

IMPLEMENTAÇÃO DE CONVERSORES ELETRÔNICOS UTILIZANDO DISPOSITIVOS FPGA/SOC ZYNQ7020

Tony Albert Lima¹ (IC), Wilson Cesar Sant'Ana (PQ)¹

¹Universidade Federal de Itajubá.

Palavras-chave: Eletrônica de Potência. FPGA. PWM. System On Chip. Zynq.

Introdução

Tanto no ambiente industrial quanto no acadêmico, espera-se que produtos sejam colocados no mercado ou que resultados experimentais sejam obtidos em tempos cada vez menores (Menghal e Laxmi, 2010) - logo, ferramentas que acelerem o processo de desenvolvimento são sempre bem vindas. A tecnologia FPGA (Field Programmable Gate Array) permite a síntese reconfigurável de circuitos eletrônicos em um microchip (Nane et al., 2016). Isso torna essa tecnologia extremamente atrativa para estudantes / pesquisadores / desenvolvedores, pois permite que uma placa outrora imensa caiba dentro de um pequeno chip. Mais importante do que isso, permite que o circuito seja facilmente modificado conforme as necessidades - sem necessitar ajustes físicos - apenas com alterações em seu código HDL (Hardware Description Language) (Gama et al., 2021).

Uma aplicação que se beneficia grandemente da tecnologia FPGA são os conversores eletrônicos de potência. Estes equipamentos possuem dispositivos semicondutores (IGBTs ou MOSFETs) chaveando em alta frequência, visando o controle de certas grandezas elétricas (tensões, correntes, etc.). O chaveamento desses dispositivos normalmente é realizado utilizando modulação PWM. Visando a operação em médias e altas tensões esses conversores podem ser arranjos em topologias multiníveis. Os arranjos multiníveis possuem como vantagens a divisão da média/alta tensão entre os semicondutores arranjos em série e uma distorção harmônica muito mais baixa. Como desvantagem, eles necessitam que um número muito grande de chaves semicondutoras seja controlado ao mesmo tempo, o que excede as capacidades (em termos de pinos de I/O) de microcontroladores ou DSPs. Já os FPGAs apresentam um número muito maior de pinos de I/O (embora, por si só, não apresentem as mesmas capacidades de processamento de um DSP). Sant'Ana et al. (2018) apresenta uma implementação em FPGA de um modulador PWM para conversores multiníveis do tipo

cascata de pontes H - entretanto, apenas o core PWM multinível (implementado em VHDL) é apresentado em uma aplicação em malha aberta, sem processamento, em um kit DE0-nano (com FPGA Altera/Intel).

Visando aplicações mais complexas (em que algoritmos de processamento precisem ser executados), Sant'Ana et al. (2021) propõe um sistema de placa mãe em que se integram (via comunicação SPI) um FPGA/SoC Zynq7020 (do fabricante Xilinx/AMD) com um DSP TMS320F28379D (do fabricante Texas Instruments). A ideia era deixar toda a parte de processamento matemático para o DSP (que possui grande poder computacional, mas possui poucos pinos de I/O) e toda a parte de controle de chaveamentos para o FPGA (que possui muitos pinos de I/O - e é reconfigurável). O Zynq7020 ainda apresenta a vantagem de possuir (integrado no mesmo chip) um processador ARM dual core, em uma configuração conhecida como System on Chip (SoC). O ARM pode executar um sistema operacional Linux e ter toda a conectividade de rede desse sistema. Além da placa mãe com FPGA/SoC e DSP, Sant'Ana et al. (2021) também apresenta uma série de outras placas para a implementação de conversores multiníveis (todas isoladas eletricamente por fibras ópticas, visando mais segurança). Uma aplicação de conversor MMC bifásico com 13 níveis em média tensão (2 kV) foi apresentada - entretanto, não havia sido implementado o balanço entre as tensões individuais em cada submódulo do conversor.

Este trabalho de IC é parte de um projeto maior, onde se pretende realizar o balanceamento das tensões dos submódulos de forma "sensorless". Esse primeiro ano de IC visou o treinamento do discente nas ferramentas Xilinx (utilizando uma placa chinesa de baixo custo com o FPGA/SoC Zynq7020, chamada ALINX), com uma implementação de conversor DC-DC do tipo buck em malha fechada. Nos próximos 17 meses de IC (FAPEMIG), o discente irá trabalhar no sistema de placas FPGA/DSP e submódulos, visando o balanceamento "sensorless".

Metodologia

Inicialmente, foi feita a familiarização do discente com ambiente Vivado (para a síntese lógica no FPGA). Com base em postagens antigas de um blog do orientador, Sant'Ana (2019), o discente realizou adaptações dos desenvolvimentos do blog (que utilizava ferramentas Altera/Intel) para as ferramentas Xilinx/AMD. O discente aprendeu a criar um projeto simples, fazer a síntese do circuito, verificar o circuito gerado, simular o circuito no Vivado, definir os pinos de I/O da placa que são ligados ao circuito gerado, gravar o bitstream na RAM da placa e executar o circuito (lendo de chaves e escrevendo em LEDs).

Em seguida, o discente aprendeu como instanciar códigos VHDL dentro de códigos Verilog (e vice-versa), visando o reaproveitamento de códigos (incluindo códigos disponíveis no OpenCores.org). O discente também realizou implementações mistas Verilog-VHDL.

Com base em Sant'Ana et al. (2018) e nas postagens do blog Sant'Ana (2019), o discente implementou um core PWM simples (de comparação de um sinal de referência com uma portadora triangular). Foi feita a simulação no Vivado desse core. Em seguida esse core foi instanciado quatro vezes, com as saídas sendo ligadas a LEDs da placa. Com entradas de referência (duty-cycles) diferentes em cada PWM, foi possível controlar a luminosidade dos LED (luminosidade variando de acordo com o duty-cycle especificado como referência).

Com base em instâncias do core PWM simples, o discente implementou um core PWM bipolar para ponte H de dois níveis e um core PWM unipolar para ponte H de três níveis (ambos testados na simulação do Vivado e verificados os pinos de saída com analisador lógico).

Com base em instâncias do PWM unipolar, o discente implementou um modulador PWM multinível para cascata de pontes H (testado na simulação do Vivado e com verificação dos pinos de saída com analisador lógico).

Visando utilizar o processador ARM do SoC Zynq, o discente realizou testes “bare metal” (sem Linux, como se fosse um microcontrolador comum). Para a programação do ARM foi utilizada a IDE Vitis (também da Xilinx). O discente foi capaz de escrever na saída serial (visualizada no terminal PuTTY). Também foi possível fazer a integração do ARM com o FPGA (ARM bare metal controlando o core PWM no FPGA).

Em seguida, o discente aprendeu como fazer o Linux do ARM controlar o FPGA (inclusive com

reconfiguração “à quente”, sem necessidade de reboot). Através de comandos recebidos pela rede (do Linux no ARM) foi possível controlar o modulador PWM do FPGA.

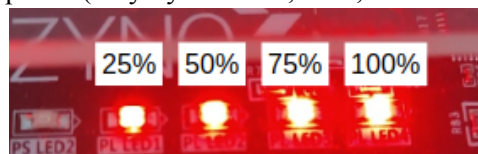
Em seguida, o discente aprendeu a ler os sensores onboard da placa ALINX e pinos de entrada do conversor analógico para digital (ADC). Este ADC é um pouco limitado (com entrada variando apenas de 0 a 1V*faixa que poderia ser estendida com amplificadores operacionais), mas foi útil para o desenvolvimento de uma aplicação em malha fechada.

Para essa aplicação, foi utilizado um conversor DC-DC do tipo buck, em que um duty-cycle de entrada produz uma variação proporcional em uma tensão. Inicialmente foi feito um teste em malha aberta, onde um valor de duty-cycle (recebido pelo Linux) controlava o core PWM, cuja saída controlava o conversor buck. Em seguida, o discente implementou um controlador em malha fechada do tipo PID, de forma que a saída se mantivesse fixa independente de perturbações (variações de carga). Os principais resultados são apresentados a seguir.

Resultados e discussão

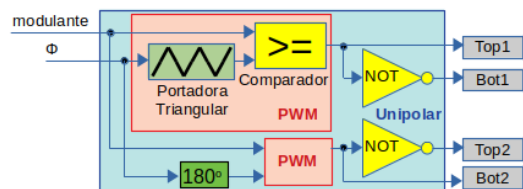
Conforme descrito na seção anterior, foi implementado um core PWM simples, que serviu de base para várias outras implementações ao longo dessa pesquisa. Um teste visualmente interessante desse core é o controle de luminosidade de LEDs. A Figura 1 apresenta uma fotografia dos LEDs ligados ao FPGA da placa ALINX. Na condição mostrada, foram utilizadas 4 instâncias do core PWM desenvolvido, cada uma delas com um duty-cycle diferente (visando luminosidades diferentes nos LEDs).

Figura 1 – Controle de luminosidade dos LEDs da placa (duty-cycles 25%, 50%, 75% e 100%).



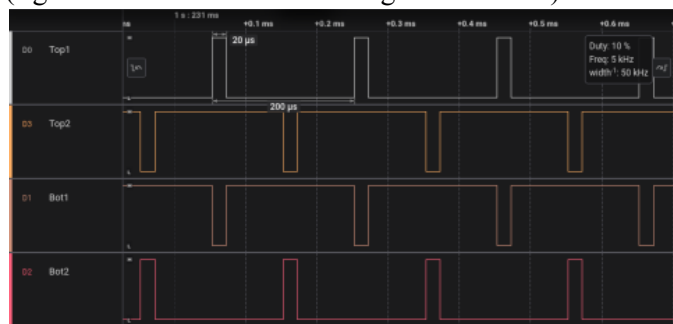
Em seguida, o core PWM foi instanciado duas vezes de forma a se obter a modulação unipolar para ponte H em 3 níveis, conforme Figura 2. Observa-se a repetição do bloco PWM e observa-se que o segundo PWM é configurado para portadora triangular defasada em 180° em relação ao primeiro.

Figura 2 – Bloco Unipolar, obtido com 2 instâncias do bloco PWM.



O bloco unipolar foi testado para um duty-cycle fixo na referência modulante. A Figura 3 apresenta as saídas de 4 pinos do FPGA, executando o bloco PWM unipolar. Na condição mostrada, tem-se uma referência de 10% no duty-cycle. Observa-se que os sinais Top1 e Bot1 (assim como Top2 e Bot2) são complementares entre si (requisito para não curto circuitar as chaves de cada “perna” do conversor, que são meias pontes).

Figura 3 – Pinos de saída do modulador PWM unipolar (figura obtida com analisador lógico SALEAE).



obs: Os sinais apresentados não possuem dead-time, pois normalmente esses tempos já estão integrados nos gate drivers, mas poderiam facilmente ser acrescentados como um bloco à parte no FPGA (tal qual apresentado em Sant’Ana (2019)).

Foram feitos testes de leitura dos sensores onboard da placa ALINX, que também possui disponível em seus conectores de I/O alguns pinos para entrada analógica. Visando uma aplicação em malha fechada, um desses pinos é utilizado para medir a saída de um conversor DC-DC do tipo buck, enquanto que o bloco PWM foi utilizado para controlar o duty-cycle do conversor. A Figura 4 apresenta uma fotografia da montagem do conversor buck e sua ligação ao FPGA da placa ALINX.

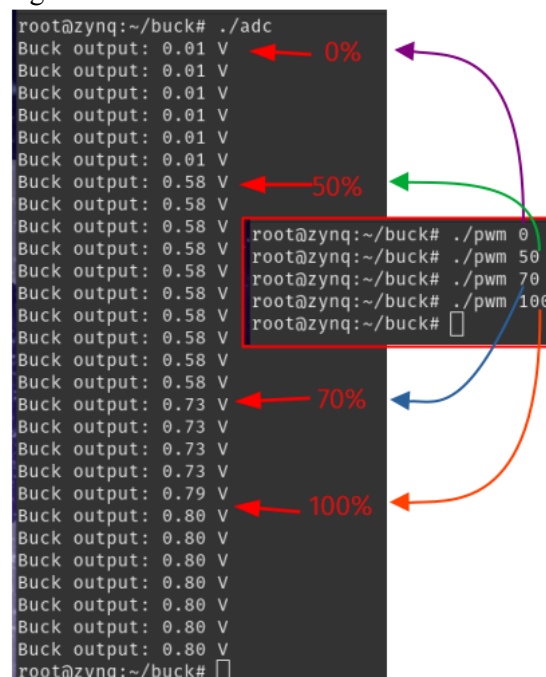
De forma a se testar esse conversor, inicialmente foi feito um controle em malha aberta, onde um dado duty-cycle correspondia a uma tensão de saída no conversor. Foram elaborados dois programas para esse teste (ambos sendo executados no Linux do ARM, que controla o PWM do FPGA). O primeiro programa controla o duty-cycle do conversor e o segundo faz uma varredura no pino de entrada analógica

e mostra o resultado no terminal. A Figura 5 apresenta a execução (simultânea, em 2 janelas de terminal SSH) dos 2 programas.

Figura 4 – Montagem do conversor buck e sua ligação à placa ALINX.



Figura 5 – Teste em malha aberta do conversor buck.

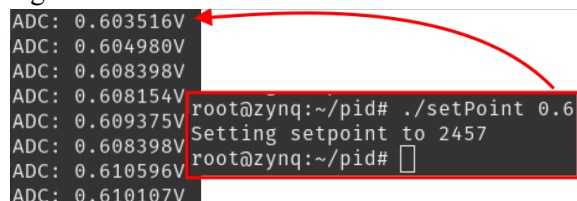


O programa “adc” executa uma varredura contínua (a cada 1 segundo) no pino do ADC e escreve os resultados na tela (até o programa ser finalizado, pressionando qualquer tecla). Já o programa “pwm” configura o duty-cycle de saída do modulador PWM (que está ligado ao pino de controle do conversor) de acordo com o argumento recebido (por exemplo “pwm 30” configura um duty-cycle de 30% e o FPGA fica nessa condição indefinidamente, até que uma nova execução do programa “pwm” modifique o duty-cycle). Observa-se na Figura 5 que o conversor apresentou por volta de 0,8V para duty-cycle 100%, 0,73 para 70% e 0,58V para 50%. Isso indica uma não linearidade desse conversor. Na verdade, esse conversor foi adaptado de

uma outra aplicação e seus detalhes não fazem parte do escopo desse trabalho. Ainda mais quando se considera que na operação em malha fechada será definida uma referência de tensão e o controlador PID irá ajustar automaticamente o duty-cycle visando a saída naquele valor.

Visando o controle em malha fechada, foi utilizado um core PID. Após alguns testes com ganhos diferentes, empiricamente se determinou que os ganhos seriam $k_p=100$, $k_i=0,01$ e $k_d=0$. Ao invés de se determinar um duty-cycle de referência (como no caso em malha aberta), foi escrito um programa “setPoint” que aceita como argumento um valor de tensão (entre 0V e 1V, que são os limites do ADC - na verdade, devido à limitação do conversor buck utilizado a máxima tensão satura em 0,8V) e o duty-cycle é ajustado automaticamente pelo controlador. Tem-se na Figura 6 a execução (simultânea, em 2 janelas de terminal SSH) dos programas “setPoint” e “adc”. Observa-se que, independentemente da não linearidade do conversor buck utilizado, a saída desejada é atingida (e, mais importante do que isso, independentemente do valor da carga do conversor).

Figura 6 – Teste em malha fechada do conversor buck.



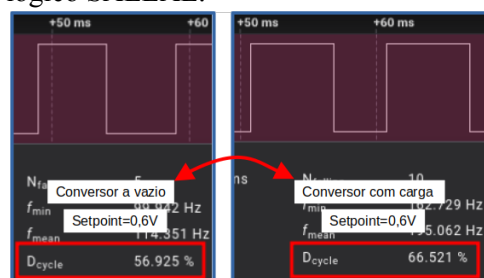
```

ADC: 0.603516V
ADC: 0.604980V
ADC: 0.608398V
ADC: 0.608154V
ADC: 0.609375V
ADC: 0.608398V
ADC: 0.610596V
ADC: 0.610107V

root@zynq:~/pid# ./setPoint 0.6
Setting setpoint to 2457
root@zynq:~/pid#
  
```

Propositalmente, a carga do conversor foi variada. Em malha aberta isso acarretaria em redução da tensão gerada (devido ao aumento de corrente). Já em malha fechada, se observou que a tensão de saída se manteve constante e que o controlador havia forçado aumento no duty-cycle para compensar o aumento da corrente. A Figura 7 apresenta a variação de duty-cycle para as condições a vazio e com carga, para set point em 0,6V.

Figura 7 – Estimação de duty-cycles com o analisador lógico SALEAE.



Conclusões

Este trabalho documentou o aprendizado do discente de IC nas ferramentas FPGA/SoC da Xilinx. Foram estudados os recursos dos ambientes Vivado e Vitis. Foram realizados testes tanto com o FPGA puro quanto testes com o ARM controlando o FPGA (tanto como “bare metal” quanto com Linux embarcado). Uma aplicação de controle de um conversor DC-DC do tipo buck (em malha fechada) foi apresentada, unindo tanto os testes de modulador PWM quanto conversão analógico para digital. Independente da carga do conversor, foi possível se obter uma tensão igual a estabelecida como referência.

Agradecimentos

Os autores agradecem à Universidade Federal de Itajubá e à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG).

Referências

Gama et al. (2021) "FPGA Prototyping Using the STEMLab Board With Application on Frequency Response Analysis of Electric Machinery," in IEEE Access, vol. 9, pp. 26822-26838, 2021, doi: 10.1109/ACCESS.2021.3058059.

Menghal e Laxmi (2010), “Real time simulation: Recent progress & perspective,” in Proc. Conf. Rec. Int. J. Electr. Eng. Electr. Syst., vol. 3, no. 1, 2010, pp. 45–59.

Nane et al. (2016), "A Survey and Evaluation of FPGA High-Level Synthesis Tools," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 10, pp. 1591-1604, Oct. 2016, doi: 10.1109/TCAD.2015.2513673.

Sant'Ana et al. (2018) "Implementação em FPGA de modulador PWM para conversores multiníveis." In Congresso Brasileiro de Automática-CBA, vol. 1, no. 1. 2018.

Sant'Ana (2019), “Blog Doing Tech Myself, label DE0nano” <http://doingtechmyself.blogspot.com/search/label/DE0nano> , de março 2018 a maio 2019.

Sant'Ana et al. (2021) “Development of a modular educational kit for research and teaching on power electronics and multilevel converters” IEEE Access, 9:127496–127514. doi: 10.1109/ACCESS.2021.3112531.