{ V}$	ondulação da tensão de saída
$C_0 := \frac{I_0 \cdot D \cdot T_s}{\Delta V_0}$	$C_0 = 1.25 \times 10^{-5} \text{ F}$	valor do capacitor C0

Cálculo do resistor de saída

$R_o := \frac{V_o^2}{P_o} =$	$R_o = 80 \Omega$	valor do resistor R0
------------------------------	-------------------	----------------------

Cáculo Físico do Indutor Boost L1 (LANGE 2012)

$$L_1 = 2 \times 10^{-3} \text{ H}$$

Produto Especificado: Núcleo Toroidal Pó de Ferro
Magmattec

Código da Peça- 1.01.0033

Produto - MMT034T4416

Número de espiras:

$$A_L := 50 \cdot 10^{-9} \text{ H}$$

$$N := \sqrt{\left(\frac{L_1}{A_L}\right)}$$

$$N = 200$$

$$N_e := \text{ceil}\left[\sqrt{\left(\frac{L_1}{A_L}\right)}\right]$$

$$N_e = 200$$

Para o projeto é assumido o número inteiro de espiras

$$L_e := 11.2 \cdot 10^{-2} \text{ m}$$

$$I_{in} = 5 \text{ A}$$

$$\mu_r := 33$$

$$\mu_0 = 1.257 \times 10^{-6} \frac{\text{kg} \cdot \text{m}}{\text{A}^2 \cdot \text{s}^2}$$

$$B_{DC} := \frac{\mu_r \cdot \mu_0 \cdot N \cdot I_{in}}{L_e}$$

$$B_{DC} = 0.37 \text{ T}$$

$$B_{sat} := 1.1 \text{ T}$$

$$N_{maxB} := \frac{0.8 \cdot B_{sat} \cdot L_e}{\mu_0 \cdot \mu_r \cdot I_{in}}$$

$$N_{maxB} = 475.343$$

$$n := \frac{A_{\text{cobre}}}{S_{\text{fio}} \text{ cm}^2} \quad n = 5.131$$

$$n_{\text{cond}} := \text{ceil} \left(\frac{A_{\text{cobre}}}{S_{\text{fio}} \text{ cm}^2} \right) \quad n_{\text{cond}} = 6$$

Possibilidade de execução:

$$K_w = 0.4$$

$$D_{\text{cobre}} := 0.45 \text{ mm}$$

$$D_{\text{int_toroide}} := 27.2 \text{ mm}$$

$$N_{\text{max_esp}} := K_w \cdot \frac{D_{\text{int_toroide}}^2}{\left[(n_{\text{cond}}) \cdot (D_{\text{cobre}})^2 \right]}$$

$$N_{\text{max_esp}} = 243.569$$

Ok! Pode ser executado ($N_e \cdot n_{\text{cond}} < N_{\text{max_esp}}$)

Comprimento do fio de cobre:

$$H := 16.5 \cdot 10^{-1} \text{ cm}$$

$$\text{Lar} := 44.5 \cdot 10^{-1} \text{ cm} - 27.2 \cdot 10^{-1} \text{ cm} \quad \text{Lar} = 0.017 \text{ m}$$

$$\text{MLT} := (2 \cdot H) + (2 \cdot \text{Lar}) \quad \text{MLT} = 0.068 \text{ m}$$

$$L_{\text{enroll}} := \text{MLT} \cdot N_e$$

$$L_{\text{enroll}} = 13.52 \text{ m}$$

Comprimento Total do Enrolamento do Indutor

$$L_{\text{enrol}} := \frac{(L_{\text{enroll}})}{100}$$

$$L_{\text{enrol_total}} := (L_{\text{enroll}}) \cdot n_{\text{cond}}$$

$$L_{\text{enrol_total}} = 81.12 \text{ m}$$

Dados:

Indutância:	$L_1 = 2 \times 10^{-3} \text{ H}$
Indução máxima:	$B_{\max} := 1.1 \text{ T}$
Fator de ocupação:	$K_w := 0.4$
Densidade de corrente:	$J_{\max} := 600 \frac{\text{A}}{\text{cm}^2}$

Seção dos Condutores:

$$S_{Lm1} := \frac{I_{\text{in}}}{J_{\max}} \quad S_{Lm1} = 8.333 \times 10^{-7} \text{ m}^2$$

$$S_{Lm} := \frac{(S_{Lm1})}{\text{cm}^2} \quad S_{Lm} = 8.333 \times 10^{-3}$$

Cálculo da bitola dos condutores:**Penetração máxima:**

$$\Delta := \frac{7.5}{\sqrt{f_{\text{chav}}}}$$

$$2 \cdot \Delta = 0.067 \text{ s}^{0.5}$$

Para este coeficiente de penetração a bitola máxima do fio é:

AWG25

$$S_{\text{fio}} := 0.001624 \text{ cm}^2$$

**Tabela de Fios Esmaltados-livro:
Projeto de Fontes Chaveadas**

$$S_{\text{fio_isolado}} := 0.002078$$

$$\rho_{\text{fio}} := 0.001419 \frac{\Omega}{\text{cm}}$$

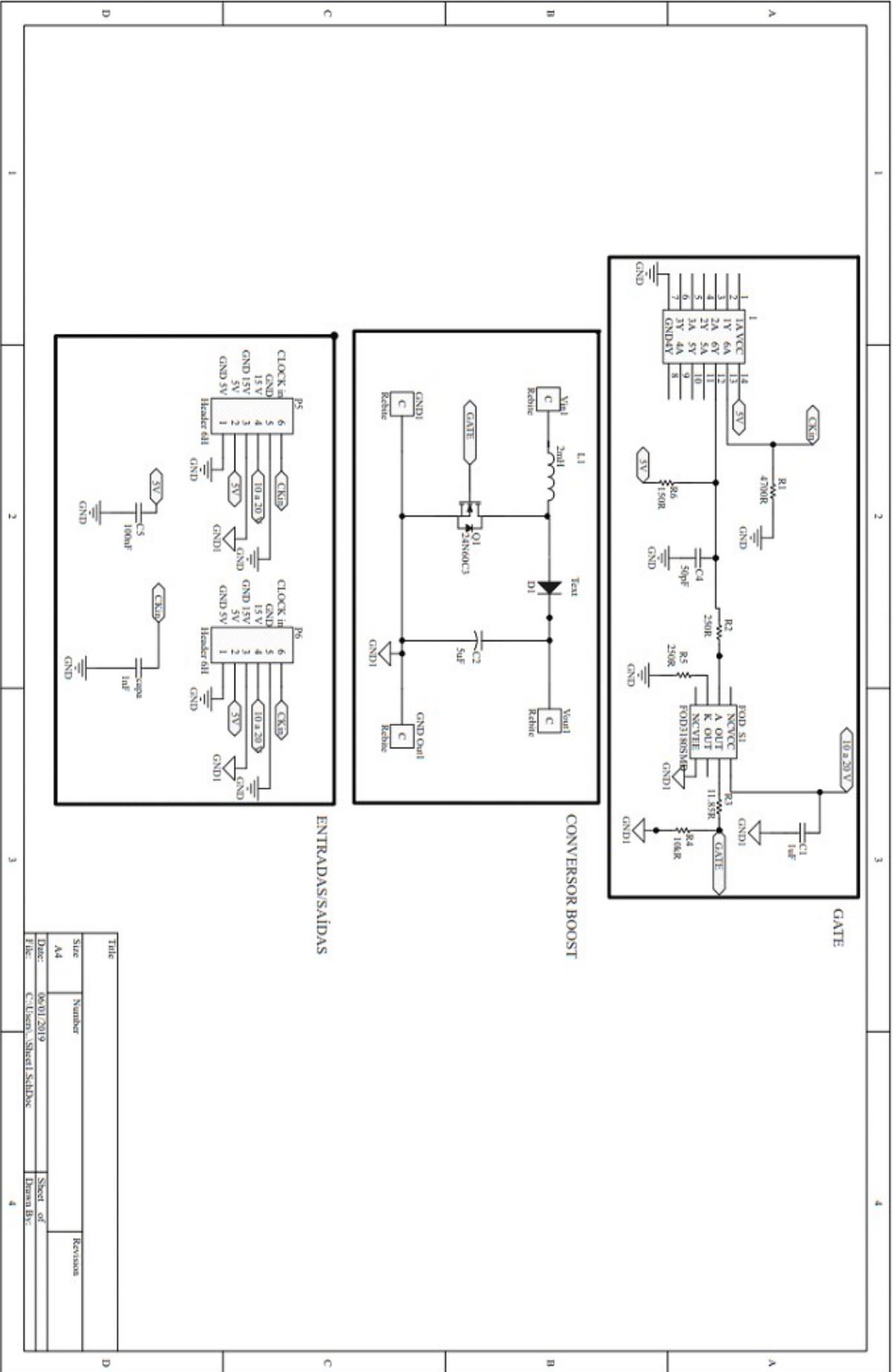
Área de Cobre:

$$A_{\text{cobre}} := \frac{I_{\text{in}}}{J_{\max}}$$

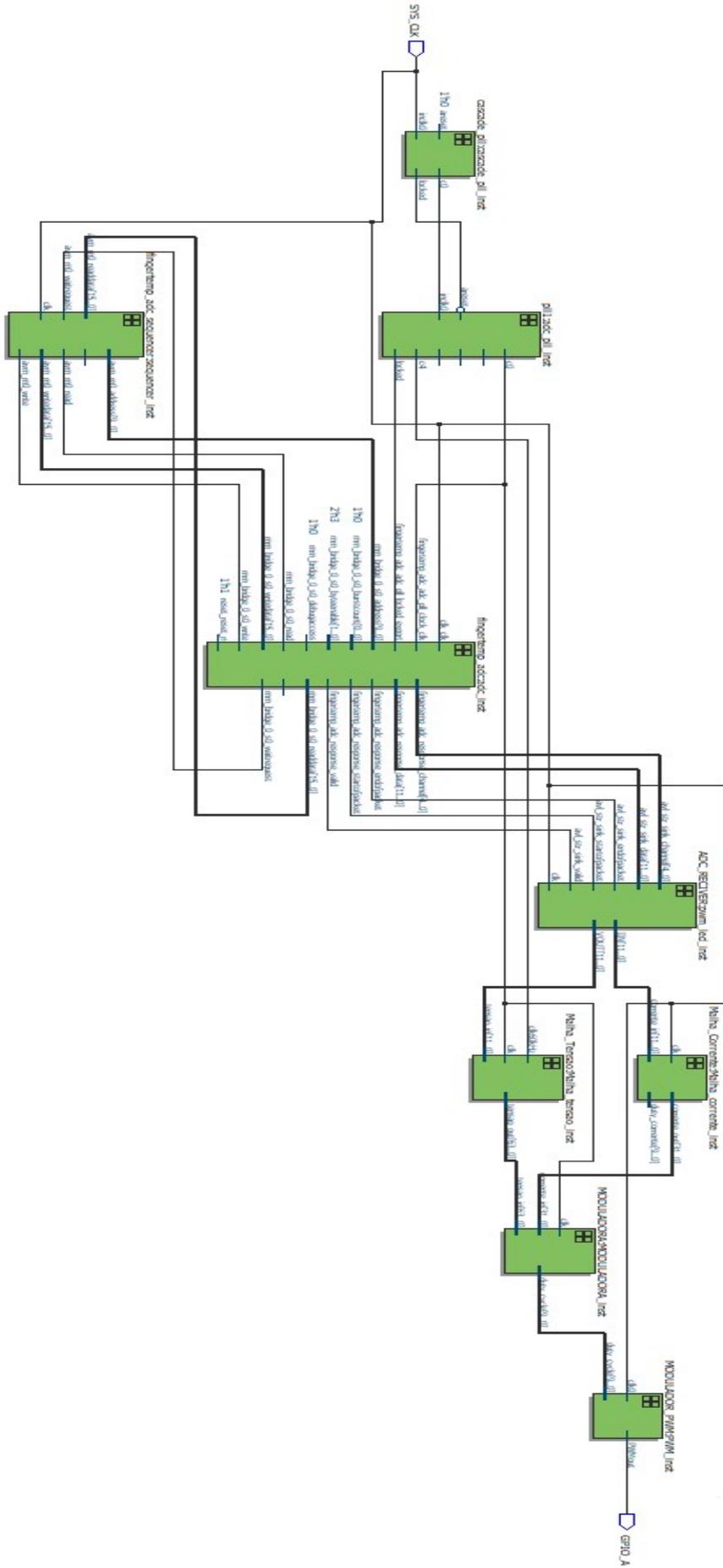
$$A_{\text{cobre}} = 8.333 \times 10^{-7} \text{ m}^2$$

Número de condutores:

**APÊNDICE C - ESQUEMÁTICO DO CONVERSOR BOOST NO ALTIUM
DESIGNER**



APÊNDICE D – VISÃO GERAL DO PROGRAMA SINTETIZADO



APÊNDICE E – CIRCUITO DIGITAL SINTETIZADO

