

Experimento 8. Sistema controle de semáforos

OBJETIVOS:

- Projetar e implementar em FPGA, usando a linguagem VHDL, um sistema de controle de semáforos.
- Implementar contadores em VHDL.
- Aprender a usar mostradores de 7 segmentos.
- Fixar o uso de técnicas de projeto modular em VHDL, desenvolvendo grandes sistemas construídos utilizando outros sistemas menores, interligados entre si.

PRÉ-RELATÓRIO:

Apresentar:

- Código VHDL de uma arquitetura para a entidade *contador10*, que é um contador de módulo 10.
- Diagrama de blocos do contador de módulo 100 (entidade *contador100*), construído usando dois contadores de módulo 10.
- Diagrama de estados da máquina que faz o controle dos semáforos (entidade *maqestados*).
- Código VHDL de uma arquitetura para a entidade *convsemaforo7seg*, que é um conversor para ajustar a saída da máquina de estados para código de 7 segmentos.
- Diagrama de blocos do sistema do visto 3, mostrando como as diferentes entidades serão interligadas (entidade *exp8visto3*).
- Arquivos UCF para cada um dos três vistos.

Lembre-se: seu pré-relatório deve ser entregue como um único arquivo PDF. Não anexe arquivos .vhd ou .ucf!

Os dois diagramas de blocos devem ser desenhados seguindo o estilo das Figuras 1b e 3b a seguir.

VISTOS:

1. Contador módulo 100 com saída BCD (a ser mostrada nos LEDs do kit de desenvolvimento), com entradas de 'reset' e 'load' síncronas, clock de 1 Hz e entrada de 'enable' (ativa em nível baixo).
2. Sistema de temporização do controle de semáforos, com 'reset' síncrono, clock de 1Hz, quatro saídas binárias (a serem mostradas nos LEDs do kit de desenvolvimento), indicando que já se passaram, respectivamente, 5 segundos (T5), 6 segundos (T6), 20 segundos (T20) e 60 segundos (T60), e com o estado do contador sendo apresentado no mostrador de 7 segmentos do kit de desenvolvimento.
3. Sistema de controle de semáforos com três bits de entrada (sensor de carros na direção norte/sul, sensor de carros na direção leste/oeste e chave de liga/desliga do sistema) e saídas apresentadas no mostrador de 7 segmentos do kit de desenvolvimento (luzes verde, amarela e vermelha de cada direção e estado do contador). O cruzamento deverá ficar liberado para determinada direção (luz verde) e bloqueado na outra direção (luz vermelha) por 60 segundos, exceto caso haja carro esperando na direção oposta e não haja carros atravessando na direção em questão e já se tenha passado pelo menos 20 segundos. Antes de trocar a pista liberada, o semáforo deverá mostrar a luz amarela (com luz vermelha no outro semáforo) por 6 segundos e, em seguida, a luz vermelha (em ambos os semáforos) por 5 segundos. A qualquer momento, se a chave de liga/desliga for desativada, os semáforos deverão entrar em estado intermitente, no qual alternarão entre a luz amarela e nenhuma luz acesa, com intervalo de 1 segundo entre cada, devendo voltar a qualquer momento ao funcionamento normal caso a chave seja reativada.

Cada um desses três vistos será detalhado a seguir. Os grupos terão duas aulas para conseguir os três vistos.

Atenção: não haverá teste deste experimento; porém, os vistos deste experimento terão peso dobrado.

DETALHAMENTO:

1. Visto 1

O sistema a ser implementado para o visto 1 é um contador módulo 100 com saída BCD (a ser mostrada nos LEDs do kit de desenvolvimento), com entradas de 'reset' e 'load' síncronas, clock de 1 Hz e entrada de 'enable' (ativa em nível baixo). A caixa preta e o diagrama de blocos da arquitetura interna desse sistema são apresentados na Figura 1. O código VHDL da entidade do visto 1 será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. As entidades *divclock* e *contador100* serão detalhadas a seguir, mas devem ser usadas aqui como componentes.

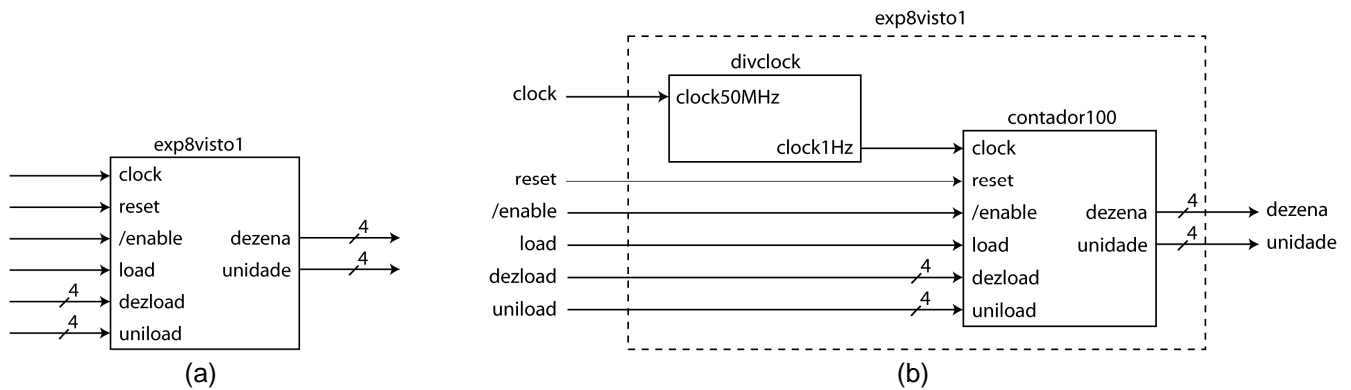


Figura 1: Caixa preta (a) e diagrama de blocos da arquitetura interna (b) do sistema a ser implementado no visto 1.

A entidade *divclock* (Figura 2a) é responsável por reduzir a frequência do clock da placa de desenvolvimento, que é de 50 MHz, para a frequência de trabalho do sistema de controle de semáforos, que é de 1 Hz. O código VHDL da entidade e da arquitetura serão fornecidos pelo professor. Nele, um contador é incrementado a cada ciclo do clock de 50 MHz e, quando atinge-se 25 milhões de ciclos contados (o que corresponde à metade do período do clock de 1 Hz), o estado do clock de saída é invertido. Assim, a cada 50 milhões de ciclos do clock de 50 MHz, teremos 1 ciclo do clock de 1 Hz, conforme desejado.

A entidade *contador100* (Figura 2b) implementa o contador em si. Trata-se de um contador que, a cada ciclo de clock, conta de 0 a 99 em representação BCD, isto é, com duas saídas de 4 bits que representam, respectivamente, o dígito da dezena e o da unidade. O código VHDL da entidade *contador100* será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. Para implementar este contador de módulo 100, você deve utilizar (como componentes) dois contadores de módulo 10. Pesquise sobre como cascatear contadores de módulo 10 de modo a construir contadores de módulo 100, módulo 1000, etc.

O contador de módulo 10 será implementado pela entidade *contador10*, cuja caixa preta é mostrada na Figura 2c. O código VHDL da entidade *contador10* será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. Este contador de módulo 10, deve ser implementado como uma máquina de estados do tipo Moore. Essa máquina terá 10 estados, cada um associado a uma saída Q de 4 bits: de "0000" (decimal 0) até "1001" (decimal 9). As transições de estado serão controladas pelas variáveis de entrada *reset* (que, de forma síncrona, leva a máquina de volta ao estado inicial se *reset* = '1'), *enable* e *RCI* (que são ativas em nível baixo, ou seja, a contagem está ativada quando *enable* = '0' e *RCI* = '0') (*RCI* vem da sigla em inglês *ripple carry-in*), e *load* e *D* (de forma síncrona, a máquina deve ser levada ao estado indicado por *D* se *load* = '1'). A saída *RCO* (sigla do inglês *ripple carry-out*) é ativa em nível baixo, e deverá ser '0' se e somente se *Q* = '1001', caso contrário será '1'. Esta saída é usada para cascatear contadores de módulo 10, de modo a construir contadores de módulo 100, módulo 1000, etc. A ação de 'reset' deve ter prioridade sobre a ação de 'load', que por sua vez deve ter prioridade sobre a contagem. Um exemplo de contador semelhante a este é apresentado no documento "Implementando um contador com reset e load síncronos em VHDL".

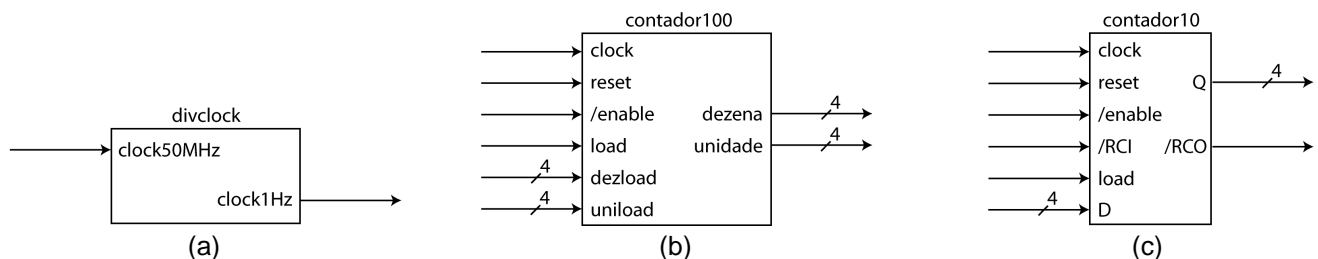


Figura 2: Caixas pretas das entidades *divclock* (a), *contador100* (b) e *contador10* (c).

2. Visto 2

O sistema a ser implementado para o visto 2 é sistema de temporização do controle de semáforos, com 'reset' síncrono, clock de 1Hz, quatro saídas binárias (a serem mostradas nos LEDs do kit de desenvolvimento), indicando que já se passaram, respectivamente, 5 segundos (T5), 6 segundos (T6), 20 segundos (T20) e 60 segundos (T60), e com o estado do contador sendo apresentado no mostrador de 7 segmentos do kit de desenvolvimento. A caixa preta e o diagrama de blocos da arquitetura interna desse sistema são apresentados na Figura 3. O código VHDL da entidade do visto 2 será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. As entidades *timeflags*, *convbinario7seg* e *mostrador* serão detalhadas a seguir, mas devem ser usadas aqui como componentes.

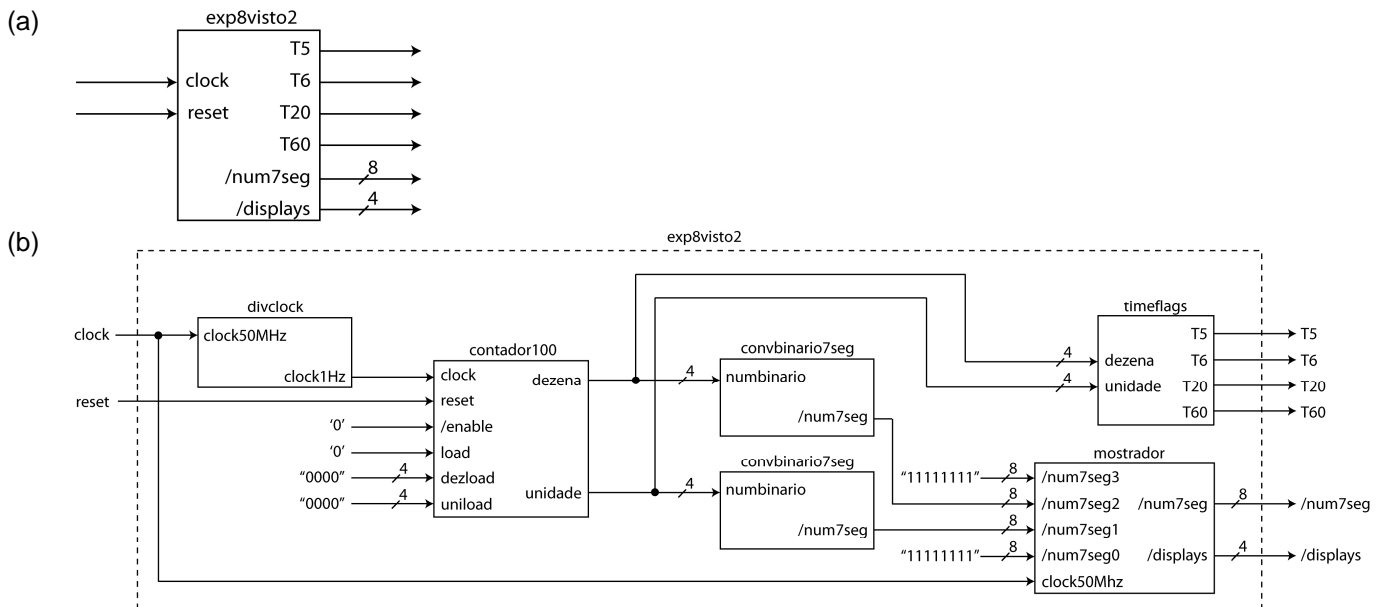


Figura 3: Caixa preta (a) e diagrama de blocos da arquitetura interna (b) do sistema a ser implementado no visto 2.

A entidade *timeflags* (Figura 4a) é responsável por verificar se já se passaram, respectivamente, 5 segundos (*T5*), 6 segundos (*T6*), 20 segundos (*T20*) e 60 segundos (*T60*). O código VHDL da entidade será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. As entradas são dois números em representação BCD, *dezena* e *unidade*, e as saídas são os quatro *flags* indicadores: *T5*, *T6*, *T20* e *T60*. Esses flags indicadores podem ser implementados usando atribuições condicionais (estrutura “when-else”) e os operadores de comparação da linguagem VHDL, conforme exemplificado a seguir. Trata-se de um circuito puramente combinacional, não sendo necessário o uso de uma estrutura “process”.

```
Ya <= '1' when (A > x"35") else -- operador "maior que"
      '0';
```

```
Yb <= '1' when (A >= x"35") else -- operador "maior ou igual"
      '0';
```

```
Yc <= '1' when (A < x"35") else -- operador "menor que"
      '0';
```

```
Yd <= '1' when (A <= x"35") else -- operador "menor ou igual"
      '0';
```

```
Ye <= '1' when (A = x"35") else -- operador "igual a"
      '0';
```

```
Yf <= '1' when (A /= x"35") else -- operador "diferente de"
      '0';
```

Nos exemplos acima, *x"35"* é a representação em hexadecimal do número binário de 8 bits “00110101”, o qual, em representação BCD, corresponde ao decimal 35, pois os quatro primeiro bits (“0011”) correspondem ao número decimal 3 e os quatro últimos (“0101”) correspondem ao número decimal 5. Você pode usar a representação hexadecimal para evitar escrever longas seqüências de bits. Por exemplo, o número *x"13F7"* corresponde ao número binário “0001001111110111”.

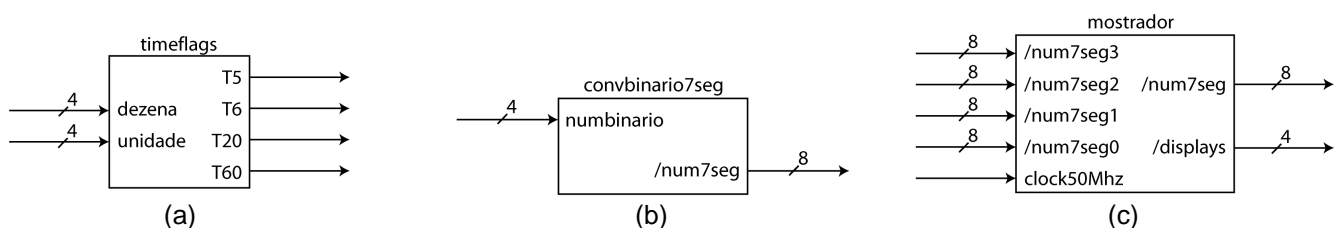


Figura 4: Caixas pretas das entidades *timeflags* (a), *convbinario7seg* (b) e *mostrador* (c).

A entidade *convbinario7seg* (Figura 4b) é responsável por converter um número binário de 4 bits em seu correspondente em representação de 7 segmentos, para que possa ser apresentado no mostrador de 7 segmentos do kit de desenvolvimento. O código VHDL da entidade e da arquitetura serão fornecidos pelo professor. Na entidade do visto 2, este conversor será usado (como componente) duas vezes, uma para converter o dígito referente à dezena e outra para converter o dígito referente à unidade. Uma explicação detalhada sobre a representação de números em código de 7 segmentos é dada no documento “Usando o mostrador de 7 segmentos da placa Basys2”.

A entidade *mostrador* (Figura 4c) é responsável por realizar a multiplexação dos pinos de entrada do mostrador de 7 segmentos, de modo a permitir mostrar até 4 dígitos diferentes simultaneamente. O código VHDL da entidade e da arquitetura serão fornecidos pelo professor. Essa entidade reduz a frequência do clock de 50 MHz para 381,47 Hz e faz com que os números (em código de 7 segmentos em lógica invertida) conectados às entradas *num7seg3*, *num7seg2*, *num7seg1* e *num7seg0* sejam mostrados, respectivamente, nos mostradores *AN3*, *AN2*, *AN1* e *AN0* do display de 7 segmentos do kit de desenvolvimento. Uma explicação detalhada sobre esse processo de multiplexação é dada no documento “Usando o mostrador de 7 segmentos da placa Basys2”.

3. Visto 3

O sistema a ser implementado para o visto 3 é um sistema de controle de semáforos com três bits de entrada — *sensorA* (indica se há carros na direção norte/sul), *sensorB* (indica se há carros na direção leste/oeste) e *ligadesliga* (chave de liga/desliga do sistema) — e saídas *num7seg* e *displays*, a serem associadas ao mostrador de 7 segmentos do kit de desenvolvimento. O mostrador deverá mostrar as luzes verde, amarela e verde de cada direção e o estado do contador, conforme ilustrado na Figura 7.

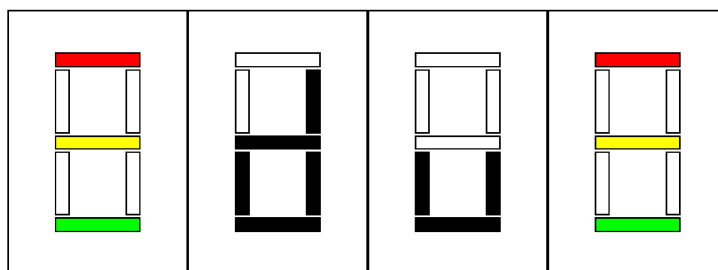


Figura 5: Codificação dos mostradores de 7 segmentos para o visto 3. Os semáforos (luz verde, amarela ou vermelha) devem ser apresentados nos mostradores *AN3* e *AN0*, enquanto que o estado do contador deve ser apresentado nos mostradores *AN2* (dezena) e *AN1* (unidade).

A caixa preta desse sistema é apresentada na Figura 6a. O código VHDL da entidade do visto 3 será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. Para este visto, serão usadas (como componentes) todas as entidades utilizadas na arquitetura interna da entidade do visto 2 (Figura 3b) — a saber: *divclock*, *contador100*, *timeflags*, *convbinario7seg* e *mostrador* — e mais duas novas entidades: *maquestados* e *convsemaforo7seg*, as quais serão detalhadas a seguir. A entrada *clock* deve ser ligada ao clock de 50 MHz do kit de desenvolvimento (pino B8), mas na arquitetura interna, deverá ser usada a entidade *divclock* para reduzir essa frequência para 1 Hz, de modo que possa ser usada nas entidades *contador100* e *maquestados*, a exemplo do que foi feito no visto 1 (Figura 1B) e no visto 2 (Figura 3b).

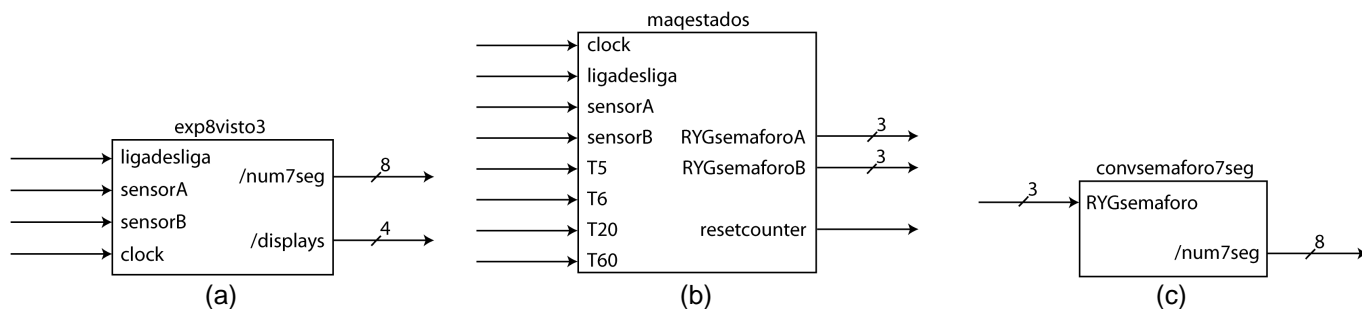


Figura 6: Caixas pretas das entidades *exp8visto3* (a), *maquestados* (b) e *convsemaforo7seg* (c).

A entidade *maquestados* é a responsável por implementar o controle dos semáforos. A caixa preta dessa entidade é apresentada na Figura 6b. O código VHDL da entidade será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. Deve ser implementada como uma máquina de estados, com duas saídas do tipo Moore — dois vetores de 3 bits *RYGsemaforoA* e *RYGsemaforoB*, que indicarão se as luzes vermelho, amarelo e verde dos semáforos das pistas norte/sul e leste/oeste, respectivamente, estarão acesas ou não (1 bit para cada cor, nessa ordem) — e uma saída do tipo Mealy — um sinal de 1 bit *resetcounter*, que indicará quando o contador de tempo deverá ser reinicializado.

As transições de estado serão controladas pelas variáveis de entrada *ligadesliga*, *sensorA*, *sensorB*, *T5*, *T6*, *T20* e *T60*. As três primeiras serão associadas às entradas de mesmo nome da entidade “top level” (*exp8visto3*) e as quatro últimas às saídas da entidade *timeflags*.

O cruzamento deverá ficar liberado para determinada direção (luz verde) e bloqueado na outra direção (luz vermelha) por 60 segundos, exceto caso haja carro esperando na direção oposta e não haja carros atravessando na direção em questão e já se tenha passado pelo menos 20 segundos. Antes de trocar a pista liberada, o semáforo deverá mostrar a luz amarela (com luz vermelha no outro semáforo) por 6 segundos e, em seguida, a luz vermelha (em ambos os semáforos) por 5 segundos. A qualquer momento, se a chave de liga/desliga for desativada, os semáforos deverão entrar em estado intermitente, no qual alternarão entre a luz amarela e nenhuma luz acesa, com intervalo de 1 segundo entre cada, devendo voltar a qualquer momento ao funcionamento normal caso a chave seja reativada.

Note que a entidade *magestados* deve, por meio da saída *resetcounter*, reinicializar o contador de tempo (implementado pela entidade *contador100*), isto é, fazer a saída *resetcounter* = ‘1’, sempre que a máquina estiver se preparando para mudar de estado, isto é, se o estado seguinte (*nextState*) for diferente do estado atual (*currentState*). Note que esta é uma saída do tipo Mealy, pois depende não só do estado atual, mas também das variáveis de entrada (que efetivamente determinam o estado seguinte). Isto pode ser implementado junto com a lógica combinacional de estado seguinte, atribuindo um valor (‘0’ ou ‘1’) a *resetcounter* sempre que um valor for atribuído à variável de estado seguinte (*nextState*). A implementação de máquinas de estado Mealy em VHDL é discutida em maiores detalhes no documento “Implementando máquinas de estados síncronas do tipo Mealy em VHDL”.

A entidade *convsemaforo7seg* (Figura 6c) será responsável por converter cada uma das saídas da máquina de estados indicadores de luzes do semáforo (*RYGsemaforoA* e *RYGsemaforoB*) em seu correspondente em representação de 7 segmentos, para que possa ser apresentado no mostrador de 7 segmentos do kit de desenvolvimento, conforme ilustrado na Figura 5. O código VHDL da entidade será fornecido pelo professor, sendo necessário escrever somente o código da arquitetura. Na entidade do visto 3, este conversor será usado (como componente) duas vezes, uma para o semáforo da direção norte/sul e outra para o da direção leste/oeste. Note que a saída dessa entidade deve utilizar lógica invertida (ativa em nível baixo), isto é, “0” indica LED aceso e “1” indica LED apagado, para que esta possa ser ligada diretamente a uma das entradas da entidade *mostrador*.