	Universidade Federal de Minas Gerais
	Escola de Engenharia – Departamento de Engenharia Eletrônica
	Trabalho de Sistemas Digitais - Teoria
	Projeto e Simulação de Circuitos Digitais com Quartus II (esquemáticos)

## DESAFIO: SISTEMA DE DESTRAVAMENTO PRESENCIAL SEQUENCIAL

- Professor: Frederico Coelho  
*Trabalho originalmente proposto pelo Prof. Hermes Aguiar Magalhães*
- Trabalho em grupo de até 4 alunos (livre associação).

### OBJETIVOS:

Este trabalho tem como objetivo promover a familiarização dos alunos com:

- Metodologia de projeto de circuitos combinacionais/sequenciais e de Máquinas de Estados Finitos (FSM).
- Edição de circuitos na forma de esquemáticos (diagrama em blocos) no ambiente Quartus II – Altera.
- Simulação em computador das entradas e saídas do circuito projetado.

### MATERIAL UTILIZADO:

- Computador com softwares a seguir instalados:
  - Altera Quartus II, Ver. 11.1, Service Pack 2 – Web Edition, gratuita <sup>1</sup>, disponível em <https://www.altera.com/download/software/quartus-ii-we/11.1sp2> (mesma versão do laboratório).
  - QSim - Quartus II Simulator Tools for Education, gratuito, compatível com o usado no laboratório <sup>2</sup>,
- Arquivo “Tutorial\_quartus2\_v1-3.pdf”.

### ATIVIDADES PREPARATÓRIAS:

1. Baixar e instalar em seu microcomputador pessoal o software Quartus II na versão indicada acima <sup>3</sup>.
2. Ler atentamente e executar em seu computador os passos definidos no arquivo *Tutorial\_quartus2\_v1-3.pdf* <sup>4</sup>.

### DESCRIÇÃO DO DESAFIO:

Vocês foram solicitados a projetar a lógica digital de um sistema que destranca a porta de um cofre apenas quando todos os membros do grupo <sup>5</sup> estiverem presentes na antessala do mesmo e apertarem alguns botões na ordem correta. Para tanto, cada um de vocês deverá estar portando um *transponder*<sup>6</sup> específico fornecido pelo fabricante do cofre. Assim que cada um aperta seu botão, o respectivo *transponder* transmite por rádio em uma frequência específica um código criptografado, que na realidade é formado por uma das letras iniciais do nome de seu dono codificado em ASCII (ou de um dos sobrenomes se houver letras iguais para os componentes do grupo) indiferente se maiúscula ou minúscula – e pelo algarismo menos significativo do seu número de matrícula (codificado em BCD). Enquanto cada usuário permanece apertando o respectivo botão na parede do cofre, o circuito de decodificação dos sinais de RF do *transponder* entrega um total de (N+M) linhas digitais paralelas para o respectivo bloco “Identifica Indivíduo X” do sistema digital, conforme ilustrado na Figura 1, onde “N” é a quantidade de linhas

<sup>1</sup> Consulte a página de “Design Software” da Altera (<https://www.altera.com/download/software/quartus-ii-we>) para obter versões anteriores à versão mais atual. A versão citada é a mesma usada nos laboratórios de Sistemas Digitais.


<sup>2</sup>A partir da versão 11.1 do Quartus II Web Edition, a ferramenta de simulação QSim está incluída no pacote, mas é uma ferramenta separada. Estando no Windows, você pode abri-la usando a linha de comando “quartus\_sh -- qsim” precedida, se necessário, do caminho de instalação do Quartus, por exemplo, “c:\altera\11.1sp2\quartus\bin\quartus\_sh -- qsim”. Caso necessite baixar diretamente o QSim, ele está disponível em <http://www.altera.com/education/univ/software/qsim/unv-qsim.html> (atenção para a escolha adequada do “Filter Materials” – VHDL e versão 11.1 para o Quartus II).

<sup>3</sup> Caso não tenha condições de instalar o software em casa, você pode programar o trabalho nos computadores dos laboratórios (salas 1101 e 1102 do Bloco 1) nos horários de monitoria (sujeito a disponibilidade de vaga nas bancadas).

<sup>4</sup> Não se preocupe em atribuir pinos físicos à pinagem lógica, uma vez que você não precisará programar este trabalho em nenhum kit de desenvolvimento, mas apenas simulá-lo no QSim.

<sup>5</sup> Grupos com menos de três alunos deverão também projetar sistemas para três usuários, usando nome(s) e número(s) de matrícula de colegas fictícios diferentes de seu(s) próprio(s).

<sup>6</sup> O projeto do *transponder* não é necessário, pois foge ao escopo desta disciplina. Considere que sua saída já está disponível na forma digital.

	Universidade Federal de Minas Gerais
	Escola de Engenharia – Departamento de Engenharia Eletrônica
	Trabalho de Sistemas Digitais - Teoria
	Projeto e Simulação de Circuitos Digitais com Quartus II (esquemáticos)

necessárias para representar a letra de cada usuário e “M” é a quantidade de linhas necessárias para representar o algarismo correspondente. Os blocos combinacionais denominados “*Identifica Indivíduo*” têm a função de externar o valor lógico “1” quando o usuário é autorizado (componente do grupo) ou “0” caso contrário.

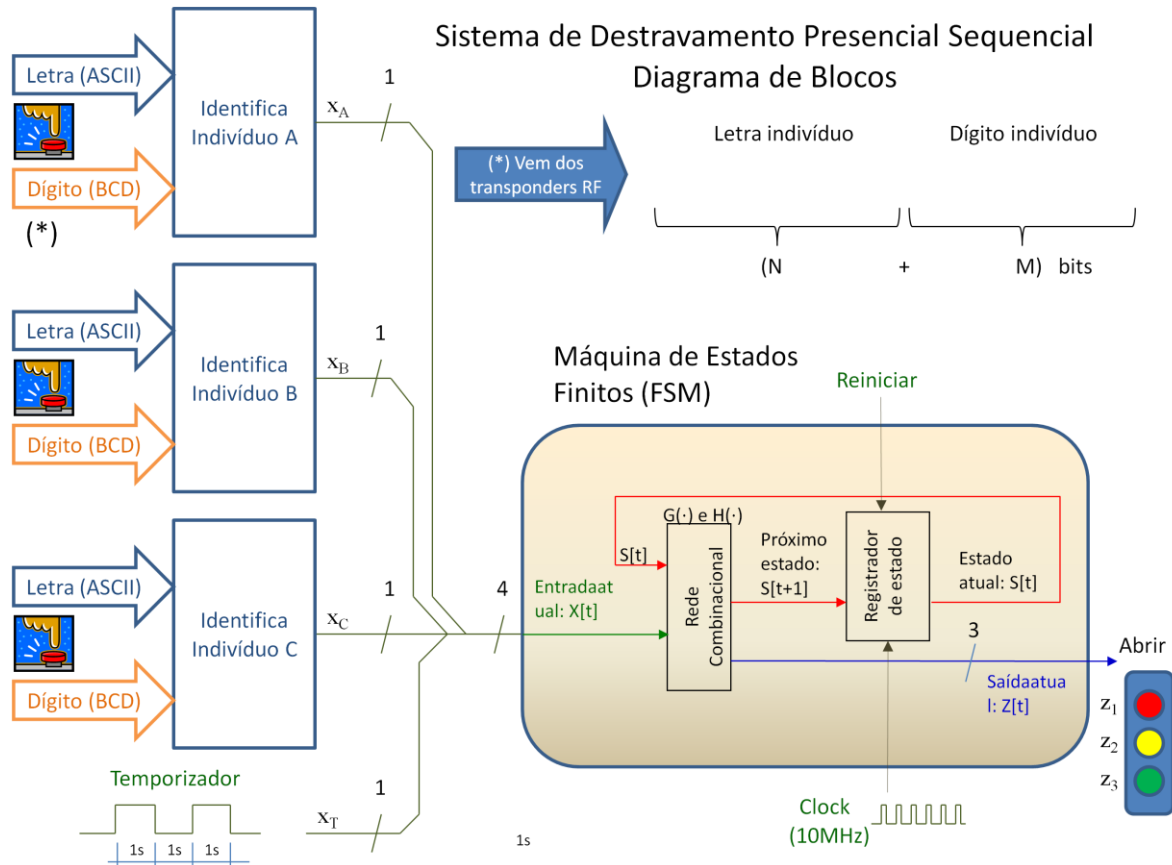



Figura 1: Diagrama de blocos do sistema de destravamento presencial sequencial

Conforme dito anteriormente, a presença dos membros do grupo na antessala do cofre não é suficiente para abri-lo. É necessário que cada membro do grupo saiba qual é o seu botão e a sua ordem de pressioná-lo relativa aos colegas, que na realidade é a ordem alfabética, seguida da ordem numérica em caso de empate. Para tanto, os três sinais lógicos de identificação dos indivíduos devem ir para o nível lógico “1” em uma sequência específica, o que requer do sistema a implementação de um circuito sequencial. Uma forma padrão para a construção de qualquer rede sequencial é a implementação canônica (ou implementação de Huffman-Moore), que baseia-se na descrição dos estados de um sistema:

$$\text{Função de transição de estado: } S[t+1] = G(S[t], X[t])$$

$$\text{Função de saída: } Z[t] = H(S[t], X[t])$$

Esta implementação de uma Máquina de Estados Finitos (FSM) está ilustrada na Figura 1, e consiste em um registrador de estado e em uma rede combinacional que implementa as funções lógicas de transição de estado  $G(\cdot)$  e de saída  $H(\cdot)$ . Como se trata de um sistema síncrono, o sinal de *clock* determina o momento em que o novo estado é carregado no registrador. A cada pulso periódico do *clock*, que aqui ocorre a cada 100ns ( $f_{\text{clock}} = 10\text{MHz}$ ), o “próximo estado” é carregado no registrador de estado e permanece armazenado lá até a ocorrência de novo ciclo do *clock*, quando é feita nova atualização e assim sucessivamente. Podemos entregar então as linhas digitais de

	Universidade Federal de Minas Gerais
	Escola de Engenharia – Departamento de Engenharia Eletrônica
	Trabalho de Sistemas Digitais - Teoria
	Projeto e Simulação de Circuitos Digitais com Quartus II (esquemáticos)

identificação positiva de cada indivíduo como entrada (sinais  $X[t]$ ) para a FSM. Esta, por sua vez, deverá ter uma arquitetura e saídas tais que permitam ou neguem a abertura do cofre, conforme descrito a seguir.

Cada membro do grupo deverá pressionar seu botão e mantê-lo pressionado após a ação do colega anterior até que o cofre se abra. A indicação de que o próximo botão pode ser pressionado é dada por um conjunto de três LEDs similares a um semáforo, que são acionados por linhas digitais de saída da FSM (sinais  $Z[t]$ ). As luzes estão inicialmente apagadas. O pressionar do botão correto pela primeira pessoa autorizada acende a luz vermelha (sinal  $z_1$ ). O pressionar do segundo botão na sequência correta por pessoa autorizada acende a luz amarela (sinal  $z_2$ ). O pressionar do terceiro botão na sequência correta por pessoa autorizada destrava o cofre e acende a luz verde (sinal  $z_3$ ) por no mínimo 3 segundos (se o grupo tiver mais de três pessoas deve-se aumentar o número de leds para o mesmo número de integrantes). Para tanto a arquitetura da FSM conta com uma entrada temporizada adicional  $x_T$  (veja a Figura 1) que consiste de uma onda quadrada de período 2 segundos. Após este tempo, as luzes voltam a se apagar e o comando de abertura do cofre é retirado. Se durante o procedimento qualquer usuário soltar o respectivo botão, o *transponder* para de transmitir e as linhas digitais de entrada daquele indivíduo vão a zero, o que faz a saída do bloco “Identifica Indivíduo” correspondente (sinal  $x_A$ ,  $x_B$  ou  $x_C$ ) também igual a zero, invalidando todo o processo. Neste caso todas as luzes se apagam e o processo volta ao estado inicial. Se algum usuário errar a sequência ou não for autorizado, o processo também volta ao estado inicial com as luzes apagadas.


TABELA 1: ASCII – AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>;</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

Source: [www.LookupTables.com](http://www.LookupTables.com)

## Lógica Combinacional

Os algarismos menos significativos de seu número de matrícula podem variar de 0 a 9 e estarão codificados em BCD (*binary coded decimal*) o que define o valor de M. Quanto às letras, como é indiferente que sejam maiúsculas ou minúsculas, o sistema deverá funcionar para ambas. Elas estão codificadas em ASCII (Tabela 1, valores em

	Universidade Federal de Minas Gerais
	Escola de Engenharia – Departamento de Engenharia Eletrônica
	Trabalho de Sistemas Digitais - Teoria
	Projeto e Simulação de Circuitos Digitais com Quartus II (esquemáticos)

decimal, hexadecimal e octal). Diante de tais premissas, examine a Tabela ASCII e procure por simplificações que levem a valores mínimos de N e projete os blocos “Identifica Indivíduo”, com codificação “*hardwired*”<sup>7</sup>. A critério do aluno, sugere-se o estudo dos blocos construtivos do Capítulo 4 do livro texto (Sistemas Digitais, Frank Vahid) e o uso da metodologia de projeto de circuitos combinacionais vista em sala resumida na Tabela 2.

**TABELA 2: METODOLOGIA PARA PROJETO DE CIRCUITOS COMBINACIONAIS**

Passo 1	Identifique a função	Crie a tabela verdade ou equações, <b><i>o que for mais natural para o problema em questão</i></b> , para descrever o comportamento do circuito lógico combinacional.
Passo 2	Converta para equações	Esse passo só é necessário se você no passo 1 descreveu o problema em uma tabela verdade. Crie uma equação para cada saída, fazendo a soma de produtos, por exemplo, ou outro método que julgar mais adequado. Simplifique as equações se necessário, usando álgebra booleana ou mapas de Karnaugh, etc.
Passo 3	Implemente as equações na forma de circuitos com portas lógicas	Para cada saída (equação), crie o circuito correspondente usando portas lógicas. Compartilhar portas entre diferentes saídas é uma opção possível.


### Lógica Sequencial

Use obrigatoriamente e documente a **Metodologia em 5 Passos** vista em sala (Tabela 3) para projetar a lógica sequencial necessária. O projeto da FSM se traduz então em:

**TABELA 3: METODOLOGIA PARA PROJETO DE CIRCUITOS SEQUENCIAIS (FSM)**

Passo 1	Capture o comportamento da FSM	Desenhe o diagrama da FSM que descreva seu comportamento. Lembre-se de certificar-se que apenas uma condição de transição de estado seja verdadeira e que haja pelo menos uma condição verdadeira para cada estado a cada ciclo de <i>clock</i> .
Passo 2	Crie a arquitetura.	Neste caso é requerida a arquitetura Canônica (arquitetura de Huffman-Moore) ilustrada na Figura 1.
Passo 3	Codifique os estados.	Diferentes escolhas de codificação de estados levarão a diferentes lógicas combinacionais no passo seguinte, com maior ou menor complexidade na implementação, dependendo de suas escolhas.
Passo 4	Crie a tabela de transição de estados $G(\cdot)$ e de saída $H(\cdot)$	Crie tabelas-verdade para a lógica combinacional de modo tal que a lógica irá gerar as <u>saídas <math>Z[t]</math></u> e os sinais de <u>próximo estado <math>S[t+1]</math></u> (lado direito da tabela verdade) a partir do <u>estado atual <math>S[t]</math></u> e das <u>entradas <math>X[t]</math></u> (lado esquerdo da tabela verdade). Estas tabelas são a base para o projeto da lógica combinacional – funções $G(\cdot)$ e $H(\cdot)$ . Na construção desta rede combinacional, aplique métodos de simplificação (mapas de Karnaugh, álgebra booleana, etc.) se necessário.
Passo 5	Implemente a lógica combinacional	Implemente a lógica combinacional e ligue-a às entradas, saídas e ao registrador (grupo de flip-flops) usando qualquer método apropriado para tal função. Neste trabalho o aluno deverá usar diagramas esquemáticos e os componentes pré-existent na biblioteca do Quartus II (portas lógicas e Flip-flops tipo D).

<sup>7</sup> Codificação “*hardwired*”: o que precisar ser ligado permanentemente em nível lógico “zero”, ligue ao terra (= Gnd ou ground). O que necessitar ser ligado permanentemente em nível lógico “1”, ligue na tensão de alimentação ou  $V_{cc}$ .

	Universidade Federal de Minas Gerais
	Escola de Engenharia – Departamento de Engenharia Eletrônica
	Trabalho de Sistemas Digitais - Teoria
	Projeto e Simulação de Circuitos Digitais com Quartus II (esquemáticos)

## ENTREGAS:

Documento eletrônico textual postado no Moodle (.DOC, .DOCX ou .PDF) contendo:

1. Definição justificada do tamanho em bits das entradas de identificação dos indivíduos, N e M os menores possível, mas que atenda a todo o alfabeto.
2. Diagrama de captura<sup>8</sup> da Máquina de Estados Finitos (estados, transições, entradas e saídas).
3. Codificação dos estados
4. Expressões lógicas e/ou tabelas verdades necessárias ao projeto da FSM: as funções lógicas de transição de estado  $G(\cdot)$  e de saída  $H(\cdot)$ , com as simplificações efetuadas devidamente documentadas.
5. Dispositivo (chip) escolhido com justificativa que relacione a capacidade do chip à quantidade de macrocélulas (ou recursos internos equivalentes) demandadas por seu projeto.
6. Documentação esquemática do projeto lógico no Quartus II contendo:
  - a. Pinos lógicos de entrada e de saída devidamente identificados usando os literais mostrados no diagrama da Figura 1.
  - b. Projeto do circuito de identificação dos indivíduos usando lógica *hardwired* para identificar os membros do grupo obedecendo às regras estabelecidas. Informar aqui qual inicial do nome foi usada por cada um.
  - c. Projeto lógico das Máquinas de Estados Finitos (FSM), contendo rede combinacional e registradores construídos com flip-flops tipo D.<sup>9</sup>
  - d. Interligação entre as partes componentes do projeto.
    - i. O trabalho foi pedido na forma de esquemáticos. A implementação em VHDL é opcional.
    - ii. Caso o grupo opte por programar em VHDL, deverá apresentar um esquemático geral com a interligação dos módulos VHDL.
7. Documentação da simulação (diagrama de temporização) do circuito para pelo menos um caso de sucesso e um caso de fracasso na abertura do cofre, mostrando entradas e saídas. A simulação deve conter “balões” numerados indicativos com setas que auxiliem o acompanhamento dos comentários. Neste caso, o tempo na escala de segundos é muito longo para o simulador. Portanto simule em escala reduzida de tempo (por exemplo, temporizador com período em  $\mu s$  ao invés de segundos).
8. Além do documento textual descrevendo o projeto e as simulações, o aluno deve entregar os arquivos de projeto para eventuais conferências e compilação pelo professor, zipando o diretório do projeto, contendo também as simulações, e preservando também as informações de subdiretórios.

---

<sup>8</sup> Diagrama de captura da FSM: trata-se do diagrama que o grupo usou para conceber o projeto e não aquele gerado automaticamente pelo software. A critério do grupo, pode-se usar qualquer ferramenta para documentar este diagrama, mas recomenda-se o uso de ferramentas UML – Unified Modeling Language. A UML tem em seu conjunto de diagramas comportamentais (*behavioral diagrams*) o diagrama de estados (*Statechart Diagram* ou *State Machine Diagram*). Por exemplo, na iniciativa de código livre temos a ferramenta StarUML: <http://staruml.sourceforge.net>. Para projetistas de sistemas embarcados, temos a ferramenta de modelagem da Quantum Leaps: <http://www.state-machine.com/qm/index.php>. Visite [http://en.wikipedia.org/wiki/List\\_of\\_UML\\_tools](http://en.wikipedia.org/wiki/List_of_UML_tools) para uma comparação entre ferramentas UML.

<sup>9</sup> Flip-flops tipo D estão disponíveis na biblioteca do Quartus sob o nome “dff”.