



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

ITALO ROGER FERREIRA MORENO PINHEIRO DA SILVA

EMULAÇÃO EM *FPGA* DE CHAVES *IGBTs* COM
CARACTERÍSTICAS REAIS

Campina Grande, Paraíba.

Fevereiro de 2012

ITALO ROGER FERREIRA MORENO PINHEIRO DA SILVA

EMULAÇÃO EM *FPGA* DE CHAVES *IGBTs* COM
CARACTERÍSTICAS REAIS

Trabalho de Conclusão de Curso
submetido ao Departamento de
Engenharia Elétrica da Universidade
Federal de Campina Grande como parte
dos requisitos necessários para obtenção
do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.

Área de Concentração: Eletrônica Industrial

Orientador:

Prof. Dr. Alexandre Cunha Oliveira

Campina Grande, Paraíba.

Fevereiro de 2012

ITALO ROGER FERREIRA MORENO PINHEIRO DA SILVA

EMULAÇÃO EM *FPGA* DE CHAVES *IGBTs* COM
CARACTERÍSTICAS REAIS

Trabalho de Conclusão de Curso
submetido ao Departamento de
Engenharia Elétrica da Universidade
Federal de Campina Grande como parte
dos requisitos necessários para obtenção
do grau de Bacharel em Ciências no
Domínio da Engenharia Elétrica.

Área de Concentração: Eletrônica Industrial

Aprovado em / /

Professor Avaliador

Universidade Federal de Campina Grande
Avaliador

Professor Alexandre Cunha Oliveira, Dr. Sc.

Universidade Federal de Campina Grande
Orientador

LISTA DE TABELAS

Tabela 1: Tempos de comutação t_{on} e t_{off} , e atrasos iniciais $t_{d(on)}$ e $t_{d(off)}$ das curvas características emuladas em <i>FPGA</i> e obtidas experimentalmente.....	19
--	----

LISTA DE ILUSTRAÇÕES

Figura 1: Esquema de um conversor R-2R.	5
Figura 2: Circuito experimental para medição das curvas características do <i>IGBT</i>	6
Figura 3: Circuito RL.	7
Figura 4: Estrutura do módulo <i>div_freq</i> implementado em <i>System Verilog</i>	9
Figura 5: Características genéricas durante a comutação do <i>IGBT</i> com tempos de comutação.	11
Figura 6: Comparação entre pulso obtido do pino do barramento de expansão da placa DE2 (cor rósea) e pulso obtido na saída analógica do conversor R-2R (cor azul).	12
Figura 7: Curvas características, V_{ce} (cor amarela) e I_c (cor verde) para o estado de condução, obtidas (a) experimentalmente e (b) pela emulação em <i>FPGA</i>	13
Figura 8: Curva característica V_{ce} (cor azul) e vetor com 50 pontos para o transitório de descida de V_{ce}	13
Figura 9: Curva característica I_c (cor verde) e vetor com 400 pontos para o transitório de subida de I_c	14
Figura 10: Curva característica V_{ce} , para o estado de condução, obtida experimentalmente (cor azul) e pela emulação em <i>FPGA</i> (cor verde).	15
Figura 11: Curva característica I_c , para o estado de condução, obtida experimentalmente (cor azul) e pela emulação em <i>FPGA</i> (cor verde).	15
Figura 12: Curvas características, V_{ce} (cor amarela) e I_c (cor verde) para o estado de bloqueio, obtidas (a) experimentalmente e (b) pela emulação em <i>FPGA</i>	16
Figura 13: Curva característica V_{ce} (cor azul) e vetor com 50 pontos para o transitório de subida de V_{ce}	16
Figura 14: Curva característica de I_c (cor verde) e vetor com 400 pontos para o transitório de descida de I_c	17
Figura 15: Curva característica V_{ce} , para o estado de bloqueio, obtida experimentalmente (cor azul) e pela emulação em <i>FPGA</i> (cor verde).	17
Figura 16: Curva característica I_c , para o estado de bloqueio, obtida experimentalmente (cor azul) e pela emulação em <i>FPGA</i> (cor verde).	18
Figura 17: Medição do atraso inicial (a) $t_{d(on)}$ e do tempo de comutação (b) t_{on}	18
Figura 18: Medição do atraso inicial (a) $t_{d(off)}$ e do tempo de comutação (b) t_{off}	19
Figura 19: Medição do tempo de subida t_r do <i>IGBT</i> obtido (a) experimentalmente e (b) pela emulação em <i>FPGA</i>	19
Figura 20: Medição do tempo de descida t_f do <i>IGBT</i> obtido (a) experimentalmente e (b) pela emulação em <i>FPGA</i>	20
Figura 21: Medição do tempo de cauda t_{tail} do <i>IGBT</i> obtido (a) experimentalmente e (b) pela emulação em <i>FPGA</i>	20

SUMÁRIO

1	Introdução	1
2	Objetivos	3
3	Materiais e métodos	4
3.1	Coleta dos pontos	5
3.2	Implementação em <i>software</i>	7
4	Análise dos resultados.....	11
5	Conclusão	22
	Bibliografia	23

1 INTRODUÇÃO

Os simuladores digitais em tempo real são aplicados em diversas áreas de atuação nas indústrias: automotiva, aeroespacial, elétrica e mecatrônica. Uma das suas aplicações mais populares e exigentes é o *Hardware-In-the-Loop (HIL)*, onde sistemas de controle reais interagem com plantas/processos simulados [1].

A simulação de plantas/processos minimiza fatores, tais como investimento temporal, financeiro e riscos inerentes a uma instalação física [2]. Dessa forma, com a implementação de sistemas *HIL* é possível compreender melhor uma dada topologia, aperfeiçoá-la por simulação, analisar condições críticas de funcionamento e estudar associação em série ou paralelo sem os riscos e os custos inerentes a uma abordagem experimental.

Entretanto, um dos grandes desafios para simulação em tempo real é a modelagem dos sistemas e suas partes constituintes, tais como os transistores bipolares de porta isolada (*IGBTs* – do inglês *Insulated Gate Bipolar Transistors*) [1]. Os simuladores em tempo real comerciais têm buscado atingir maiores níveis de detalhamento, no entanto, limitações de modelagem e tempo de processamento, ainda são fatores limitantes [1].

Os *IGBTs* são dispositivos controlados por tensão, similares a um transistor de efeito de campo de óxido metálico (*MOSFETs* – do inglês *Metal-Oxide-Semiconductor Field-Effect Transistors*). Eles apresentam baixas perdas de chaveamento e condução, além de muitas das características atraentes dos *MOSFETs* de potência, tais como facilidade de excitação da porta, corrente de pico e robustez [3]. Esses transistores ainda reúnem características de comutação dos transistores de junção bipolares de potência (*BJTs* – do inglês *Bipolar Junction Transistors*) e elevada impedância dos *MOSFETs*. A utilização desses dispositivos é bastante difundida em circuitos de controle de potência, em aplicações que exige a comutação rápida de carga de alta corrente, acionamentos de máquinas elétricas, fontes de alimentação e outras aplicações de média potência. O tempo de comutação dos *IGBTs* pode alcançar faixa de nanosegundos, o que exige alta velocidade de processamento dos simuladores em tempo real para a representação das suas curvas características reais.

A velocidade associada à implementação de modelos de sistemas físicos em *hardware*, aliada a capacidade de execução de tarefas em paralelo, são características otimizadas quando implementados em dispositivos *FPGA* (do inglês – *Field-Programmable Gate Arrays*). Portanto, a possibilidade de emulação das curvas características reais de *IGBTs*, baseada em *FPGA*, tem sido investigada, tendo como elemento motivador a maior velocidade de processamento potencialmente alcançável.

2 OBJETIVOS

Este trabalho tem como objetivo principal emular as características reais de um *IGBT*. Para isso serão implementadas tabelas em *FPGA* que contenham os pontos das curvas de tempo de resposta de abertura e fechamento da chave. Para o desenvolvimento do projeto, será realizado o ensaio de um *IGBT*, do qual serão coletados pontos que constituem as curvas características reais desse dispositivo. Na sequência, os pontos coletados serão armazenados na memória do *FPGA*. Para a emulação em *FPGA* das curvas características, será implementada em *System Verilog* uma máquina de estados, que seleciona pontos armazenados em *FPGA* de acordo com o chaveamento do *IGBT* emulado e com as condições de simulação. Ao fim deste trabalho, serão analisadas as características elétricas, como tempos de comutação e perdas de comutação, obtidas experimentalmente e pela emulação em *FPGA*.

3 MATERIAIS E MÉTODOS

Um dos objetivos deste trabalho é o ensaio de *IGBTs* para o levantamento dos pontos que constituem as curvas características reais desses dispositivos. As curvas características reais ou curvas de comutação consistem nos gráficos V_{ce} e I_c em função do tempo, para os estados de condução e bloqueio do *IGBT*, com os respectivos tempos de comutação. Tem-se V_{ce} a tensão nos terminais (coletor-emissor) da chave, I_c a corrente que circula pela chave.

O material utilizado no experimento consiste basicamente em:

- Gerador de sinais para geração dos pulsos de gatilho da chave em teste;
- Dois *IGBTs* IRG4BC30UD da *International Rectifier* cujos parâmetros nominais são: i) tensão V_{ce} de 600 V e ii) corrente I_c de 12 A @ $T_c = 100^\circ\text{C}$;
- Fonte de alimentação MCE, modelo STF: 15 V e 5 A;
- *Driver* IR2125 da *International Rectifier*;
- Fonte *Kepeco*, modelo BOP 50-20MG: 0 à 50 V e 0 à 20 A;
- Carga RL, da qual a parte resistiva é composta por 3 resistores em paralelo de 50 Ω /200 W e a parte indutiva é um indutor $L = 253$ mH;
- Placa de desenvolvimento Altera DE2 – dispositivo *FPGA Cyclone EP2C70F896*.

Os sinais de saída do *FPGA* são sinais digitais, portanto, há necessidade de implementação de um conversor digital analógico (D/A) para a visualização das curvas características do *IGBT* em teste (*D.U.T.* – do inglês *Device Under Test*). O sinal analógico é obtido a partir de uma informação digital de saída da placa DE2, representada por números binários de 12 *bits*. Esta informação foi aplicada em um conversor D/A R-2R, Figura 1, utilizando resistores de 10 k Ω . Escolheu-se este conversor devido à facilidade de construção. O nível de tensão que pode ser obtido nos pinos da placa DE2 é de 3,3 V. Logo, para uma palavra com largura de 12 *bits*, cada *bit* é conectado a um nó de um arranjo de resistores de maneira escalonada, que se inicia do *bit* menos significativo até o *bit* mais significativo, conforme a Figura 1. Dessa maneira,

para cada *bit* estará associado um ganho de tensão e a saída analógica será a soma da tensão de cada nó.

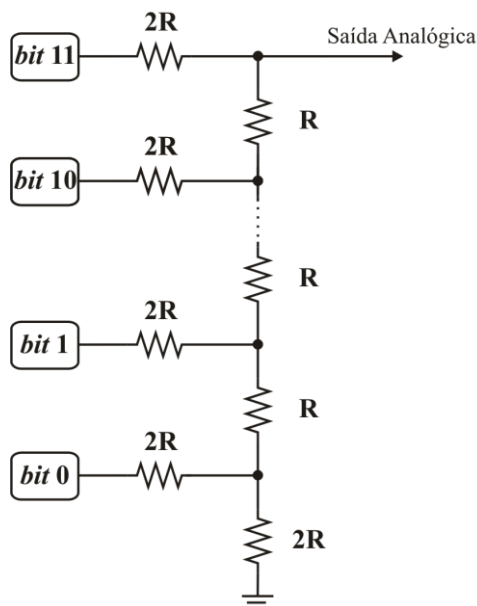


Figura 1: Esquema de um conversor R-2R.

3.1 COLETA DOS PONTOS

Os pontos que compõem as curvas características reais da chave foram obtidos a partir do circuito experimental da Figura 2. A configuração da Figura 2 encontra-se também disponível no *datasheet* do IRG4BC30UD como topologia para aferição das curvas de comutação do dispositivo. A utilização de outro *IGBT* com gatilho conectado ao emissor permite que os picos de tensão, consequentes da variação forçada da corrente durante as comutações do *D.U.T.*, sejam absorvidos pela própria carga. Ademais, isso minimiza os estresses de tensão, aos qual o *D.U.T.* estará submetido.

Os pulsos que acionam o *D.U.T.*, a partir do *driver* IR2125, são fornecidos pelo gerador de sinais a uma frequência de 1 kHz, 50% de ciclo de trabalho e amplitude de 0 e 3,3 V. O IR2125 é um *driver* para *MOSFET/IGBT* de alta tensão que trabalha com alta frequência de chaveamento. A saída do IR2125 possui um estágio *buffer*, com capacidade de geração de pulsos de alta corrente de até 3,3 A [4]. Utilizou-se a fonte MCE de 15 V para alimentação do *driver*.

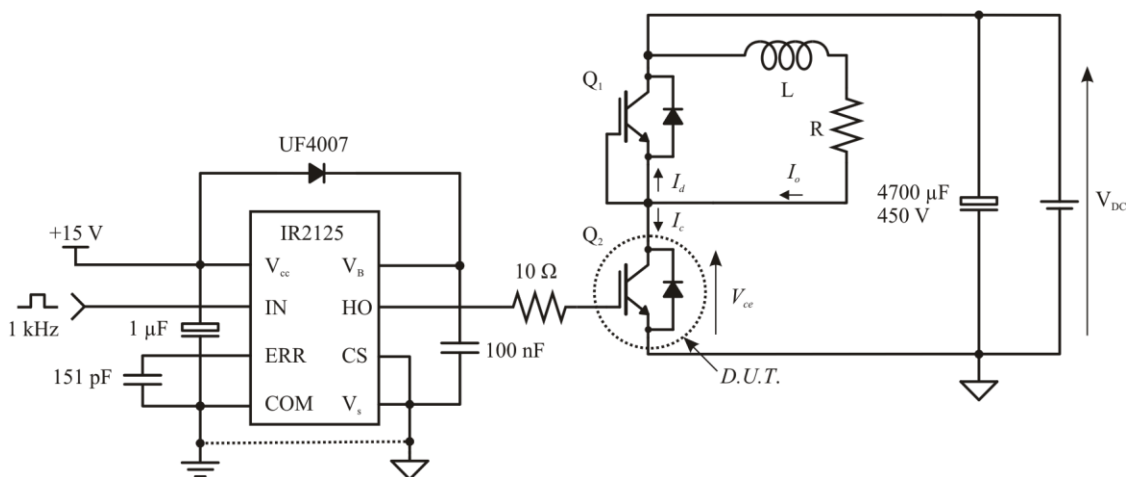


Figura 2: Circuito experimental para medição das curvas características do IGBT.

O levantamento dos pontos que constituem as curvas de comutação da chave é realizado a partir da coleta das amostras fornecidas pelo osciloscópio digital a uma frequência de amostragem 200 MHz. No MATLAB®, os pontos coletados foram manipulados para que os mesmos se adequassem a frequência de amostragem de 50 MHz do *FGPA*. Em outras palavras, para um transitório de 1 μ s, são coletados aproximadamente 200 pontos pelo osciloscópio digital, porém o *FPGA* configurado a 50 MHz só poderá armazenar 50 pontos (desses 200), pois 50 pontos lidos a cada 20 ns (50 MHz) representará 1 μ s de transitório. Ademais, também no MATLAB®, esses pontos foram convertidos para o sistema p.u. (por unidade) a uma tensão base de 40 V e corrente base de 1 A.

Na borda de subida do sinal de gatilho, foram coletados dois vetores com os pontos correspondentes ao transitório de descida do sinal V_{ce} e ao transitório de subida do sinal I_c . Enquanto, na borda de descida do sinal de gatilho, foram coletados outros dois vetores correspondentes ao transitório de subida do sinal V_{ce} e ao transitório de descida do sinal I_c .

É importante destacar que a frequência de amostragem do *FGPA* pode ser aumentada, utilizando o sistema de controle de frequência em malha fechada, denominado *PLL* (do inglês *Phase-Locked Loop*). Este sistema, disponível na placa DE2, compara a diferença de fase entre o sinal de entrada e o sinal de saída do oscilador de tensão controlada (*VCO* – do inglês *Voltage-Controlled Oscillator*). O aumento da frequência de amostragem do *FGPA* permite a leitura de mais pontos armazenados na memória *ROM* do *FPGA*.

3.2 IMPLEMENTAÇÃO EM *SOFTWARE*

A programação da placa DE2 foi realizada a partir da ferramenta de síntese lógica: Quartus II. Essa ferramenta permite a implementação dos módulos em linguagem de *hardware* (por exemplo, *System Verilog*), a depuração de erros na descrição, a simulação do projeto etc. Neste projeto, o simulador foi desenvolvido na linguagem de *hardware System Verilog*.

Para emulação das chaves em *FPGA*, implementou-se um circuito discreto equivalente ao circuito experimental da Figura 3.

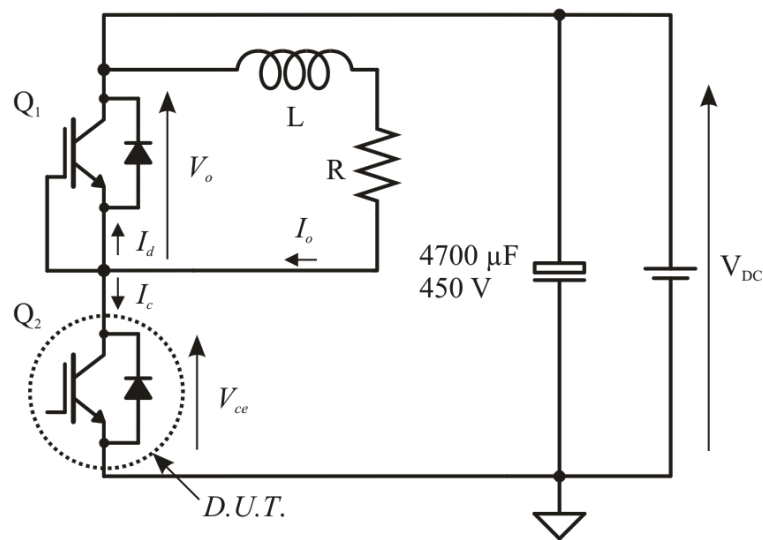


Figura 3: Circuito RL.

Da Figura 3, conclui-se que:

$$v_{CE}(t) = (1 - q)V_{DC} \quad (1)$$

$$v_o(t) = V_{DC} - v_{CE}(t) \quad (2)$$

$$v_D = -v_o(t) \quad (3)$$

Onde, q é o estado chave ($q = 1$ chave Q_2 ligada e $q = 0$ chave Q_2 desligada).

A equação contínua do circuito da Figura 3 é dada por:

$$v_o(t) = L \frac{d}{dt} i_o(t) + R i_o(t) \quad (4)$$

Segundo a definição de derivada:

$$\frac{d}{dt} i_o(t) = \lim_{h \rightarrow 0} \frac{i_o(t+h) - i_o(t)}{h} \quad (5)$$

Para um h pequeno, é possível aproximar a equação (5) para:

$$\frac{d}{dt} i_o(t) \approx \frac{i_o(t+h) - i_o(t)}{h} \quad (6)$$

Substituindo, então, a equação (6) em (4), tem-se a forma discreta final da equação (4) em função da corrente da carga $i_o(t + h)$:

$$i_o(t + h) = i(t) - \frac{h}{L} R i_o(t) + \frac{h}{L} v_o(t) \quad (7)$$

Logo,

$$i_c = q \cdot i_o(t + h) \quad (8)$$

$$i_D = (1 - q) \cdot i_o(t + h) \quad (9)$$

Dessa maneira, as equações que constituem o comportamento discreto das curvas de comutações da chave Q_2 podem ser avaliadas em (1), (3), (8) e (9).

O simulador em tempo real baseado em *FPGA* simula o comportamento do circuito discreto da Figura 3 a partir das equações (1) a (9). A construção das curvas de comutação do *D.U.T.* foi realizada a partir da leitura dos pontos coletados dos ensaios experimentais, salvos em uma tabela de registros na memória *ROM* do *FPGA*.

Para criar uma memória *ROM* no Quartus II é necessário utilizar o editor de memória, acessando-o pela barra de menus do Quartus II: *Tool ->Mega Wizard Plug-In Manager...* Nesta opção é possível especificar opções de variação personalizada de diversas *megafunctions*. A *megafunction* a ser personalizada será a memória *ROM* de uma porta, a qual se encontra no compilador de memória. O assistente permite definir valores de parâmetros e seleção de portas opcionais, tendo ao fim do mesmo, a geração de um módulo em código *Verilog*. Este módulo gerado indica as portas de entrada, *address*, através da qual serão selecionadas as posições de memória a serem lidas, o sinal de *clock* e a porta de saída, *q*, através da qual se tem acesso ao dado armazenado na posição de memória apontada. Também nesta sessão, foi selecionado o tamanho da palavra em 12 *bits* e a quantidade de palavras a serem reservadas para a memória *ROM*. Essa quantidade de palavras é definida por um número na potência de 2, ou seja, se por acaso fosse necessário utilizar 3000 palavras de 12 *bits*, seriam reservados 4096 (2 a potência de 12) palavras de 12 *bits* na memória *ROM*. A cada módulo criado a partir do procedimento de geração da memória *ROM* está associado um arquivo de inicialização de memória (*arquivo.mif*). É neste arquivo tipo *mif*, onde serão registrados os pontos que constituem as curvas de comutação da chave. Para cada curva de comutação foi criado um arquivado tipo *mif*.

Para a configuração do *PLL* no Quartus II, utiliza-se também o editor de *megafunctions*, acessando-o pela barra de menus do Quartus II: *Tool ->Mega Wizard Plug-In Manager...* A *megafunction* personalizada é a *ALTPLL*, referente aos

dispositivos de *I/O* (do inglês *In/Out*). O assistente permite definir valores da frequência de entrada e dos pré-escaladores (divisor ou multiplicador da frequência do *clock* de entrada), da fase inicial e do ciclo de trabalho (*duty cycle*) da frequência de saída. É possível também no assistente, selecionar portas opcionais, tendo ao fim do mesmo, a geração de um módulo em código *Verilog*. No módulo gerado é declarada a porta de entrada, *inclk0*, referente ao sinal de *clock* de entrada e a porta de saída, *c0*, referente ao sinal de *clock* de saída. É importante destacar que, a partir de um sinal de *clock* de entrada, podem-se obter no *PLL*, até três sinais de *clock* de saída de parâmetros distintos.

Toda estrutura de um módulo utilizando *Verilog* ou *System Verilog* é iniciada com a função *module* seguido do “nome do módulo” e finalizada com um *endmodule*. De um modo geral, essa estrutura contém: as declarações de portas de entrada/saída (E/S); os registradores, fios e parâmetros; a funcionalidade descrita pela linguagem de *hardware* e as atribuições. Para exemplificar, será descrito o módulo *div_freq* na Figura 4, o qual se trata de um dos módulos implementados neste projeto.

```

Início do módulo:
    module div_freq(clk_in, clk_out);
Declaração de portas de E/S, registradores e parâmetros:
    input clk_in;
    output clk_out;
    reg [24:0] div_50;
    reg [24:0] acc;
    parameter f = 1000; //Frequência de operação desejada
Funcionalidade do módulo:
    always@(posedge clk_in)
    begin
        div_50 = div_50 + 1;
        acc = 25000000/f;
        if(div_50 >= acc)
        begin
            div_50 = 0;
            clk_out = clk_out^1; //Operação de Toogle
        end
    end
Atribuição:
    //Espaço para atribuições
Fim do módulo:
    endmodule
  
```

Figura 4: Estrutura do módulo *div_freq* implementado em *System Verilog*.

Vale destacar que a ferramenta Quartus II executa as operações aritméticas no sistema de representação de números inteiros em complemento de 2. Para o cálculo das operações aritméticas, utilizou-se a aritmética de ponto fixo. A aritmética de ponto fixo

é usada geralmente quando a velocidade de *hardware*, custo e complexidade são fatores essenciais [5]. No entanto, trata-se de uma aritmética mais complexa por necessitar uma análise mais cuidadosa, tendo em vista, que os fatores de escala devem ser considerados. Essa restrição pode ser contornada com a implementação de blocos matemáticos para realização de operações aritméticas em ponto flutuante.

O *FPGA* executa as operações aritméticas em complemento de 2 e armazena somente resultados inteiros nos registradores. Para aumentar a precisão dos resultados e efetuar cálculos com valores menores que 1 (μm), é necessária uma solução alternativa. A alternativa foi deslocar 10 *bits* à esquerda em todas variáveis (de 12 *bits*). Sendo assim, as variáveis para cálculos das equações discretas são representadas por 22 *bits*, e quando essas variáveis são enviadas para as saídas do *FPGA* são deslocadas 10 *bits* à direita.

Realizado o registro dos pontos que representam as curvas de operação do *IGBT*, implementou-se uma máquina de estados que seleciona os pontos na memória *ROM* do *FPGA* de acordo com as condições de simulação. A máquina de estados inicia a partir da detecção de borda, subida ou descida, do sinal de gatilho da chave. Ao ser detectada a borda de subida/descida do sinal de gatilho, a máquina de estados habilita a contagem de um contador que aponta para os endereços das respectivas curvas, V_{ce} e I_c , retornando da memória *ROM* o valor normalizado dos pontos coletados. Assim, os pontos dos sinais de V_{ce} e I_c que seriam calculados apenas pelas equações (1) e (8) são ponderados pelos pontos coletados através de uma operação de multiplicação. Ao fim da contagem, que coincide com o número de pontos coletados (regime transitório), a máquina de estados retorna ao cálculo dos pontos pelas equações (1) a (9) (regime permanente).

4 ANÁLISE DOS RESULTADOS

Nesta secção serão analisados os resultados obtidos de experimentos e de emulação em *FPGA* para o circuito da Figura 2. Para alimentar os *IGBTs* e a carga, utilizou-se a fonte *Kepeco* regulada em 20 V. Na emulação, os resultados foram obtidos em p.u., com tensão base igual a 40 V e corrente base igual a 1 A. Utilizou-se uma frequência de amostragem de 50 MHz do *FPGA* para a leitura dos pontos que constituem as curvas características, V_{ce} e I_c , do *D.U.T.*

A Figura 5 ilustra as curvas características genéricas V_{ce} e I_c do *IGBT* para os estados de condução e bloqueio, com os respectivos tempos de comutação. É possível observar durante a comutação para o estado de condução do *IGBT*, a presença de um atraso inicial, $t_{d(on)}$, entre o instante em que a tensão de gatilho V_{ge} está em nível alto e o instante em que a corrente I_c começa a aumentar. O intervalo de tempo para que I_c aumente de 10% a 90% do valor final é definido como t_r , enquanto que o intervalo de tempo entre 10% de I_c e 10% de V_{ce} é denotado de t_{on} .

Analogamente, ainda na Figura 5, para o bloqueio do *IGBT*, tem-se o tempo de atraso entre desligamento de V_{ge} e o instante em que V_{ce} aumenta, definido como $t_{d(off)}$. O tempo de descida t_f de I_c define o intervalo de 90% a 10% do valor final, enquanto t_{off} é o intervalo de tempo entre 10% de V_{ce} e 10% de I_c . O intervalo de tempo que define a corrente de cauda encontra-se entre 10% de I_c e 1% de I_c .

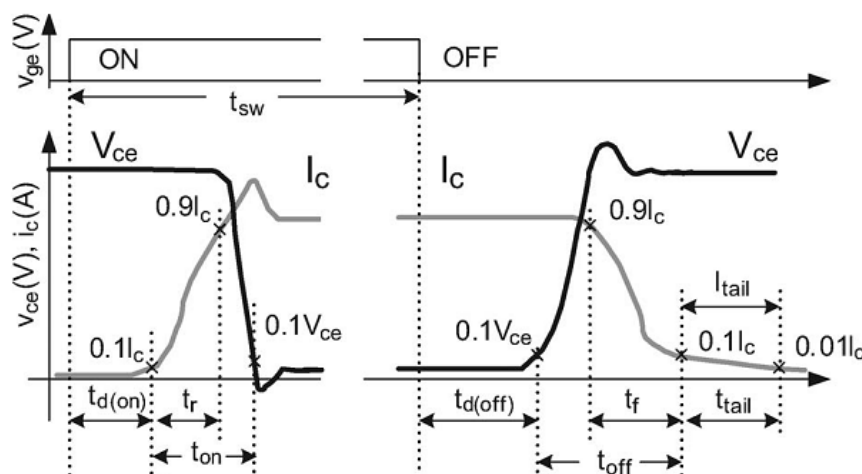


Figura 5: Características genéricas durante a comutação do *IGBT* com tempos de comutação.
FONTE: adaptado de [1].

Observou-se durante a emulação em *FPGA* das curvas características que o conversor R-2R, apesar de resistivo, proporcionava um atraso no sinal de saída analógico de aproximadamente 1 μ s, devido a presença de capacitâncias parasitas. Este tempo de resposta frente a um sinal obtido diretamente do pino do barramento de expansão da placa DE2 pode ser visualizado na Figura 6.

O desempenho do conversor R-2R pode ser melhorado com o uso de *drivers* na saída dos pinos da placa DE2, os quais ampliarão o nível de potência do sinal gerado pelo *FPGA*. Deste modo, é possível utilizar resistores de menor valor, reduzindo este atraso.

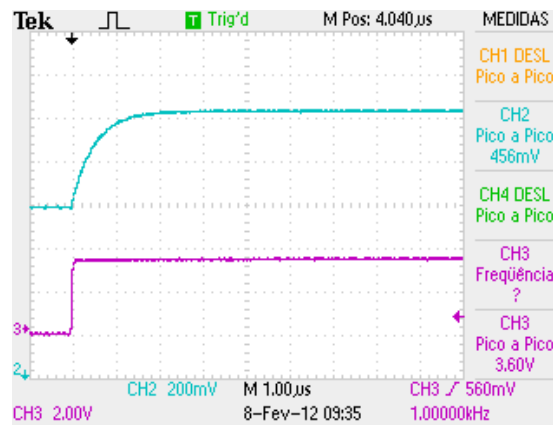


Figura 6: Comparação entre pulso obtido do pino do barramento de expansão da placa DE2 (cor rósea) e pulso obtido na saída analógica do conversor R-2R (cor azul).

É importante destacar que como o nível de tensão que pode ser obtido nos pinos da placa DE2 é de 3,3 V, assim, os resultados obtidos da emulação em *FPGA* apresentam um *offset* de aproximadamente 1,65 V. Isso é necessário para que seja possível representar valores negativos na faixa de tensão de 0 a 3,3 V. Portanto, o “zero real” é representado por 1,65 V, os valores negativos pela faixa de tensão de 0 a 1,64 V e os valores positivos pela faixa de tensão de 1,66 V a 3,3 V.

As formas de onda do transitório da comutação forçada para o estado de condução da chave são ilustradas na Figura 7. Nesta figura, observa-se que o transitório de descida de V_{ce} é de aproximadamente 1 μ s, enquanto o transitório de subida de I_c é de aproximadamente de 8 μ s. Desta curva, Figura 7 (a), foram coletados exatamente 2484 pontos. Utilizando o MATLAB®, coletaram-se dois vetores que representassem o mais real possível os transitórios das curvas de V_{ce} e de I_c : um vetor de 50 pontos para o transitório de descida de V_{ce} (Figura 8) e um vetor de 400 pontos para o transitório de

subida de I_c (Figura 9). Percebe-se que para um tempo de amostragem do *FPGA* de 20 ns (50 MHz), tem-se 1 μ s (50*20 ns) para o transitório de tensão e 8 μ s (400*20 ns) para o transitório de corrente. O resultado da emulação das curvas características a partir destes vetores é ilustrado na Figura 7 (b).

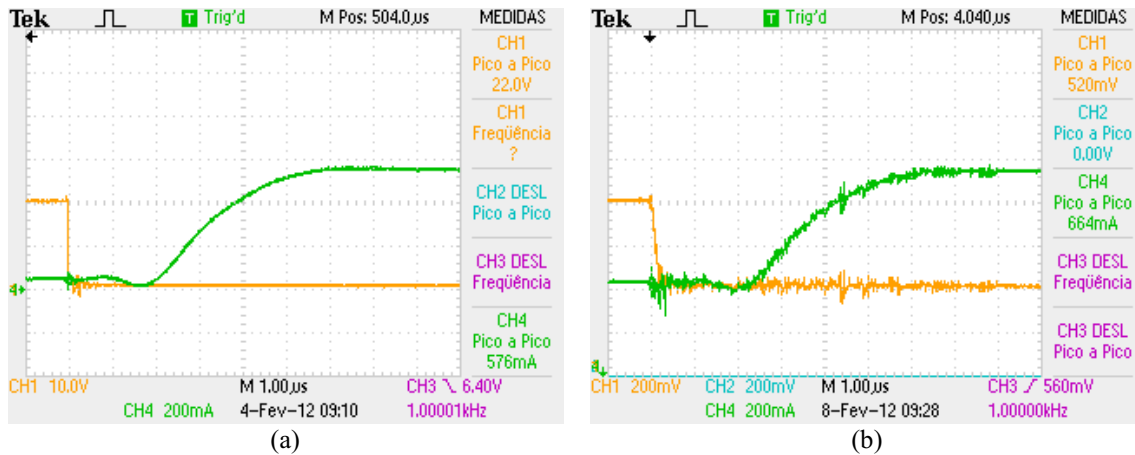


Figura 7: Curvas características, V_{ce} (cor amarela) e I_c (cor verde) para o estado de condução, obtidas (a) experimentalmente e (b) pela emulação em *FPGA*.

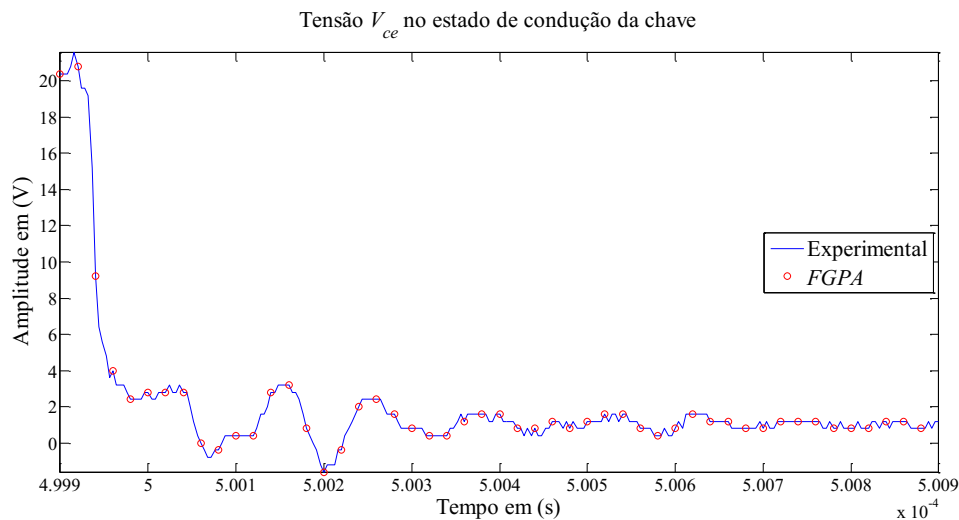


Figura 8: Curva característica V_{ce} (cor azul) e vetor com 50 pontos para o transitório de descida de V_{ce} .

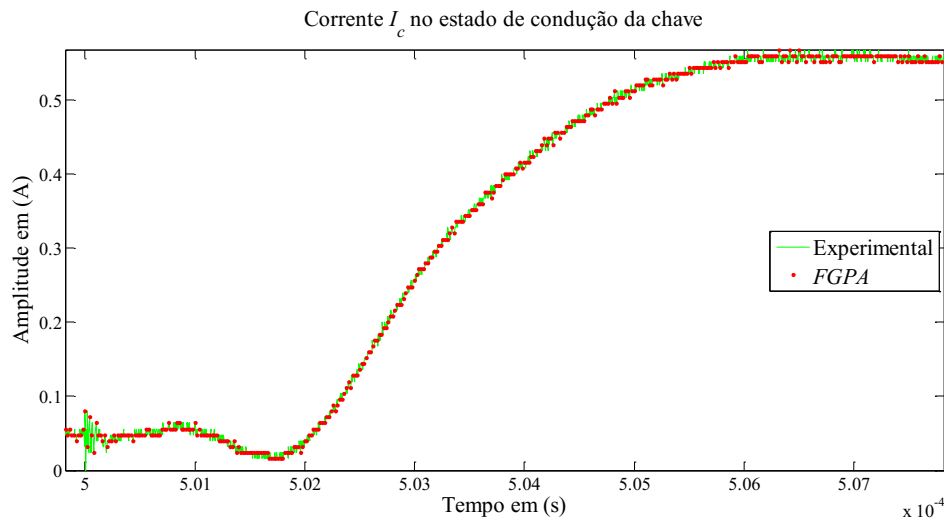


Figura 9: Curva característica I_c (cor verde) e vetor com 400 pontos para o transitório de subida de I_c .

Com a utilização do *PLL*, disponível na placa DE2, o número de pontos nos transitórios das curvas de V_{ce} pode ser aumentado para 200 pontos. Para isso, configura-se o *PLL* para gerar um sinal de *clock* em 200 MHz (5 ns), obtendo um transitório de tensão de 1 μ s (200*5 ns).

É possível observar na Figura 7 (b) que a utilização do conversor R-2R filtrou as componentes de frequência mais altas, impossibilitando a análise do transitório de V_{ce} . Como o transitório de I_c é mais lento, esta forma de onda apresenta-se de forma semelhante ao obtido experimentalmente.

Observa-se nos resultados das curvas emuladas a presença de ruídos, a qual é justificada pela mudança de estados dos *bits* (efeito de chaveamento do circuito digital de 0 para 1 ou 1 para 0), que resulta em oscilações na corrente do sinal digital. Essas oscilações de corrente refletem no “terra” dos CIs (Circuitos Integrados). Como neste experimento, houve a conexão entre os “terras” do circuito digital e do circuito analógico, o sinal do analógico pode apresentar essas oscilações da corrente na forma de ruídos. Esses ruídos podem ser minimizados com a presença de apenas um ponto de conexão entre os “terras” digital e analógico, além da utilização de um indutor de *bounce*, que filtra as componentes CA do sinal do “terra”. Outra ação para minimização dos efeitos de comutação dos dispositivos digitais no comportamento do “terra” é a utilização de capacitores de desacoplamento junto aos pinos de alimentação dos CIs digitais, absorvendo as variações rápidas de tensão nos pinos destes.

Multiplicando as amplitudes dos sinais de tensão e corrente da Figura 7 (b) pelos valores de base, 40 V e 1 A, respectivamente, verifica-se que as mesmas são compatíveis com as amplitudes dos sinais obtidos na Figura 7 (a). No MATLAB®,

plotou-se os gráficos referentes às curvas características emuladas a partir dos pontos da Figura 7 multiplicadas pelos valores de base, conforme a Figura 10, para as curvas de V_{ce} , e Figura 11, para as curvas de I_c .

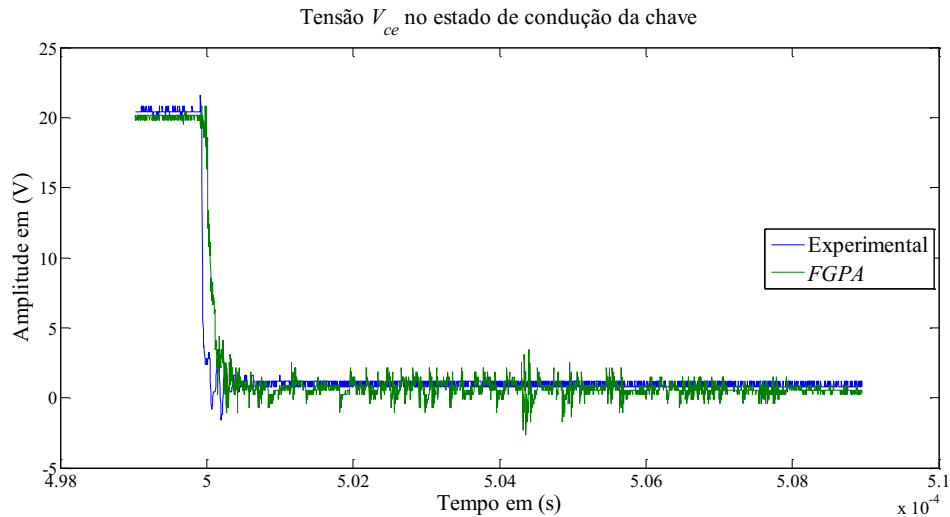


Figura 10: Curva característica V_{ce} , para o estado de condução, obtida experimentalmente (cor azul) e pela emulação em *FPGA* (cor verde).

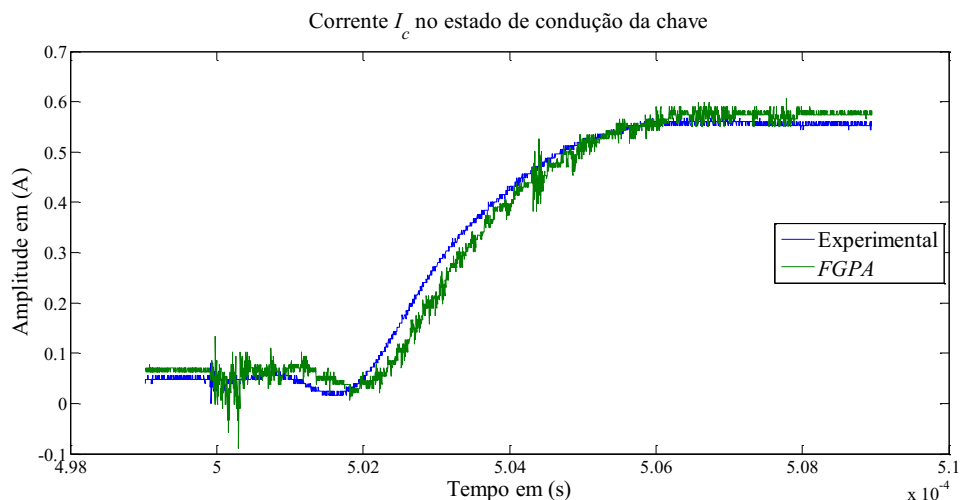


Figura 11: Curva característica I_c , para o estado de condução, obtida experimentalmente (cor azul) e pela emulação em *FPGA* (cor verde).

As formas de onda do transitório da comutação forçada para o estado de bloqueio da chave são ilustradas na Figura 12. De forma similar a análise do transitório para o estado de condução, observa-se na Figura 12 (a) que o transitório de subida de V_{ce} é de aproximadamente $1\mu s$ e o transitório de descida de I_c é de aproximadamente $8\mu s$. Neste caso, também, foram coletados 2484 pontos da curva. Desses 2484 pontos, foi obtido um vetor de 50 pontos para o transitório de subida de V_{ce} (Figura 13) e um vetor

de 400 pontos para o transitório de descida de I_c (Figura 14). O resultado da emulação das curvas características a partir destes vetores é ilustrado na Figura 12 (b).

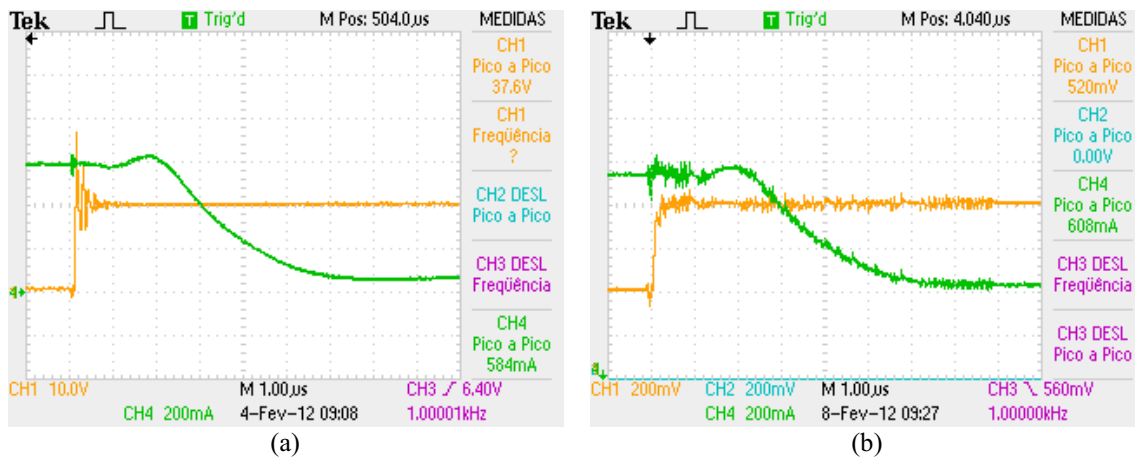


Figura 12: Curvas características, V_{ce} (cor amarela) e I_c (cor verde) para o estado de bloqueio, obtidas (a) experimentalmente e (b) pela emulação em *FPGA*.

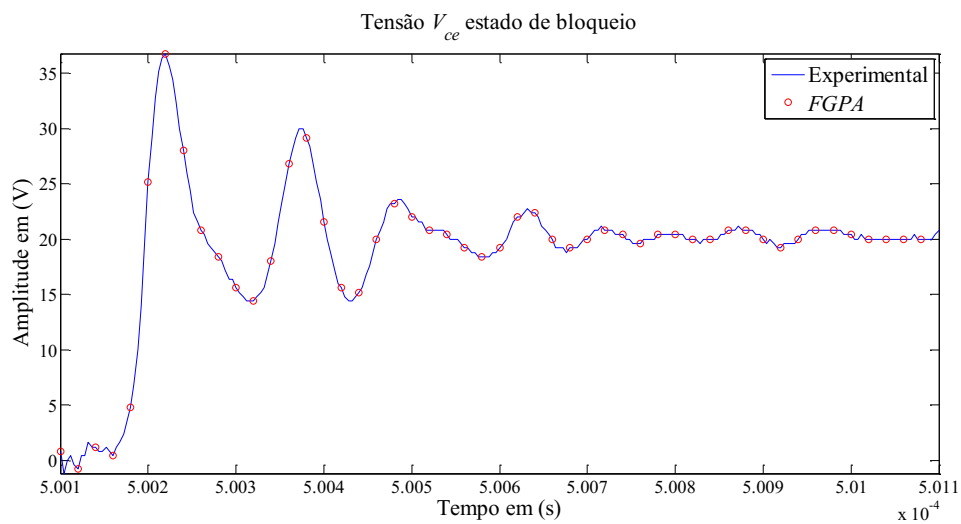


Figura 13: Curva característica V_{ce} (cor azul) e vetor com 50 pontos para o transitório de subida de V_{ce} .

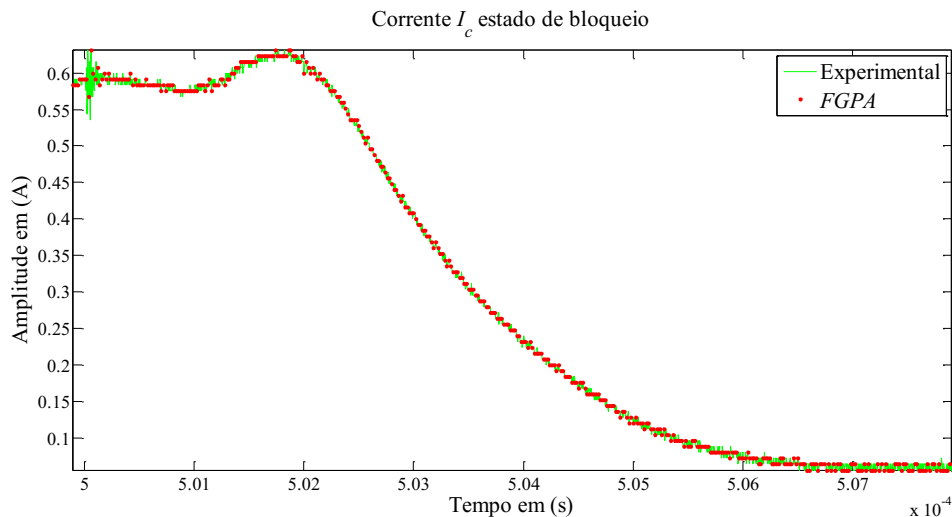


Figura 14: Curva característica de I_c (cor verde) e vetor com 400 pontos para o transitório de descida de I_c .

Observa-se também na Figura 12 (b), o amortecimento causado pelo conversor R-2R no transitório de subida de V_{ce} . A curva de I_c , Figura 12 (b), assemelha-se com o resultado obtido experimentalmente, Figura 12 (a). Multiplicando as amplitudes dos sinais de tensão e corrente da Figura 12(b) pelos valores de base, 40 V e 1 A, respectivamente, verifica-se que as mesmas são compatíveis com as amplitudes dos sinais obtidos na Figura 12 (a). No MATLAB®, plotou-se os gráficos referentes às curvas características emuladas a partir dos pontos da Figura 12 multiplicadas pelos valores de base, conforme a Figura 15, para as curvas de V_{ce} , e Figura 16, para as curvas de I_c .

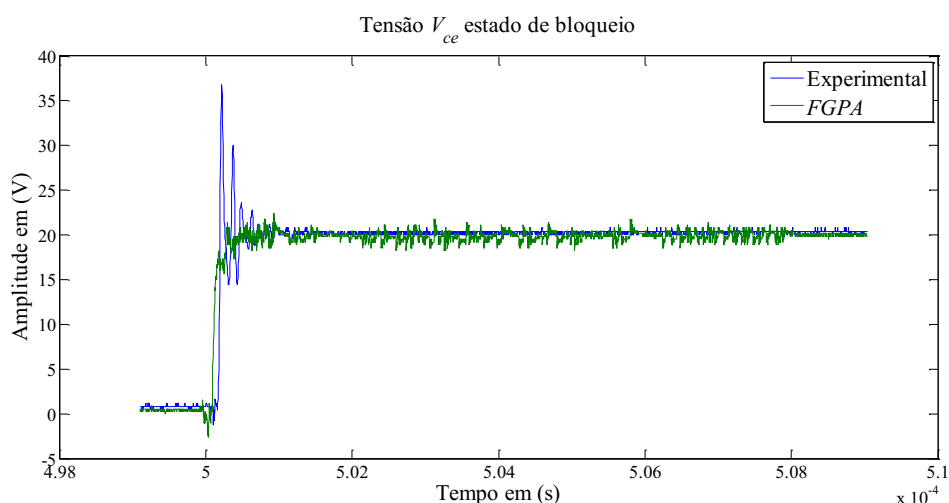


Figura 15: Curva característica V_{ce} , para o estado de bloqueio, obtida experimentalmente (cor azul) e pela emulação em *FPGA* (cor verde).

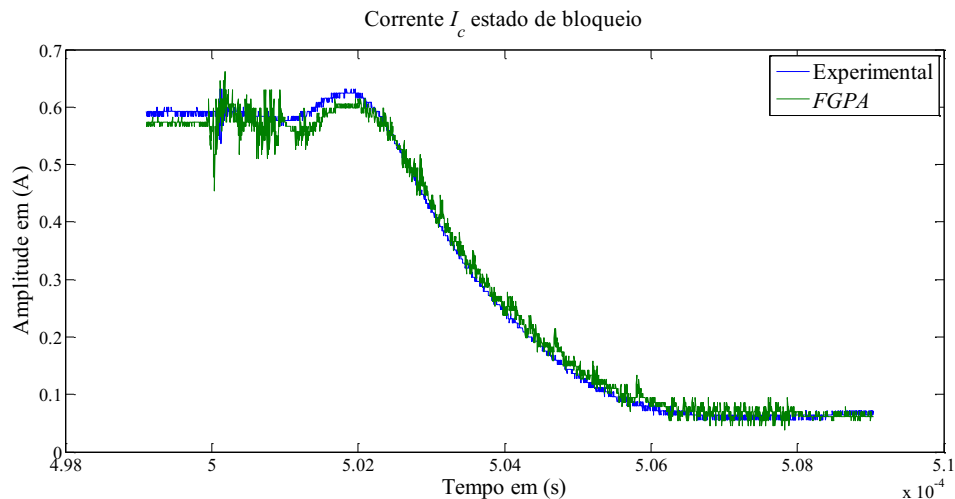


Figura 16: Curva característica I_c , para o estado de bloqueio, obtida experimentalmente (cor azul) e pela emulação em *FPGA* (cor verde).

Na Figura 17 é ilustrada a medição do atraso inicial $t_{d(on)}$ e do tempo de comutação t_{on} do transitório para o estado em condução, e na Figura 18 é ilustrada a medição do atraso inicial $t_{d(off)}$ e do tempo de comutação t_{off} do transitório para o estado em bloqueio. Pelo osciloscópio digital, não foi possível medir com exatidão estas características elétricas, t_{on} , $t_{d(on)}$, t_{off} e $t_{d(off)}$, emuladas em *FPGA*, devido ao amortecimento causado pelo conversor R-2R, entretanto, os valores desses tempos podem ser obtidos a partir dos vetores que contém os pontos que são armazenados no *FPGA*, conforme a Tabela 1.

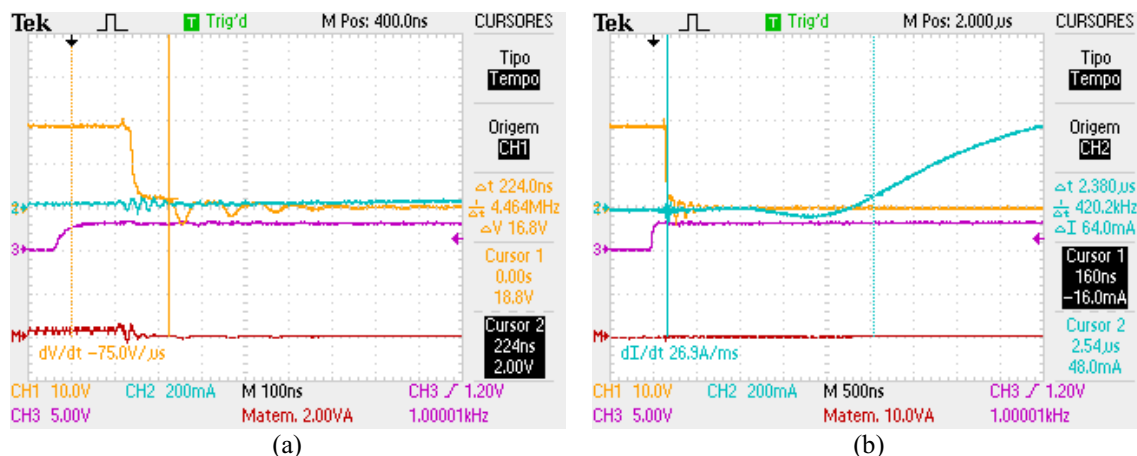


Figura 17: Medição do atraso inicial (a) $t_{d(on)}$ e do tempo de comutação (b) t_{on} .

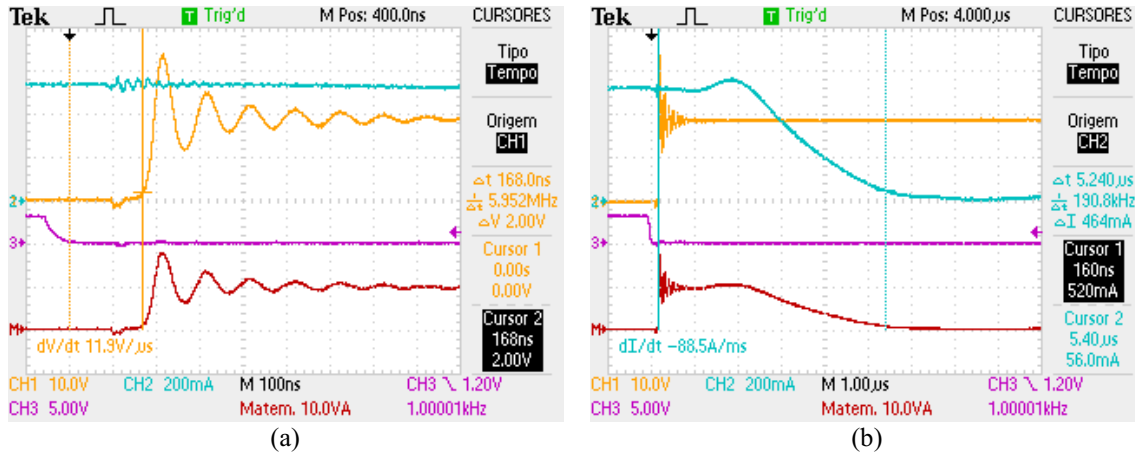


Figura 18: Medição do atraso inicial (a) $t_{d(off)}$ e do tempo de comutação (b) t_{off} .

Tabela 1: Tempos de comutação t_{on} e t_{off} , e atrasos iniciais $t_{d(on)}$ e $t_{d(off)}$ das curvas características emuladas em *FPGA* e obtidas experimentalmente.

	<i>FPGA</i>	Experimental
$t_{d(on)}$	210,0 ns	224,0 ns
t_{on}	2,34 µs	2,38 µs
$t_{d(off)}$	240,0 ns	168,0 ns
t_{off}	4,85 µs	5,24 µs

Na Figura 19 é ilustrada a medição do tempo de subida t_r do *D.U.T.*. Os cursores de tempo do osciloscópio delimitam o intervalo que compreende 10% e 90% do valor da corrente I_c . Na tabela estão registrados o tempo de subida t_r obtido experimentalmente e na emulação em *FPGA*. Na Figura 19 (b), emulação em *FPGA*, o tempo t_r é igual a 2,720 µs, valor bastante próximo ao tempo t_r obtido experimentalmente, 2,760 µs, conforme observado na Figura 19 (a).

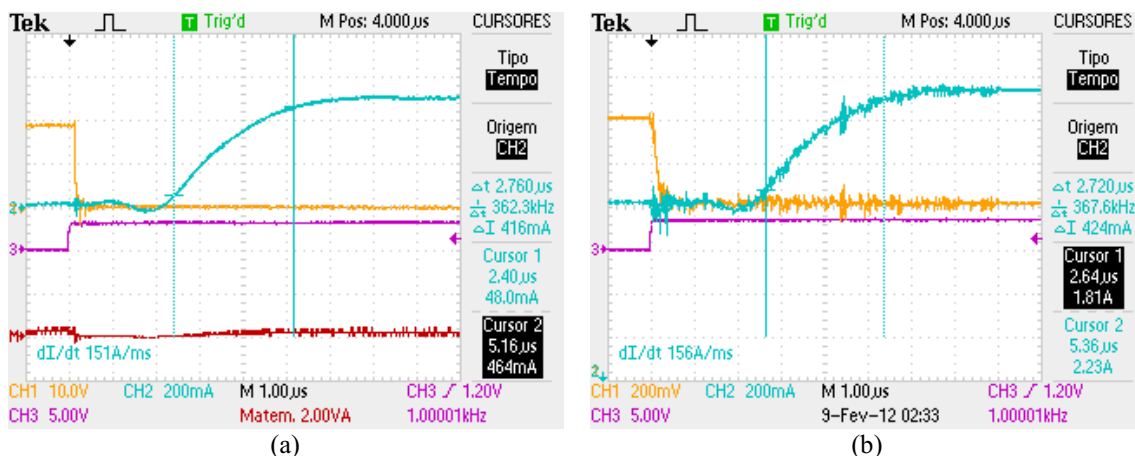


Figura 19: Medição do tempo de subida t_r do *IBGT* obtido (a) experimentalmente e (b) pela emulação em *FPGA*.

Analogamente, na Figura 20 é ilustrada a medição do tempo de descida t_f do *D.U.T.*. Os cursores de tempo do osciloscópio delimitam o intervalo que compreende

90% e 10% do valor da corrente I_c . Na Figura 20 (a), observa-se que o tempo t_f obtido experimentalmente é igual a 2,840 μs , e na Figura 20 (b), a partir da emulação em *FPGA*, o tempo t_f é igual a 2,880 μs .

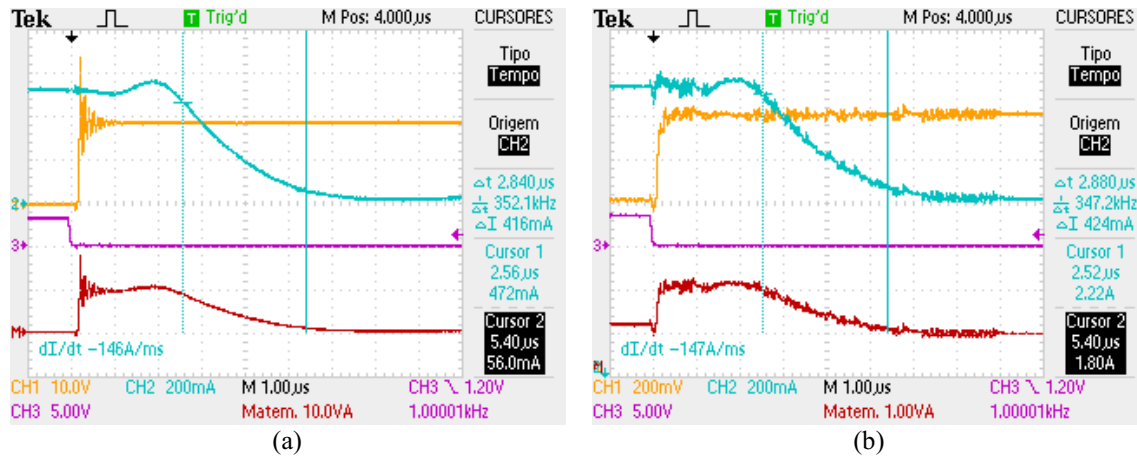


Figura 20: Medição do tempo de descida t_f do IGBT obtido (a) experimentalmente e (b) pela emulação em *FPGA*.

Na Figura 21 é ilustrada a medição do tempo de cauda t_{tail} do *D.U.T.*. Os cursores de tempo do osciloscópio delimitam o intervalo que compreende aproximadamente 10% e 1% do valor da corrente I_c . Na Figura 21 (a), observa-se que o tempo t_{tail} obtido experimentalmente é igual a 588,0ns, e na Figura 21 (b), a partir da emulação em *FPGA*, o tempo t_{tail} é igual a 596,0 ns.

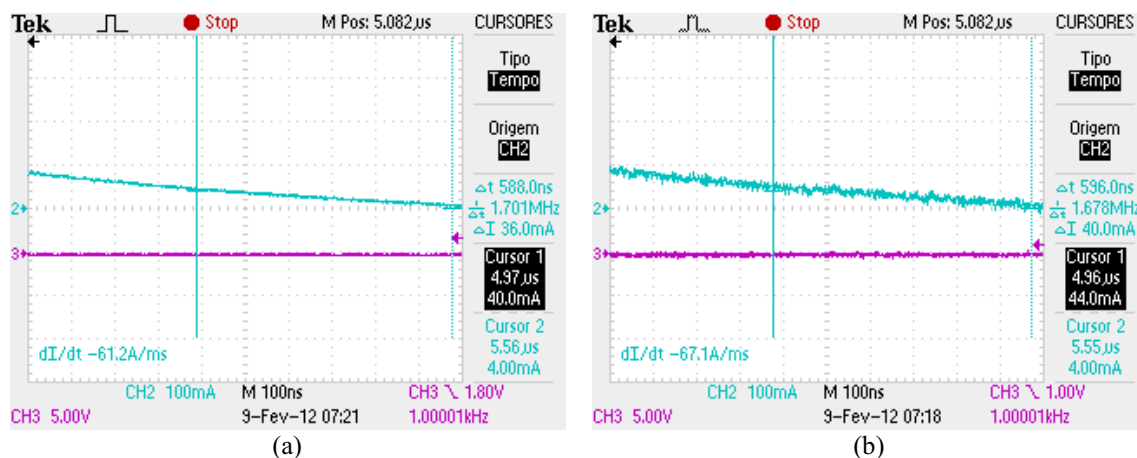


Figura 21: Medição do tempo de cauda t_{tail} do IGBT obtido (a) experimentalmente e (b) pela emulação em *FPGA*.

As perdas no chaveamento são decorrentes da dissipação de potência durante o transitório do chaveamento. Para uma alta frequência de *PWM* (do inglês *Pulse-Width Modulation*), essas perdas tornam-se significativas e devem ser consideradas. As perdas

por chaveamento são proporcionais ao produto da tensão V_{ce} pela corrente I_c no instante da comutação. Os dados de tensão V_{ce} e corrente I_c obtidos no experimento e na emulação em *FPGA* foram multiplicados ponto a ponto, resultado nas formas de onda (sinais de cor vermelha da Figura 20) das perdas durante a comutação forçada do *IGBT* para estado de bloqueio.

5 CONCLUSÃO

Neste trabalho foi possível emular as características reais, tais como o tempo de resposta de comutação, tempo de corrente de cauda e perdas durante o transitório para estado de bloqueio do *IGBT* modelo IRG4BC30UD.

A partir dos ensaios, foi realizado um levantamento dos pontos que constituem as curvas características reais do *IGBT*. Para emulação deste dispositivo, implementou-se em *System Verilog* uma máquina de estados que seleciona a tabela que contém estes pontos coletados no experimento e armazenados no *FPGA*.

Devido às limitações da aparelhagem do laboratório, o experimento foi realizado para uma tensão de 20 V, corrente de 520 mA e uma carga RL. Desta maneira, não foi possível comparar os dados referentes às características elétricas medidas neste experimento com os dados fornecidos no *datasheet* do fabricante do *IGBT* em teste. Os dados registrados no *datasheet* do fabricante são resultados de ensaios realizados para valores de tensão de 480 V (80% da tensão nominal) e corrente de 12 A (corrente nominal).

Apesar deste trabalho não ser avaliado nas mesmas condições de ensaio que as condições fornecidas pelo fabricante, pode-se validar a emulação das curvas características em comparação com os resultados obtidos experimentalmente. Os valores de tempos de comutação da emulação em *FPGA* são compatíveis com os resultados reais.

Os ruídos presentes nas curvas de emulação em *FPGA* podem ser mitigados com a presença de apenas um ponto de conexão entre os “terras” digital e analógico, além da utilização de um indutor de *bounce*, que filtra as componentes CA do sinal do “terra”. A utilização de capacitores de desacoplamento junto aos pinos de alimentação dos CIs é outra ação para minimização dos ruídos.

Sugere-se como continuação do trabalho a substituição do conversor R-2R por outro de tempo de resposta ultrarrápido e a realização de ensaios com valores de tensão e corrente próximos aos valores testados pelo fabricante do *IGBT*.

BIBLIOGRAFIA

1. MYAING, A.; DINAHAHI, V. FPGA-Based Real-Time Emulation of Power Electronic Systems With Detailed Representation of Device Characteristics. **Industrial Electronics, IEEE Transactions on**, 58, n. 1, Janeiro 2011. 358-368.
2. CHIBANTE, R. F. M. **Desenvolvimento de um modelo para IGBTs otimizado por um método de base experimental**: Tese(Doutorado em Ciências de Engenharia). Porto: Faculdade de Engenharia da Universidade do Porto, 2005.
3. RASHID, M. H. Transistores de Potência. In: RASHID, M. H. **Eletrônica de Potência: Circuitos, Dispositivos e Aplicações**. São Paulo: Makron Books, 1999. Cap. 8.
4. INTERNATIONAL RECTIFIER. **Datasheet: Current Limiting Single Channel Driver**. Eletronic Publication. 2003.
5. JONES, D. Fixed-Point Number Representation, 28 Dezembro 2004. Disponível em: <<http://cnx.org/content/m11930/1.2/>>. Acesso em: 17 Março 2011.
6. ABNT. **NBR 10520 - Informação e documentação - Citações em documentos - Apresentação**. Associação Brasileira de Normas Técnicas. [S.l.]: ABNT. 2002. p. 7.
7. ABNT. **NBR 14724 - Informação e documentação — Trabalhos acadêmicos — Apresentação**. Associação Brasileira de Normas Técnicas. [S.l.]: ABNT. 30 dez. 2005. p. 9.
8. ABNT. **NBR 6023 - Informação e documentação - Referências - Elaboração**. Associação Brasileira de Normas Técnicas. [S.l.]: ABNT. 2002. p. 24.
9. ABNT. **NBR 6034 - Informação e documentação - Índice - Apresentação**. Associação Brasileira de Normas Técnicas. [S.l.]: ABNT. 2005. p. 4.
10. ABNT. **NBR 6028 - Informação e documentação - Resumo - Apresentação**. [S.l.]: Associação Brasileira de Normas Técnicas. 2003. p. 2.
11. WU, B. High-Power Semiconductor Devices. In: WU, B. **High-Power Converters and AC Drives**. Hoboken, New Jersey: John Wiley & Sons, 2006. Cap. 2.
12. INTERNATIONAL RECTIFIER. **Datasheet: Insulated Gate Bipolar Transistor with Ultrafast Soft Recovery Diode**. Eletronic Publication. 2000.