

ROTEIRO LABORATÓRIO 02 – MODELAGEM DE SISTEMAS DIGITAIS UTILIZANDO LINGUAGEM DE DESCRIÇÃO DE HARDWARE E FPGA

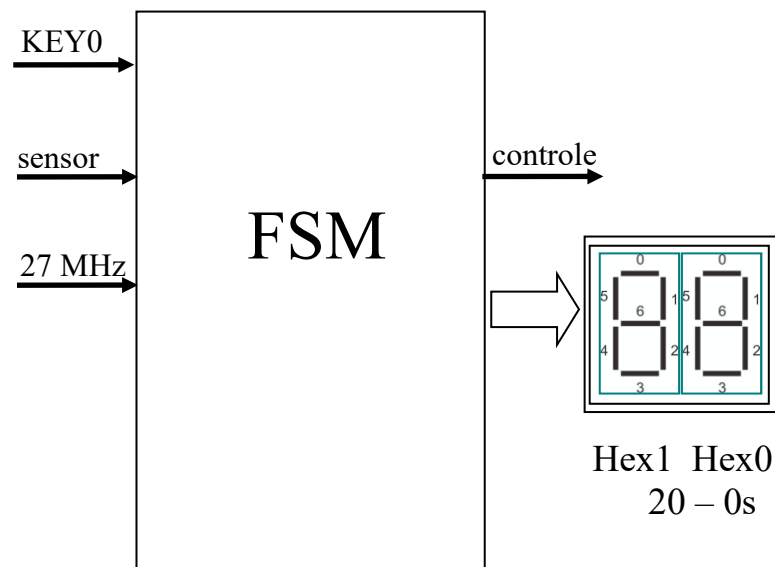
Observações:

- a) Integrantes por equipe: 05 (máximo)
- b) Cada equipe deverá escolher um dos projetos descritos para realizar a implementação;
- c) A apresentação pública dos resultados será realizada por cada equipe de modo presencial no Laboratório A31. Deverão ser apresentados em slides os seguintes itens:
 - Diagrama de estados da FSM;
 - Esquema elétrico dos circuitos analógicos;
 - Demonstração no Kit FPGA.
- d) Datas das apresentações: 10/12/2025;
- e) Tempo das apresentações: 10 minutos por equipe (no máximo);
- e) Nota máxima do Projeto: 10,0 pontos (Segunda Avaliação)
- f) Avaliação final:
 - 15 de dezembro de 2025;
 - Assunto: Máquinas de Estados e Processamento de Vídeo em FPGA.

Observação: Não haverá a partir do dia 12/11/25, pois cada equipe estará livre para implementar o projeto escolhido. Os kits de FPGA estarão disponíveis nos dias e horários da aula, exceto no dia 19/11/25.

Projeto 01 (10,0)

Utilize um sensor de temperatura e o motor para projetar um sistema com a seguinte funcionalidade: Quando for detectado o acionamento do sensor, a saída denominada controle deverá acionar o motor enquanto o sensor estiver acionado. Quando o sensor for desativado, a saída controle deverá continuar acionando o motor durante 20 segundos. O tempo deverá ser exibido no display de modo decrescente. Ao término do tempo, a FSM verifica novamente o sensor e todo ciclo se repete.



KEY0: Inicialização do sistema em nível lógico zero (0). Quando esta chave for pressionada, o display deverá exibir 00.

SUGESTÕES:



- O Sensor De Temperatura Termistor Arduino K275 é um módulo eletrônico utilizado como chave térmica de segurança em diversos tipos de projetos que demandam maior cuidado e proteção de determinados circuitos e/ou ambientes.
- Muito eficiente, o Módulo Chave Térmica K275 é composto por um pequeno NTC, um comparador LM393 e um potenciômetro, onde permite ao projetista ajustar e selecionar determinada temperatura entre 20 e 80°C.
- O Sensor de Temperatura Termistor Arduino K275 possui funcionamento simplificado e funcional, sendo que quando atingir a temperatura pré-ajustada ele vai variar sua saída (0 ou 1) ou vice versa,

conforme for sua programação, permitindo saber quando a temperatura de um circuito atinge o parâmetro escolhido.

ESPECIFICAÇÕES:

- Modelo: K275;
- Comparador: LM393;
- Alimentação: 3.3V a 5V;
- Detecção de temperatura: 20 a 80°C;
- Tipo de saída: Digital (0 ou 1);
- Dimensões: 32x14x6mm;



ESPECIFICAÇÕES:

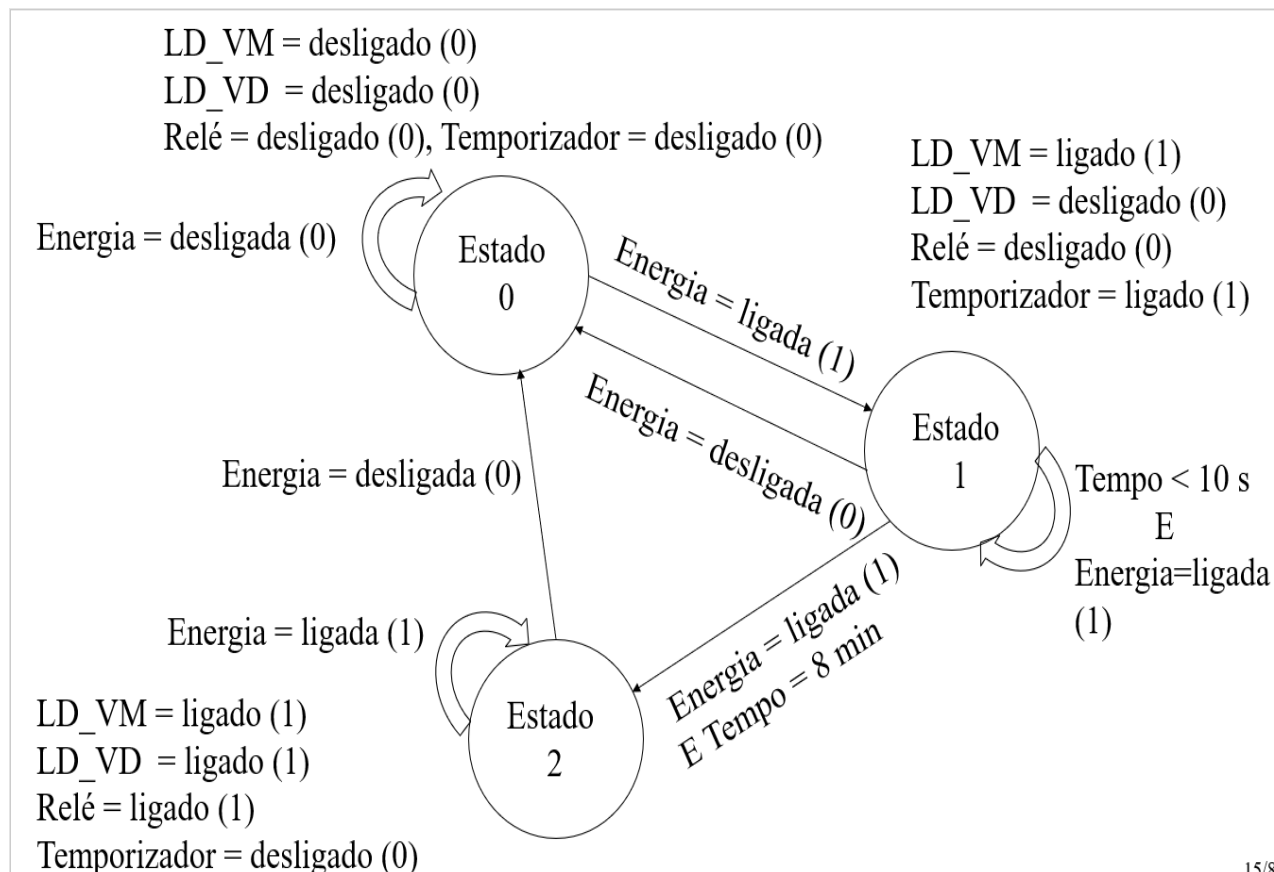
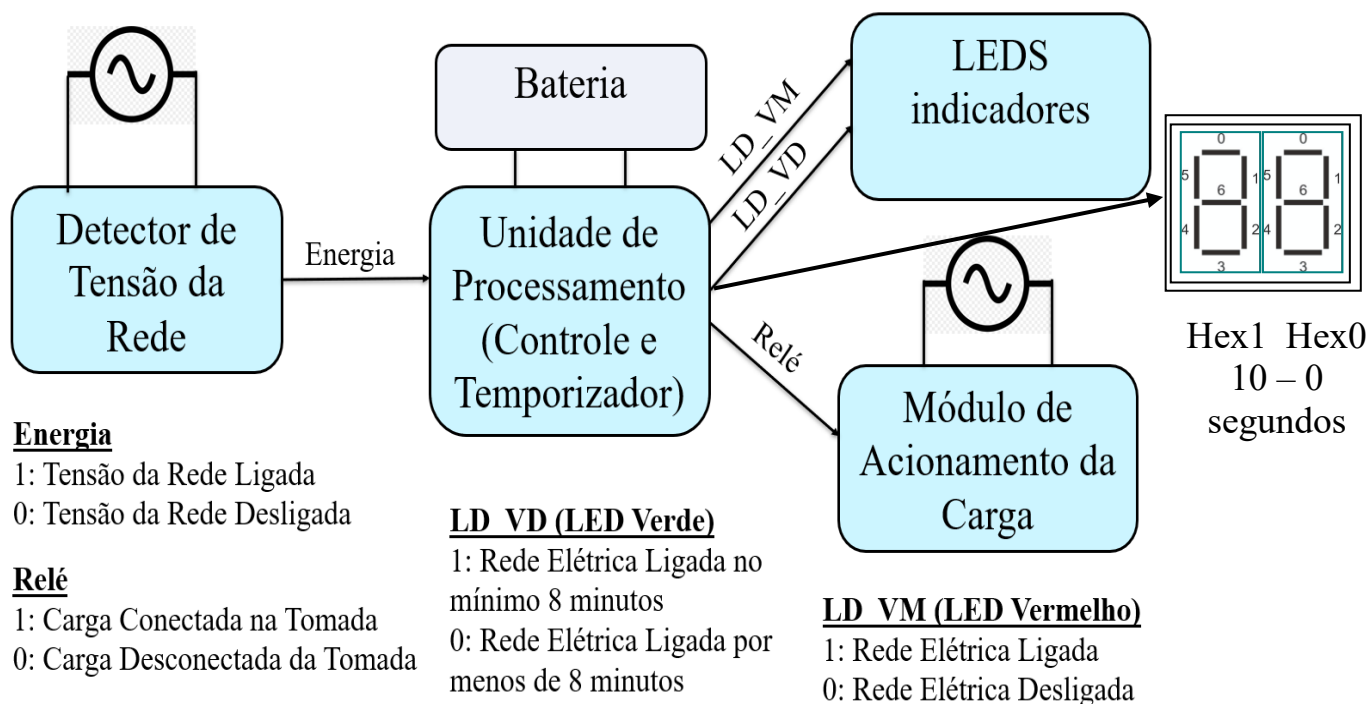
- Tensão de funcionamento: 3 a 6V;
- Corrente sem carga: ~10mA (3V) e ~20mA (6V);
- Diâmetro do eixo: 2.0mm;
- Saída do comprimento do eixo: 75mm;
- Velocidade: ~3000RPM (3V) e ~6000RPM (6V);

- Elabore o diagrama de estados;
- Efetue os testes no kit altera utilizando uma chave para simulação da presença do sensor e uma vez validada a funcionalidade, o sensor deverá ser conectado nos pinos do GPIO.
- A equipe deverá apresentar o projeto completo com o sensor e motor funcionando em conjunto com o Kit FPGA;
- A equipe deverá explicar também o funcionamento dos circuitos analógicos com o respectivo dimensionamento dos componentes.

Projeto 02 (10,0)

Implementar o sistema completo ilustrado na figura abaixo com sua respectiva máquina de estados.

Deve-se exibir o tempo de modo decrescente em um **DISPLAY DE 07 SEGMENTOS**.



OBSERVAÇÕES:

- a) o código apresentado em sala de aula pode ser utilizado como base para implementação do sistema;
- b) A equipe deverá apresentar o sistema completo montado com os circuitos analógicos (sensor de energia) e o circuito com relé para acionamento de uma carga conectada à rede elétrica, por exemplo, uma lâmpada, devidamente interligados as portas de entrada e saída do kit FPGA;
- c) A entrada que receberá o sinal do sensor de energia deverá ser conectada a dois flip-flops em série, implementados no FPGA, para evitar problemas de leitura desse sinal;
- d) O circuito de detecção da tensão de rede elétrica deverá ser isolado da rede elétrica por meio de um opto-acoplador ou transformador;

Projeto 03 (10,0)

Elabore um sistema de processamento de vídeo capaz de realizar as operações descritas na tabela abaixo. A seleção de cada operação deverá ser feita por meio de uma única chave controlada por uma máquina de estados. Quando inicializado, o sistema deverá exibir o vídeo normal (operação 0). Ao pressionar a chave de seleção, o sistema exibirá a imagem em cores com a componente Y somada com 15 conforme ilustrado na tabela abaixo e assim sucessivamente. Os valores com o número da operação deverão ser exibidos em um display de 07 segmentos.

- a) Elabore o diagrama de estados;
- b) Efetue os testes no kit altera DE2_115 utilizando chaves de seleção manual para testes do chaveamento de vídeo sem a máquina de estados inicialmente. Uma vez validado o chaveamento, faça a inserção da máquina de estado para verificar a seleção cíclica usando uma única chave.
- c) Utilize o projeto DE2_115_TV_brilho como base para inserir módulo de chaveamento de vídeo e o clock TD_CLK27 para máquina de estados, ou seja, o chaveamento deverá ser sincronizado.

NÚMERO DA OPERAÇÃO	IMAGEM NA TELA
0	Imagem Normal
1	Y + 15, CbCr
2	Y + 50, CbCr
3	Y + 75, CbCr
5	Imagem Normal
4	Y - 15, CbCr
5	Y - 50, CbCr
6	Y - 75, CbCr

Projeto 04 (10,0)

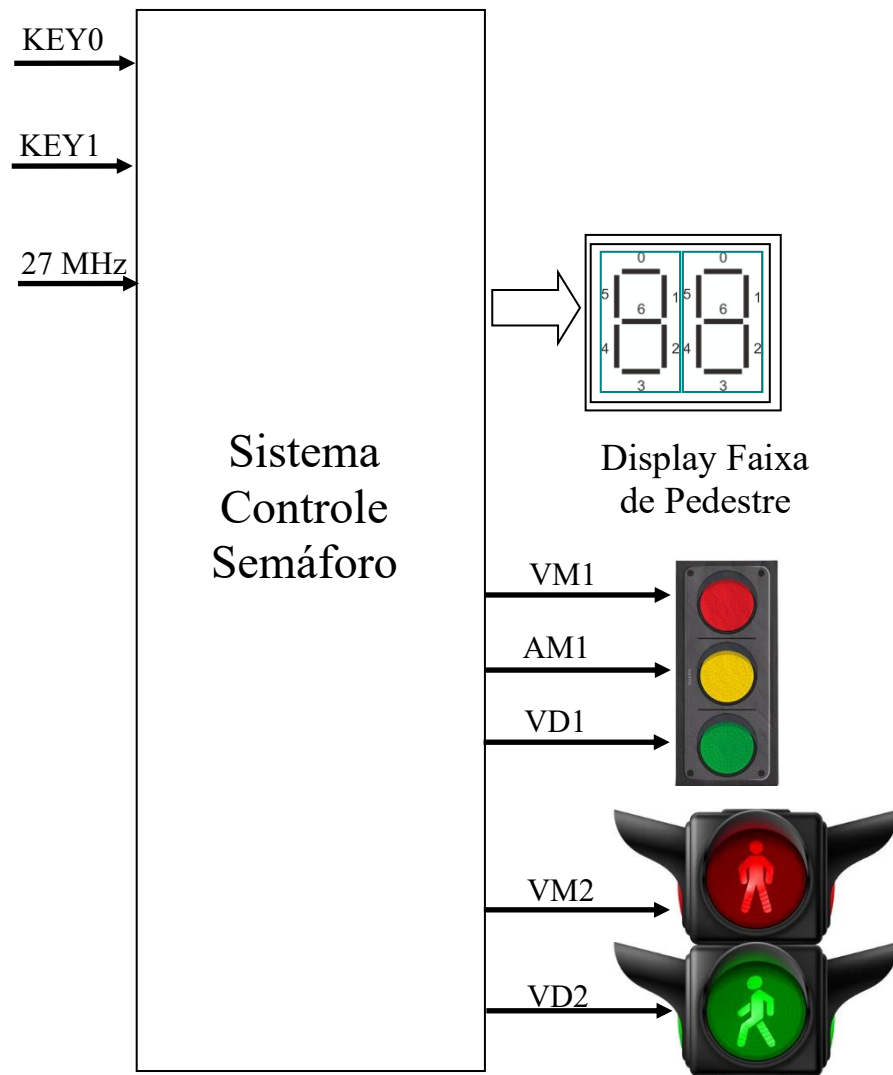
Elabore um sistema de processamento de vídeo capaz de realizar as operações descritas na tabela abaixo. A seleção da funcionalidade deverá ser feita por meio de uma única chave controlada por uma máquina de estados. Quando inicializado, o sistema deverá exibir o vídeo normal. Ao pressionar a chave de seleção pela primeira vez, o sistema exibirá a imagem escala de cinzas baseado na componente R. Quando pressionada pela segunda vez, o Negativo da imagem será exibido e assim sucessivamente até retornar a exibir o vídeo normal.

- a) Elabore o diagrama de estados;
- b) Efetue os testes no kit altera DE2_115 utilizando chaves de seleção manual para testes do chaveamento de vídeo sem a máquina de estados inicialmente. Uma vez validado o chaveamento, faça a inserção da máquina de estado para verificar a seleção cíclica usando uma única chave.
- c) Utilize o projeto DE2_115_Camera como base para inserir módulo de chaveamento de vídeo e o clock D5M_PIXLCLK para máquina de estados, ou seja, o chaveamento deverá ser sincronizado.

OPERAÇÃO	IMAGEM NA TELA
0	Vídeo Normal
1	Imagem em escala de cinzas
2	Imagem negativa
3	Imagem binarizada baseada na componente R (Limiar igual a 1000). Elementos de imagem acima do limiar devem ser destacados em branco. Os que estiverem iguais ou abaixo do limiar devem ser destacados de preto.
4	Imagem binarizada baseada na componente B (Limiar igual a 1000). Elementos de imagem acima do limiar devem ser destacados em branco. Os que estiverem iguais ou abaixo do limiar devem ser destacados de preto.
5	Tela Escura

Projeto 05 (10,0)

Projete o sistema para controle do semáforo ilustrado na figura abaixo, de acordo com as seguintes especificações:



- Elabore o diagrama de estados;
- A equipe deverá apresentar o projeto completo com a chave do pedestre, bem como os semáforos em conjunto com o Kit FPGA;

DESCRIÇÃO DAS FUNCIONALIDADES DO SISTEMA

KEY0: Inicialização do sistema em nível lógico zero (0). Quando esta chave for pressionada, o display exibirá 00.

KEY1: Chave de solicitação de passagem do pedestre. Quando acionada em nível lógico 1, o sistema deverá manter o semáforo de passagem dos veículos em vermelho e no verde o semáforo do pedestre, bem como exibir a contagem decrescente no display de 45 segundos para passagem do pedestre. O sistema deverá primeiramente completar o ciclo de operação corrente para depois atender à solicitação do pedestre. Se não for solicitada a passagem do pedestre o semáforo de passagem dos veículos deverá permanecer no verde e o do pedestre em vermelho.

Ao final da contagem (45 segundos) deverá emitir um bipe com duração de 3 segundos e depois o semáforo para o pedestre ficará no vermelho e de passagem dos veículos no verde.

Especificações de tempo para semáforo de passagem dos veículos quando pedestre acionar a chave KEY1:

- a) Verde 01: 5 segundos
- b) Amarelo 01: 5 segundos
- c) Vermelho 01: 50 segundos

SUGESTÃO:



O buzzer do tipo ativo contém um circuito oscilador embutido, assim basta você energizar o componente para que o mesmo comece a emitir um beep contínuo. Ao conectar o buzzer ativo no circuito, verificar a polaridade (pinos + e -)

Especificações:

- Buzzer tipo ativo
- Tensão de operação: 4 à 8VDC
- Corrente de operação: 30mA
- Saída de som mínima (a 10cm): 85dB
- Frequência de ressonância: 2300±300 Hz
- Temperatura de operação: -27 a +70 °C

Projeto 06: (10,0)

Utilize o sensor de distância ilustrado na figura abaixo para projetar uma FSM que exibe em um display de 07 segmentos com 03 dígitos a distância em centímetros medida pelo sensor. Quando a distância medida for menor ou igual a 25 cm, o sistema deverá acionar um alarme (buzina).



O Sensor de Distância Ultrassônico HC-SR04 é capaz de medir distâncias de 2cm a 4m com ótima precisão e baixo preço. Este módulo possui um circuito pronto com emissor e receptor acoplados e 4 pinos (VCC, Trigger, ECHO, GND) para medição.

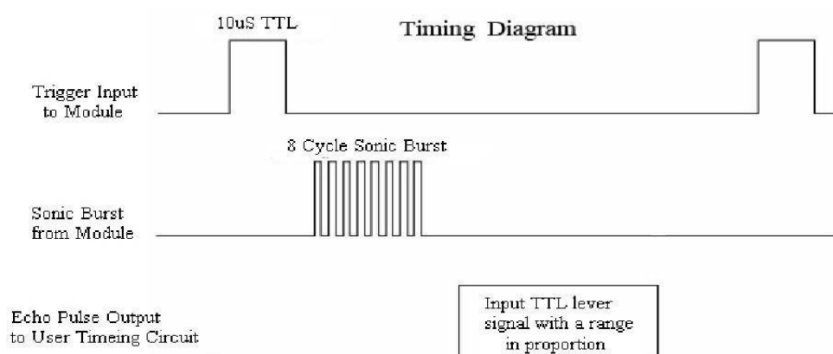
Funcionamento:

Para começar a medição é necessário alimentar o módulo e colocar o pino Trigger em nível alto por mais de 10us. Assim, o sensor emitirá uma onda sonora que, ao encontrar um obstáculo, rebaterá de volta em direção ao módulo. Durante o tempo de emissão e recebimento do sinal, o pino ECHO ficará em nível alto. Logo, o cálculo da distância pode ser feito de acordo com o tempo em que o pino ECHO permaneceu em nível alto após o pino Trigger ter sido colocado em nível alto.

$$\text{Distância} = [\text{Tempo ECHO em nível alto} * \text{Velocidade do Som}] / 2$$

A velocidade do som poder ser considerada idealmente igual a 340 m/s, logo o resultado é obtido em metros se considerado o tempo em segundos. Na fórmula, a divisão por 2 deve-se ao fato de que a onda é enviada e rebatida, ou seja, ela percorre 2 vezes a distância procurada.

$$\text{Distância (cm)} = [\text{Tempo ECHO em nível alto } (\mu\text{s}) / 58]$$



Especificações:

- Alimentação: 5V DC
- Corrente de Operação: 2mA
- Ângulo de efeito: 15°
- Alcance.: 2cm ~ 4m
- Precisão.: 3mm

SUGESTÃO:



O buzzer do tipo ativo contém um circuito oscilador embutido, assim basta você energizar o componente para que o mesmo comece a emitir um beep contínuo. Ao conectar o buzzer ativo no circuito, verificar a polaridade (pinos + e -)

Especificações:

- Buzzer tipo ativo
- Tensão de operação: 4 à 8VDC
- Corrente de operação: 30mA
- Saída de som mínima (a 10cm): 85dB
- Frequência de ressonância: 2300±300 Hz
- Temperatura de operação: -27 a +70 °C

Observações para projetos com sensores na entrada:

A entrada `signal_in` (entrada do sensor) deverá ser conectada a dois flip-flops em série para sincronizar o sinal com o clock do sistema e evitar problemas de leitura. Segue um exemplo dessa implementação.

```
module fsm (input clock, clear, signal_in, .....)  
  
  reg signal_in_last, signal_in_last1;  
  
  always @(posedge clock or negedge clear)  
    if(!clear) begin  
  
        signal_last1 <= 0;  
        signal_last <= 0;  
  
    end  
  
    else begin  
  
        signal_last <= signal_in;  
        signal_last1 <= signal_last;  
  
    end  
  
end
```

Observações para projetos com chaves push-botton para acionamento das operações:

Utilize um detector de transição para efetuar a detecção da mudança de estado da chave conforme o código a seguir. Desta maneira, cada vez que a chave for pressionada, será gerado um pulso na saída do detector de transição da largura do clock. Este pulso poderá ser utilizado para acionar a FSM utilizada para controlar todo o sistema:

```
module detector_trans (input signal_in, clock, clear, output reg edge_detected);  
  
  reg signal_in_prev;  
  
  //----- Detector de transição -----  
  
  always @(posedge clock or negedge clear)  
    if(!clear) begin  
  
        edge_detected <= 1'b0;  
        signal_in_prev <= 1'b0;  
  
    end  
  
    else if( signal_in_prev == 1'b0 && signal_in == 1'b1) begin  
  
        edge_detected <= 1'b1;  
        signal_in_prev <= signal_in;  
  
    end  
  
    else begin  
  
        edge_detected <= 1'b0;  
        signal_in_prev <= signal_in;  
  
    end  
  
endmodule
```

