

UNIVERSIDADE FEDERAL DE MINAS GERAIS
COLÉGIO TÉCNICO

Lâmpada Cromo-Climatizadora

Introdução aos sistemas lógicos

Professor orientador: Leandro Maia
21/11/2015

Carolina Alves da Rocha Silva – 106B
Augusto Lopes Ferreira França – 203A

Introdução

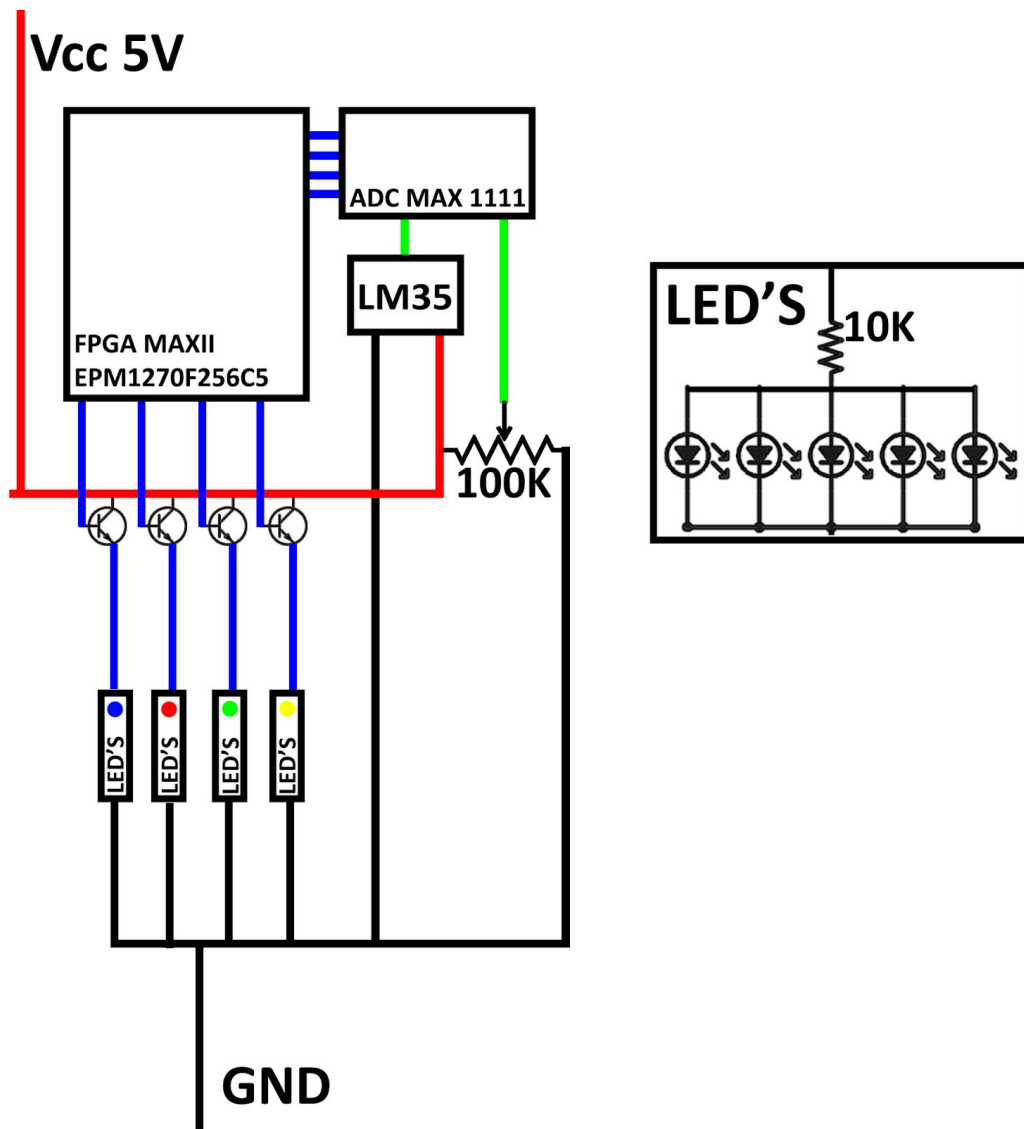
Imagine um deserto, um lugar seco, árido, e muito muito quente, como o Saara. Agora imagine um lugar tão seco quanto, só que frio, muito frio, como a Antártida. Mentalmente, você não conseguiu sentir realmente a temperatura, no entanto você sabe que aquela imagem na sua mente é de um lugar frio. Dito isso pode-se considerar que certas características da imagem podem causar efeitos na noção de temperatura.

Agora, se considerar um ambiente, somente com iluminação branca, e com temperatura variando, poderia reduzir o conforto das pessoas habitando tal espaço se elas passaram muito tempo ali. E um caso em que aconteceria exatamente isso são abrigos nucleares, que manteriam as pessoas confinadas lá por tempo indeterminado, onde estas pequenas diferenças de conforto seriam potencializadas pelo tempo de exposição.

Pensando nisso, foi concebido um conceito de lâmpada pra evitar esse efeito da falta de coloração, que associaria a intensidade de 4 conjuntos de leds, sendo um deles de cada cor, às medidas de temperatura e umidade do ambiente. Seria feita a associação da intensidade dos leds da seguinte maneira : vermelhos em proporção a temperatura, azuis inversamente proporcionais, verdes proporcionais a umidade, e amarelos inversamente, de forma a criar uma climatização de cores do ambiente.

O trabalho

Esquemático do circuito



2.2- Automatic pulse modulation

$$\begin{array}{lcl} 1333200 & \text{-----} & 256 \\ x & \text{-----} & \text{saida_adc_integer} \end{array}$$

$$256x = 1333200 * \text{saida_adc_integer}$$

$$x = \frac{(1333200 * \text{saida_adc_integer})}{256}$$

$$256$$

O auto pulse mod é um componente criado pelo grupo que converte a saída do ADC, que é um vetor de 8 bits, em um número inteiro positivo, já fazendo a regra de 3 mostrada na figura 1. O código está sendo mostrado na figura 2. Mandou-se como genérico para ele o período led, que é uma constante com valor 1333200. Esse valor foi definido para que o led piscasse a 50Hz. A entrada recebida é o vetor que sai do conversor AD e a saída o sinal saída_adc_integer.

```
ENTITY AUTO_PULSE_MOD IS
  GENERIC (PERIOD_W : INTEGER := 6666000);
  PORT (  ENTRADA : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
         CLOCK : IN STD_LOGIC;
         PULSE_W : OUT INTEGER);
END AUTO_PULSE_MOD;

ARCHITECTURE behavior OF AUTO_PULSE_MOD IS

  SIGNAL PULSE_W_I: INTEGER := 0;

  BEGIN

    PROCESS (CLOCK)

    BEGIN
      IF RISING_EDGE(CLOCK) THEN
        PULSE_W_I <= (PERIOD_W/256) * TO_INTEGER (UNSIGNED(ENTRADA));
      END IF;
    END PROCESS;

    PULSE_W <= PULSE_W_I;

  END behavior;
```

Figura SEQ Figura * ARABIC 2: Código do componente auto pulse mod

PWM

Utilizou-se no trabalho um PWM para regular o brilho dos leds de acordo com o potenciômetro. Foi mandado como duty a saída do conversor AD convertida em inteiro pelo componente auto pulse mod e como período máximo foi mandada a constante período_led, que tinha como valor 1333200. Esse valor foi escolhido, porque para que o led piscasse a uma frequência de 50 Hz.

A entidade do componente pwm está sendo mostrada na figura 1 e o modo como foi mandado valores para ele na entidade top, na figura 2.

```
-----COMPONENTES RELACIONADOS AO PWM-----
APM1: auto_pulse_mod GENERIC MAP (período_led) PORT MAP (saida_adc,clock_adc, saida_adc_integer);
PWM1: pwm GENERIC MAP (período_led) PORT MAP (clk, saida_adc_integer, saida_pwm);
```

Figura 4: Mandando valores para o componente

Demultiplexador

Para que se pudesse demonstrar o funcionamento do controle de pwm dos leds, utilizou-se um demultiplexador com quatro botões de controle para apresentar configurações diversas para o funcionamento dos leds, mostrados na tabela abaixo:

Botão 1	Botão 2	Botão 3	Botão 4	Vermelho	Azul	Verde	Amarelo
1	1	0	1	PWM	Apagado	Apagado	Apagado
1	0	0	1	Apagado	PWM	Apagado	Apagado
0	1	0	1	Apagado	Apagado	PWM	Apagado
0	0	0	1	Apagado	Apagado	Apagado	PWM
--	--	0	0	PWM	PWM	PWM	PWM
--	--	1	0	PWM	NOT PWM	PWM	NOT PWM

É importante lembrar que os botões têm lógica invertida. Assim, quando colocamos que o botão 1 está em 0, queremos dizer que ele está apertado.

O sensor DHT11

O sensor digital¹ DHT11 requer uma alimentação de 3,0 a 5,0 VDC (voltage direct current), sendo 5,5 a VDC máxima. A corrente necessária para o seu funcionamento é de 200uA a 500mA e em stand-by de 100uA a 150uA. Ele tem uma faixa de medição de umidade entre 20 e 90% UR e uma precisão de $\pm 5,0\%$ UR. O sensor consegue medir temperaturas de 0°C a 50°C com uma precisão de $\pm 2,0^\circ\text{C}$. O tempo de resposta é menor que 5 segundos.

O DHT11 possui 4 pinos, sendo o primeiro deles o VCC, o segundo aquele que transmite os dados, o terceiro é um N.C. e o quarto o GND. Os dados são transmitidos da seguinte maneira: 8 bits para a parte inteira da umidade, 8 bits para a parte decimal, 8 bits para a parte inteira da temperatura, 8 para a parte decimal e mais 8 bits de checksum, totalizando 40 bits.

Para o seu devido funcionamento, é necessário mandar um sinal em 0 durante 18ms ou mais, seguido de um alto (1) durante 20 a 40 ms. Logo após ele começará a transmitir os dados.

Não foi utilizado pois durante os teste com o FPGA, o funcionamento não se deu como esperado, e como o problema não era do próprio sensor, dado que testes foram feitos com o no Arduíno, e o sensor funcionava como deveria.

O sensor LM35

O sensor de temperatura LM35 trabalha de 4V a 30V utilizando menos de 60μ ampère. O seu aquecimento é em torno de 0.08°C e ele possui 3 pinos, sendo eles: VDC, dados e GND.

Ele trabalha com valores de temperatura de -55°C a 150°C, com uma precisão de $\pm 0,25^\circ\text{C}$. Sua saída é um sinal de 10mV para cada Grau Celsius de temperatura. O LM35 não necessita de qualquer calibração externa ou “trimming” para fornecer com exatidão valores temperatura.

Para um protótipo final utilizaria-se este sensor pra controle de temperatura, mas durante a demonstração foi utilizado um LDR, devido a facilidade de provocar mudanças de intensidade de luz.

O Light Dependent Resistor (LDR)

O LDR é um resistor variável cuja resistência varia conforme a intensidade da luz que incide sobre ele, ou seja, à medida que a intensidade da luz aumenta, a sua resistência diminui.

O LDR é construído a partir de material semicondutor com elevada resistência elétrica. Quando a luz que incide sobre o semicondutor tem uma frequência suficiente, os fótons que incidem sobre o semicondutor libertam elétrons para a banda condutora que irão melhorar a sua condutividade e assim diminuir a resistência.

Sua resistência varia linearmente com a intensidade de luz incidente, obedecendo à equação $R = CLa$, onde **L** é a luminosidade em Lux e **C** e **a** são constantes dependentes do processo de fabricação e material utilizado.

O potenciômetro

O potenciômetro é um componente eletrônico que tem sua resistência elétrica que é alterada manualmente. Também chamado de resistor variável, ou de reostato, eles normalmente possuem três terminais e um eixo giratório que permite variar a resistência.

Esse componente utiliza-se de um ponto de contato que ao mover-se varia a quantidade de material resistivo que a corrente passa desde a entrada, dessa forma criamos um dispositivo que altera a resistência de maneira que possamos mecanicamente controlada. Devido a essa característica o mesmo foi utilizado pra simular as medidas de umidade, visto que não foi encontrado um sensor capaz de repor as medidas de umidade do DHT11.

Conclusões

Com a realização do trabalho, foi possível praticar tanto a parte de código em VHDL quanto a montagem na protoboard. Além disso, tivemos a oportunidade de fixar o conteúdo aprendido nas práticas e fazer algo que nós mesmos escolhemos.

Foi muito difícil dia após dia testarmos o projeto com o FPGA e não funcionar. Tivemos que desistir do sensor que queríamos usar duas semanas antes de entregar o trabalho, e tivemos diversos problemas com transistores, o que acabou se tornando uma boa aprendizagem, já que descobrimos que o PNP não consegue regular uma corrente enquanto controla uma tensão maior do que a sua, logo tivemos que utilizar um NPN.

Porém, ver o trabalho funcionando devidamente, nos causou uma felicidade muito grande, já que estávamos tensos com o fato de na simulação do Quartus tudo acontecer perfeitamente e na vida real não. Assim, apesar de todos os contratemplos, foi um trabalho ao mesmo tempo aterrorizante e divertido.