

PCS 3115 (PCS2215)

Sistemas Digitais I

Projeto: Turbo

Prof. Dr. Marcos A. Simplicio Jr.

v.2020

Enunciado

- Projetar módulo de “turbo” para os 8 botões de um controle:

- Os comandos relativos a cada botão são enviados ao console repetidamente se aquele botão for mantido pressionado.

- O projeto com Fluxo de Dados e Unidade de Controle

- Obs.: não é cobrado pelo juiz, mas vai facilitar sua vida...

- Comportamento:

- 8 bits p/ indicar os botões pressionados: **Button** = $B_7B_6B_5B_4B_3B_2B_1B_0$.



Saída:

	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
cmd (7,1,0):	1	0	0	0	0	0	1	1
cmd (nada):	0	0	0	0	0	0	0	0

- Chave de sensibilidade (**sensib**): 1, 2, 4 ou 8. Após pressionamento de um conjunto de botões e seu envio ao console, o controle **aguarda sensib bordas de clock** e começa a **repetir os botões** pressionados; antes disso, cmd = “00000000” (“nenhum botão pressionado”)

Enunciado

- Projetar módulo de “turbo” para os 8 botões de um controle:
 - Exemplos: **Button** = bits lidos durante borda de subida do clock

Sensib = 1 : in

clock: in	↑	↑	↑	↑	↑	↑	↑	↑
button: in	00000011	00000011	00000011	00000011	00000011	00100010	00100010	00100010
cmd: out	00000011	00000000	00000011	00000011	00000011	00100010	00000000	00100010

↑ outro comando

Sensib = 2 : in

clock: in	↑	↑	↑	↑	↑	↑	↑	↑
button: in	00000011	00000011	00000011	00000011	00000011	00100010	00100010	00100010
cmd: out	00000011	00000000	00000000	00000011	00000011	00100010	00000000	00000000

↑ outro comando

Sensib = 4 : in

clock: in	↑	↑	↑	↑	↑	↑	↑	↑
button: in	00000011	00000011	00000011	00000011	00000011	00100010	00100010	00100010
cmd: out	00000011	00000000	00000000	00000000	00000000	00100010	00000000	00000000

↑ outro comando

Sensib = 8 : in

...

3

Encaminhamento de solução

- Entidade fornecida:

```
entity turbo is port (  
    clock, reset: in bit; -- reset (p/ reiniciar máq. estados)  
    button: in bit_vector(7 downto 0); -- Botões  
    sensib: in bit_vector(3 downto 0); -- Sensibilidade  
    cmd: out bit_vector(7 downto 0) -- Resultado p/ console  
);  
end entity;
```

4

Encaminhamento de solução

(a) Qual seria um possível circuito para calcular um sinal de “Push”, que indica que um conjunto de botões foi pressionado?

- Logo, o módulo deve enviar algo diferente de “00000000”

5

Encaminhamento de solução

(a) Qual seria um possível circuito para calcular um sinal de “Push”, que indica que um conjunto de botões foi pressionado?

- Logo, o módulo deve enviar algo diferente de “00000000” na saída

$$\text{Push} = \text{OR}(B_i), \text{ onde } i = 0 \dots 7$$

6

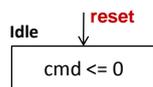
Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.

7

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



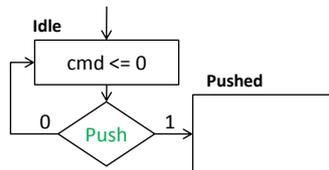
Estado inicial da UC: nenhum botão pressionado, então envio 00000000 como saída

Pergunta: Quando sair desse estado...?

8

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



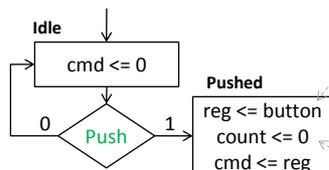
Quando algum botão for pressionado!
sinal $Push = 1$

Pergunta: O que fazer nesse estado?

9

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



1. Guardar valor de entrada:

- Permite comparação posterior, para saber se botões mudaram

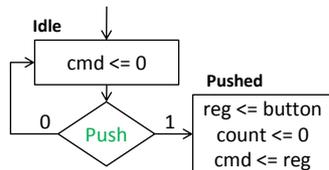
2. Iniciar um contador: necessário para tratar "sensib"

3. Enviar botões para saída cmd: botões pressionados

10

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.

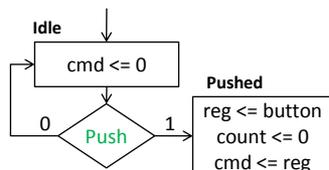


Pergunta: quando sair de estado Pushed?

11

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



Pergunta: quando sair de estado Pushed?

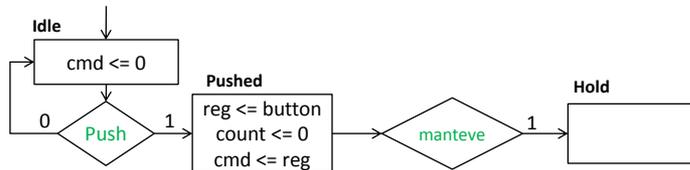
Jogador pode:

- **Manter botões** pressionados: entramos no controle de sensibilidade
- **Trocar os botões:** voltamos p/ Pushed, com a nova entrada
- **Soltar os botões:** voltamos p/ Idle

12

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



→ **Manter botões** pressionados: **que teste fazer?**

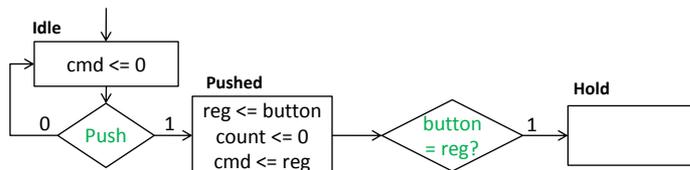
Trocar os botões: voltamos p/ Pushed, com a nova entrada

Soltar os botões: voltamos p/ Idle

13

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



→ **Manter botões** pressionados: **button = reg**

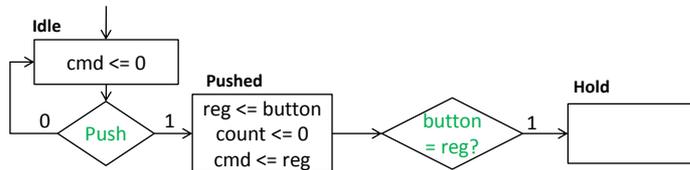
Trocar os botões: voltamos p/ Pushed, com a nova entrada

Soltar os botões: voltamos p/ Idle

14

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



Manter botões pressionados: $button = reg$

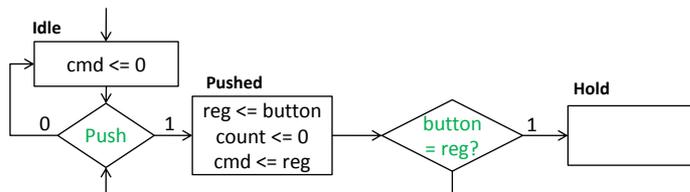
→ Trocar os botões: $que\ teste\ fazer?$

Soltar os botões: voltamos p/ Idle

15

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



Manter botões pressionados: $button = reg$

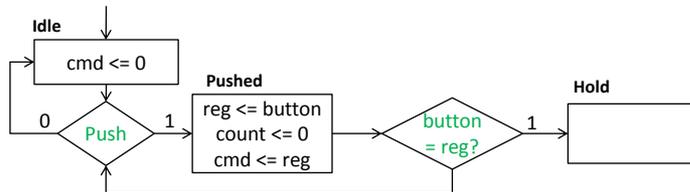
→ Trocar os botões: $button \neq reg \text{ AND } Push = 1$

Soltar os botões: voltamos p/ Idle

16

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



Manter botões pressionados: `button = reg`

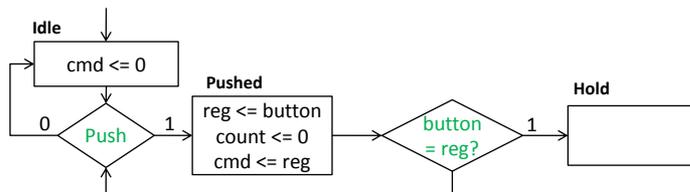
Trocar os botões: `button != reg AND Push = 1`

➔ Soltar os botões: `button != reg AND Push = 0` (já está pronto)

17

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



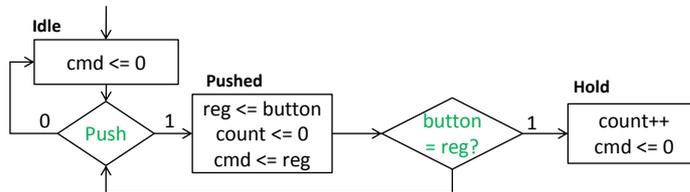
Manter botões pressionados: o que fazer no estado "Hold"?

- Controle de sensibilidade: como tratar?

18

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



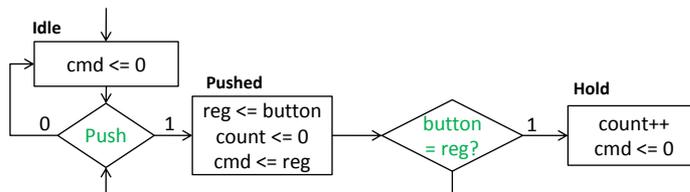
Manter botões pressionados: o que fazer no estado "Hold"?

- Iniciar uma contagem: count
- Saída deve ser 00000000

19

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.

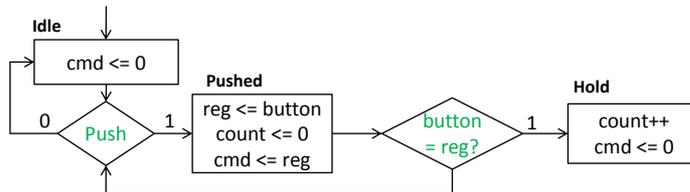


E quando **sair** do estado Hold? Saímos **pra onde**?

20

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



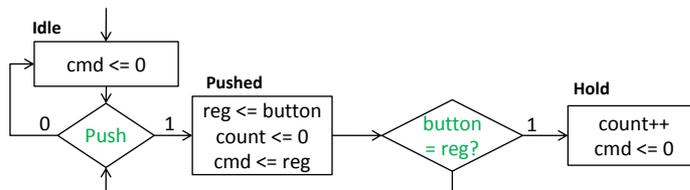
E quando **sair** do estado Hold? Saímos **pra onde**?

- Enquanto $\text{count} < \text{sensib}$: permanecer em Hold
- Depois disso, se botões mantidos pressionados: vamos para estado Turbo
- Se botão for solto, voltamos a Idle; se trocar botão, voltamos a Pushed

21

Encaminhamento de solução

(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



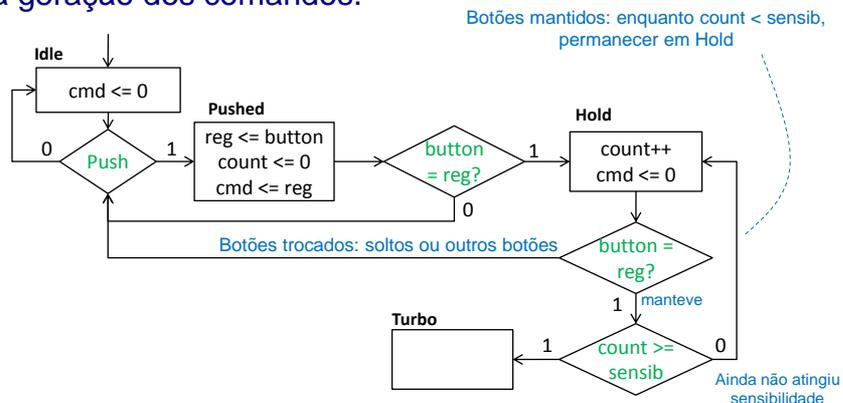
E quando **sair** do estado Hold? Saímos **pra onde**?

- Enquanto $\text{count} < \text{sensib}$: permanecer em Hold
- Depois disso, se botões mantidos pressionados: vamos para estado Turbo
- Se botão for solto, voltamos a Idle; se trocar botão, voltamos a Pushed

22

Encaminhamento de solução

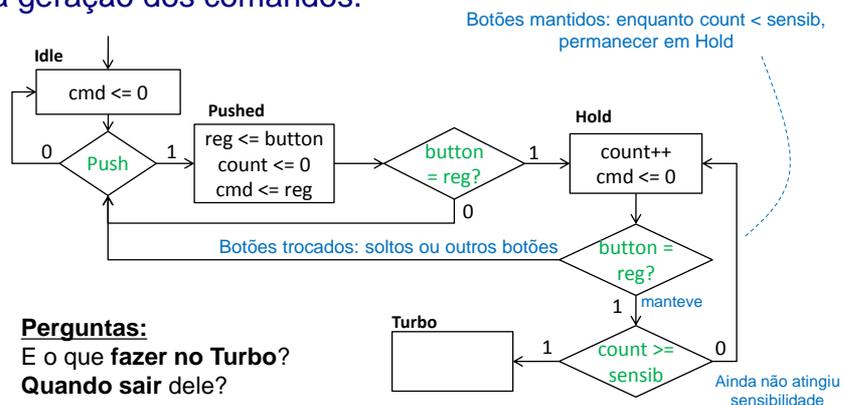
(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



23

Encaminhamento de solução

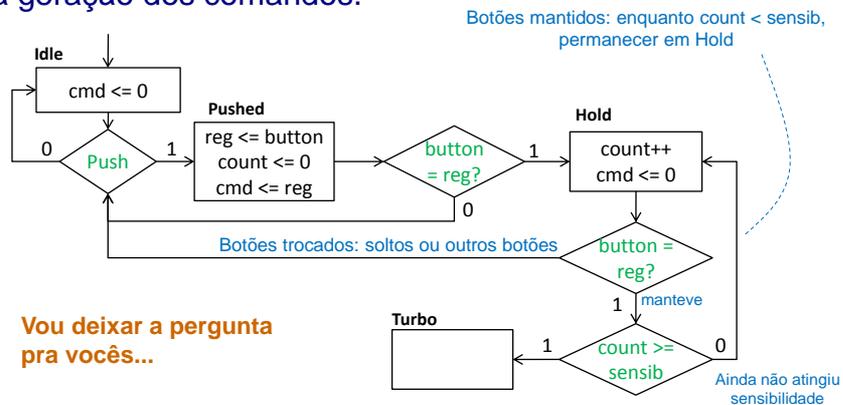
(b) Apresente o diagrama ASM de alto nível para esse circuito. Pede-se que seja adotado o modelo Moore para a geração dos comandos.



24

Encaminhamento de solução

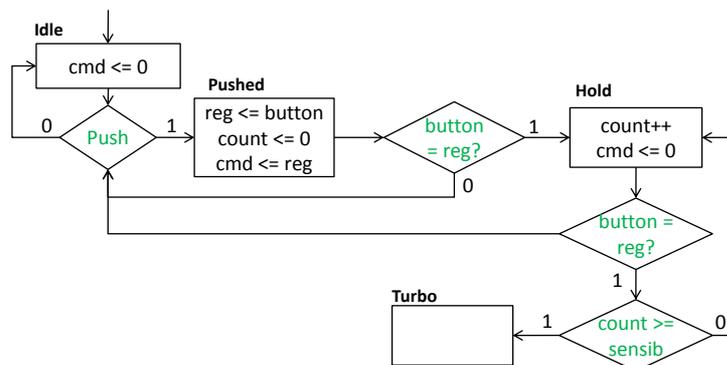
(b) Apresente o diagrama ASM de alto nível para esse circuito. Pedese que seja adotado o modelo Moore para a geração dos comandos.



25

Encaminhamento de solução

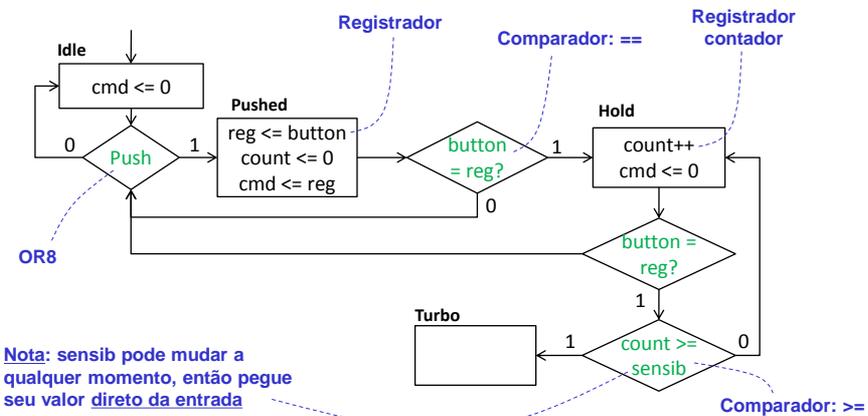
(c) Liste os elementos necessários para a construção do Fluxo de Dados



26

Encaminhamento de solução

(c) Liste os elementos necessários para a construção do Fluxo de Dados

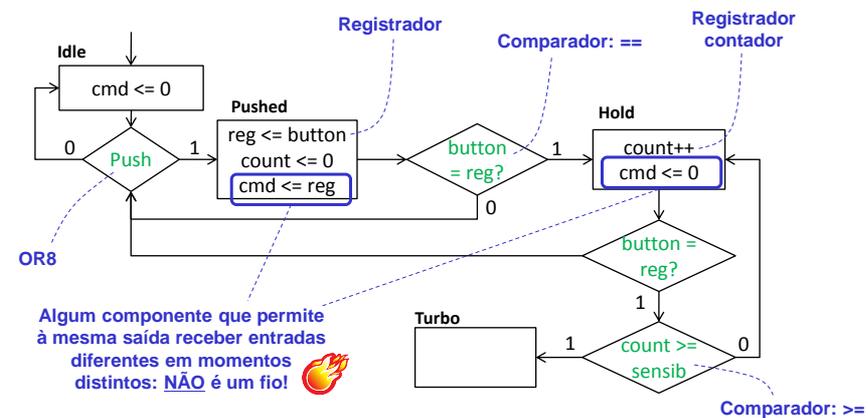


Nota: sensib pode mudar a qualquer momento, então pegue seu valor direto da entrada

27

Encaminhamento de solução

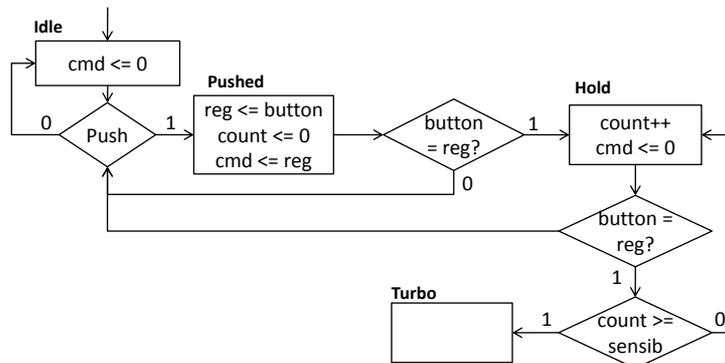
(c) Liste os elementos necessários para a construção do Fluxo de Dados e então desenhe um diagrama esquemático com a interligação destes elementos



28

Encaminhamento de solução

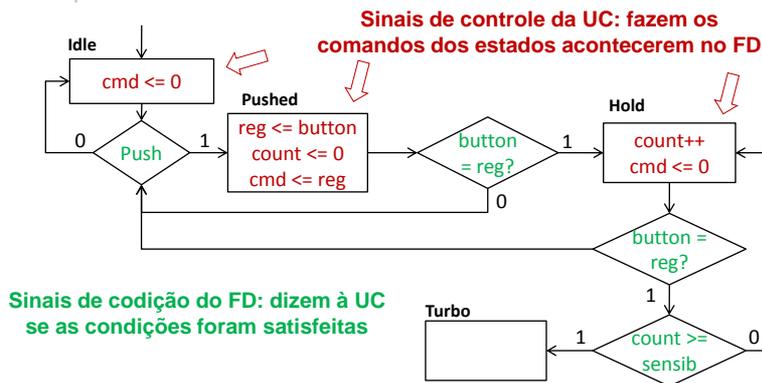
- (d) Construa o diagrama ASM da Unidade de Controle, permitindo a construção de um circuito digital síncrono. Para facilitar, assuma que o FD é ativado nas bordas de descida do clock: assim, os sinais de controle da UC podem ser habilitados na borda de subida, junto com cada transição de estados, e estarão estabilizados quando forem processados pelo FD naquele estado.



29

Encaminhamento de solução

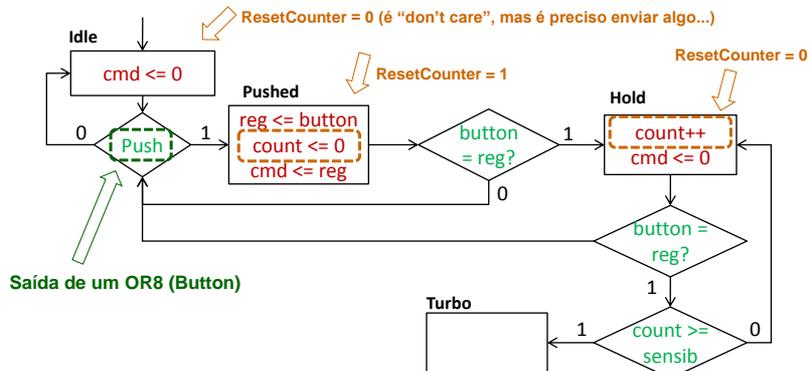
- (d) Construa o diagrama ASM da Unidade de Controle, permitindo a construção de um circuito digital síncrono. Para facilitar, assuma que o FD é ativado nas bordas de descida do clock: assim, os sinais de controle da UC podem ser habilitados na borda de subida, junto com cada transição de estados, e estarão estabilizados quando forem processados pelo FD naquele estado.



30

Encaminhamento de solução

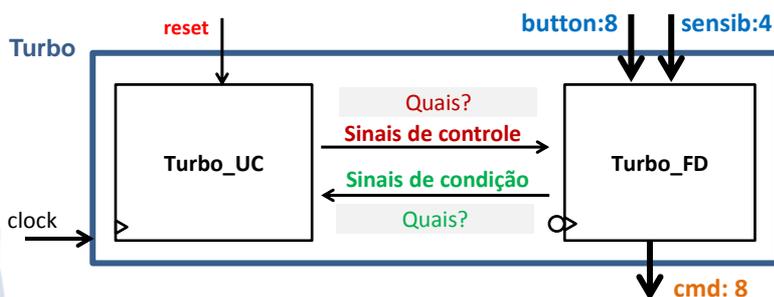
- (d) Construa o diagrama ASM da Unidade de Controle, permitindo a construção de um circuito digital síncrono. Para facilitar, assuma que o FD é ativado nas bordas de descida do clock: assim, os sinais de controle da UC podem ser habilitados na borda de subida, junto com cada transição de estados, e estarão estabilizados quando forem processados pelo FD naquele estado. **Exemplos**



31

Encaminhamento de solução

- (e) Apresente as descrições em VHDL para os seguintes elementos: Unidade de Controle (nomeado `turboUC`); o Fluxo de Dados (nomeado `turboFD`); o projeto completo conectando `turboUC` e `turboFD` (nomeado `turbo`); e o testbench.



32