

Relatório do projeto de Laboratórios e Sistemas Digitais

Radiador

João Alcatrão: 76763

Miguel Oliveira: 108772

Introdução

O projeto que nos foi atribuído apresenta como objetivo a realização, em linguagem VHDL aplicável ao kit de desenvolvimento Altera DE2-115, de um controlador para radiadores de aquecimento doméstico.

Para que este projeto fosse bem desenvolvido, foram-nos apresentadas também as seguintes funcionalidades essenciais que devemos implementar para o funcionamento apropriado de um radiador:

-controlador on-off: componente que determina quando se deve ligar ou desligar o radiador. A ativação do radiador é uma função da temperatura ambiente (T_{amb}) e de uma temperatura de referência (T_{ref}), e ainda de uma janela de histerese Δ determinada por nós mesmos (decidimos que a janela Δ seria de 0.2°C ; duas vezes maior que a menor unidade de temperatura a ser considerada, que é a décima). A função que determina o comportamento do radiador é a seguinte: se $T_{amb} > T_{ref} + \Delta/2$, o radiador liga. Se $T_{amb} < T_{ref} - \Delta/2$, o radiador desliga. Nos outros casos, o radiador permanece no estado em que estava;

-Programação horária: a Tref mencionada acima varia de acordo com um perfil horário (ciclo diário) pré-determinado, e apresenta 3 valores distintos: Tgelo = 4°C, Tnoite = 16°C e Tsol = 20°C.

A Tref assume o valor de Tgelo entre as 20h e a meia noite, e entre a meia e as 5h da manhã. Assume o valor Tnoite entre as 5h e as 7h, e entre as 18h e as 20h. E assume o valor Tsol nas restantes horas do ciclo diário (entre as 7h e as 18h).

Este perfil horário está dividido em 48 blocos de meia hora, e para endereçar este perfil em função da hora do dia, é necessário um relógio sob o formato HH:MM ou D:HH:MM (no nosso caso, foi utilizado o último), que deve poder ser acertado pelo utilizador.

Estes são as duas funcionalidades base para a modelação do nosso radiador. Mas para as implementarmos, precisamos de implementar um conjunto de componentes para as suportar e interligar. Com as recomendações do guião providenciado, e com as necessidades que foram surgindo ao longo do trabalho, implementámos os seguintes componentes, organizados na arquitetura apresentada abaixo.

Arquitetura

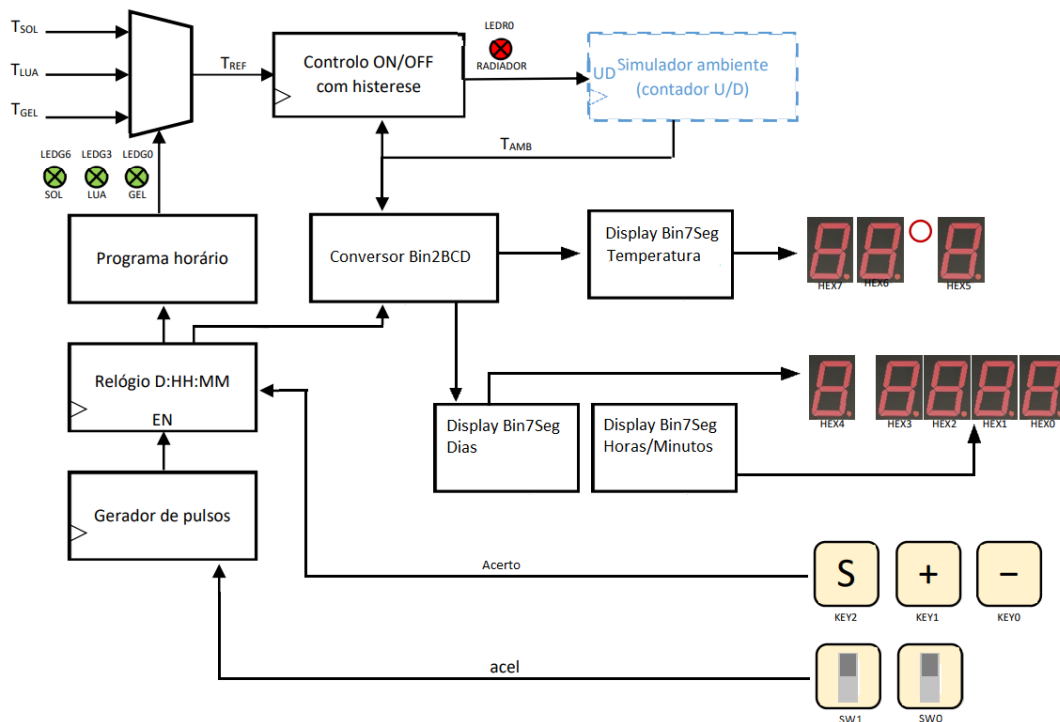


Fig1. Arquitetura do radiador.

Para que o radiador em si possa desempenhar corretamente a sua função, é necessário que receba constantemente informação sobre a temperatura ambiente. Para tal, foi também modelado um simulador, para representar a temperatura ambiente, e que a calcula em função do estado do radiador da seguinte maneira: se o radiador estiver ligado, o simulador aumenta, conforme o seu contador interno (a cada 6 pulsos), a T_{amb} em 0.1°C relativamente ao que estava. Caso contrário, o simulador diminui a T_{amb} , segundo o contador (a cada 11 pulsos), em 0.1°C (o contador simplesmente permite que o aumento ou decréscimo da temperatura não seja feito a cada pulso, como é o caso dos minutos). Tanto o crescimento como o decréscimo da T_{amb} ocorrem de forma linear.

O outro input que o radiador precisa para funcionar conforme nos é pedido, é a temperatura de referência, T_{ref} . Esta é dada por uma ROM com 48×2 (um para a semana, outro para o fim de semana) endereços de dados, que correspondem à T_{ref} correspondente a cada meia hora do ciclo diário.

O valor de T_{ref} dado pela ROM depende do endereço que for selecionado, que em si depende do valor temporário que recebe, e recebe-o do relógio, uma componente bastante complexa, que inclui uma máquina de estados (com 4 estados: normal, acerto de minutos, acerto de horas, e acerto de dias) e um contador interno, que conta os minutos, as horas e os dias num ciclo semanal.

É importante salientar que o contador do relógio está ativo no estado normal, sendo que o tempo não muda quando o relógio se encontra num estado diferente deste.

Por sua vez, o relógio incrementa os minutos (e caso se chegue aos 60 minutos, as horas, e caso se chegue às 24 horas, os dias) a cada pulso que recebe do gerador de pulsos (isto é, no estado normal do relógio apenas).

O relógio recebe ainda inputs de acerto para mudar de estado, e inputs para aumentar ou diminuir a unidade de tempo em questão em cada um destes estados.

O gerador de pulsos, por sua vez, emite um pulso dependendo da frequência selecionada, que pode ser 1 de 4: frequência normal (1 pulso por minutos), frequência de 1Hz, frequência de 20Hz, e ainda frequência de 120Hz.

Por fim, o conversor Bin2BCD transforma as sequências binárias da temperatura e do tempo em sequências BCD, para que sejam apresentadas em displays hexadecimais. O valor da temperatura é representado em 9 bits, os minutos em 6, as horas em 5, e os dias em 3, pelo que o conversor devolve sequências de 12 bits, 8, e 4 bits (múltiplos de 4, como os displays hexadecimais requerem) para posterior apresentação nos displays.

Implementação

Tendo a arquitetura descrita acima em conta, temos os seguintes componentes, com os seguintes inputs/outputs e funções:

Radiador (controlador ON/OFF): recebe como inputs o sinal de clock/pulso, e os valores de T_{amb} e T_{ref} (`std_logic_vector(8 downto 0)`, para representa a temperatura com valores estritamente positivos, e com uma casa decimal), e produz como output um sinal (`std_logic`) a indicar o seu estado. O seu estado muda se o radiador estiver desligado e se verificar que $T_{amb} < T_{ref} - \Delta/2$, ou se o radiador estiver ligado e se verificar que $T_{amb} > T_{ref} + \Delta/2$.

O radiador em si uma máquina de estados, cujo diagrama se segue:

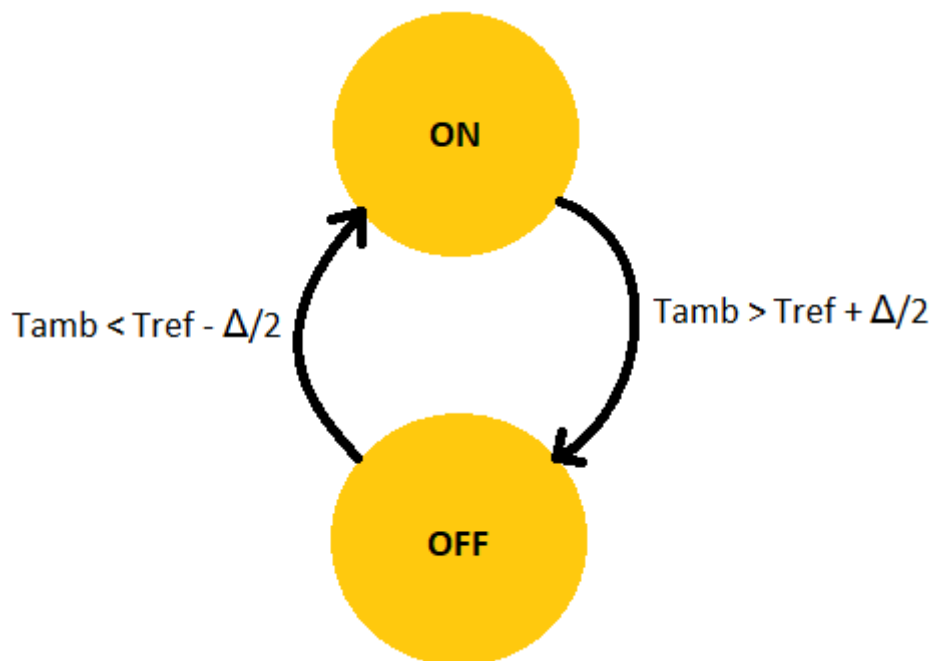


Fig2. Diagrama da máquina de estados do radiador.

Simulador: recebe como input o sinal de clock/pulso, e o output do radiador, e devolve como output o valor de Tamb. O simulador aumenta e diminui a Tamb de forma linear, sendo que, recebendo um input ON do radiador, aumenta a Tamb em 0.1°C a cada 6 pulsos (se o input permanecer ON nesses 6 pulsos) e, caso receba um input OFF, diminui a Tamb nesse mesmo valor a cada 11 pulsos (se o input permanecer OFF durante esse período).

Segue-se abaixo uma waveform com o funcionamento esperado (inputs são clk e tref, outputs são Tamb e on_off) do radiador e simulador.

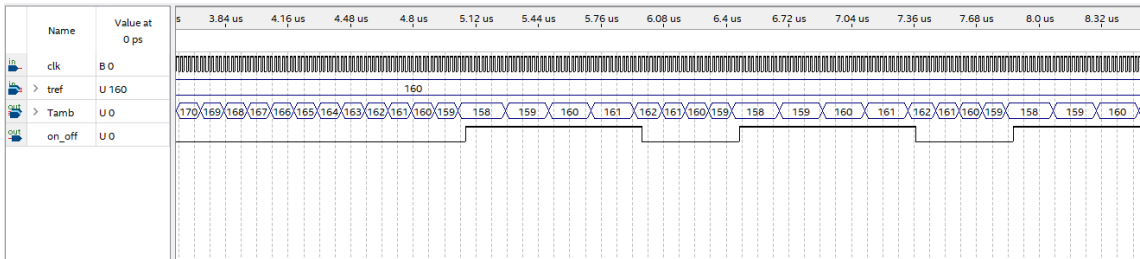


Fig3. Waveform do funcionamento do radiador e simulador.

ROM: memória de leitura com 2 tabelas de endereços, que recebe como input o valor do tempo, e produz como output um valor de Tref. O output depende do endereço selecionado pela ROM, que é determinado pelo valor de tempo recebido da seguinte maneira: se o valor dos dias do tempo corresponder a um dia dos primeiros 5 dias da semana, seleciona um valor da primeira tabela, correspondente ao endereço que é o valor da soma “número de horas” * 2 + “número de minutos” // 30 (“//” aqui representa a divisão inteira).

Se o valor dos dias corresponder a um dia de fim-de-semana, o mesmo processo ocorre mas é selecionado um valor da segunda tabela, sendo que o endereço selecionado em ambos os casos representa 1 de 3 possíveis valores de Tref (que se encontram descritos na Introdução, secção da Programação Horária).

Relógio (máquina de estados, debouncer e contador): máquina de estados finitos com 4 estados. Fundamentalmente, trata-se de três contadores ascendentes, sem sinal (unsigned), responsáveis por manter o valor dos minutos, horas e dia; ao receber um pulso do gerador de pulsos incrementa por um o valor dos minutos. Tem também a lógica necessária para incrementar as horas e dias quando se chega aos 60 minutos e às 24 horas, respetivamente.

Em relação à máquina de estados, o estado de funcionamento “normal” é descrito acima, os restantes estados servem para ajustar independentemente os contadores dos minutos, horas e dias. Para alterar o estado atual é usado o botão “S” associado à KEY2, com os restantes botões incrementando ou decrementando o contador selecionado. Para além disso tem também a lógica necessária para limitar os valores de cada sinal: entre 0 e 59 para os minutos, entre 0 e 23 para as horas e entre domingo a sábado para os dias.

Por fim, com base nos valores dos sinais dos minutos, horas e dia é calculado o índice para selecionar o programa na ROM.

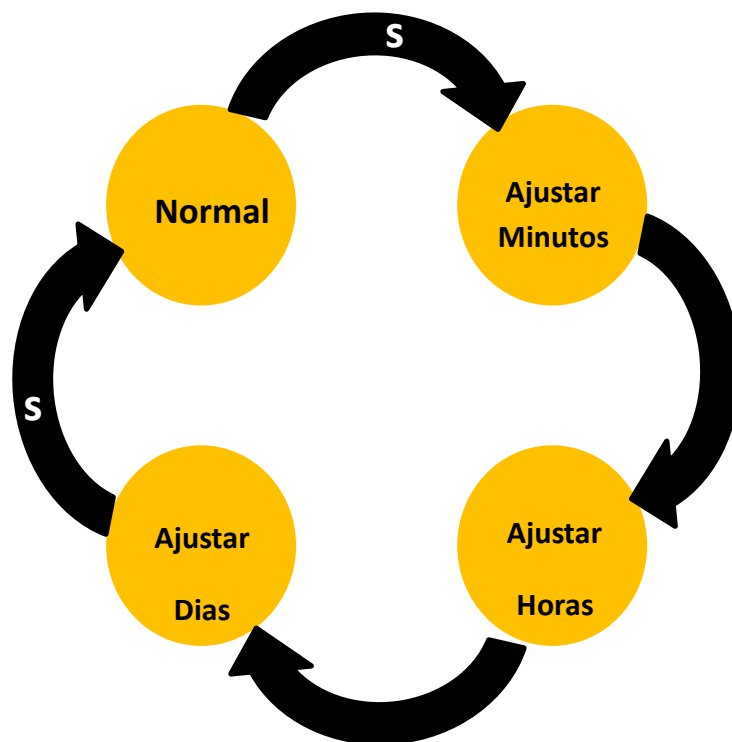


Fig4. Diagrama da máquina de estados do relógio.

Gerador de pulsos: unidade que recebe como input um sinal de clock (mais especificamente, o sinal de clock de 50MHz da FPGA usada nas aulas e no projeto), um valor de frequência/acelerador (que o utilizador pode escolher usando os switches da FPGA; mais informação será apresentada no Manual de Utilizador), e que devolve como output um pulso correspondente à frequência desejada.

Manual de Utilizador

LED R0: estado do radiador. Se este tiver ON, também estará o LED.

LED G 0,3,6: Tref em vigor. O LED G0 ativa quando o Tref for Tgelo, o LED G3 ativa quando o Tref for Tlua, e o LED G6 ativa quando o Tref for Tsol.

HEX 5,6,7: valor da temperatura em decimal, sendo que o HEX5 representa a casa das décimas (e o HEX6 a casa das unidades, e o HEX7 a casa das dezenas).

HEX 0,1,2,3,4: valor do tempo. Os HEXs 0 e 1 representam os minutos, os HEXs 2 e 3 as horas e o HEX4 o dia da semana.

SW 0,1: acelerador de frequência. O valor 0 (ambos os switches desligados) está associado à frequência de 1 pulso por minuto, o valor 1 (switch 1 desligado, switch 0 ligado) está associado à frequência de 1Hz (1 pulso por segundo), o valor 2 (switch 1 ligado, switch 0 desligado) está associado à frequência de 20Hz, e o valor 3 (switches 0 e 1 ligados) está associado à frequência de 120Hz.

KEY 0,1,2: modo de acerto do relógio. A KEY2 altera o estado do relógio para o próximo (normal ->acerto_minutos ->acerto_horas ->acerto_dias), e as KEYS 1 e 0 permitem incrementar ou decrementar o valor do tempo em acerto, respetivamente.

Conclusão

O radiador funciona como pretendemos que funcione e como foi pedido, tal como o simulador e o seu comportamento linear, a ROM, o relógio (e todos os seus estados), e o gerador de pulsos. Outras funcionalidades não essenciais que foram sugeridas, como o piscar dos valores dos dias, horas e minutos em modo de acerto, a desativação do display dos dias em modo normal, também foram implementadas com sucesso. Foi ainda implementando um debouncer para possibilitar uma utilização correta e plausível do modo de acerto do relógio.

Os objetivos que foram traçados foram alcançados, o projeto ficou desenvolvido como desejávamos, e o nosso conhecimento sobre programação VHDL, sobre modelação de máquinas de estados, sobre conversão de valores binários para BCD, e sobre a matéria de Laboratórios e Sistemas Digitais no seu global, foi bastante aprofundado. Como tal, estamos muito satisfeitos com o nosso esforço e desempenho, que foi igualmente repartido entre ambos.