

ARQ1 _ Guia_17

Tema: Projeto de circuitos sequenciais

Orientação geral:

Apresentar todas as soluções em apenas um arquivo com formato texto (.txt).
Sugere-se usar como nome Guia_xx.txt, onde xx indicará o guia, exemplo Guia_01.txt.
Todos os arquivos deverão conter identificações iniciais com o nome e matrícula, no caso de programas, usar comentários.
As implementações e testes dos exemplos em Verilog (.v) fornecidos como pontos de partida, também fazem parte da atividade e deverão ter os códigos fontes entregues **separadamente**, com o código fonte, a fim de que possam ser compilados e testados. Entregar os módulos de testes.
Sugere-se usar como nomes Guia_01yy.v, onde yy indicará a questão, exemplo Guia_0101.v
As saídas de resultados, opcionalmente, poderão ser copiadas ao final do código, em comentários.
Quaisquer outras anotações, observações ou comentários poderão ser colocadas em arquivo texto (README.txt) acompanhando a entrega.

Outras formas de solução serão **opcionais**; não servirão para substituir as atividades a serem avaliadas. Caso entregues, poderão contar apenas como atividades extras.
Os programas com funções desenvolvidas em C, Java ou Python (c, .java, py), como os modelos usados para verificação automática de testes das respostas; caso entregues, também deverão estar em arquivos **separados**, com o código fonte, a fim de serem compilados e testados.
As execuções deverão, preferencialmente, serão testadas mediante uso de redirecionamento de entradas e saídas padrões, cujos dados/resultados deverão ser armazenados em arquivos textos.
Os resultados poderão ser anexados ao código, ao final, como comentários
Planilhas, caso venham a ser solicitadas, deverão ser **programadas** e/ou usar funções nativas. Serão suplementares e opcionais, e deverão ser entregues em formato texto, preferencialmente, com colunas separadas por tabulações ou no formato (.csv), acompanhando a solução em texto.
Arquivos em formato (.pdf), fotos, cópias de tela ou soluções manuscritas também poderão ser aceitos como recursos suplementares para visualização, mas não servirão como substitutos e **não** terão validade para fins de avaliação.
Os *layouts* de circuitos deverão ser entregues no formato (.circ), identificados internamente.
Os *layouts* de diagramas deverão ser entregues no formato (.jff), identificados externamente.
Figuras exportadas pela ferramenta serão aceitas apenas como arquivos para visualização, mas **não** terão validade para fins de avaliação. Separar versões completas (a) e simplificadas (b).

Atividade: Arquitetura de Computador – Projeto de Máquina de Estados Finitos

Implementar a máquina descrita pela tabela abaixo, após completar com as transições em falta:
Todos os modelos deverão ser testados em simulador (Logisim).

atual				próximo		D		SR		JK		T	
m	t	t	t	a	b	FF	FF	FF	FF	FF	FF	FF	FF
0	a	b	c	t+1	t+1	Da	Db	SaRa	SbRb	JaKa	JbKb	Ta	Tb
0	0	0	0	0	0								
1	0	0	1	0	1								
2	0	1	0	0	1								
3	0	1	1	1	1								
4	1	0	0	1	0								
5	1	0	1	0	0								
6	1	1	0	1	1								
7	1	1	1	1	0								

01.) com *flip-flops* D

02.) com *flip-flops* JK em modo SR

03.) com *flip-flops* JK em modo *toggle*

04.) Montar um contador assíncrono com 3 *flip-flops* T em série, com $T1=T2=T3=1$, $T1_clk=clk$ $T2_clk=Q1$ $T3_clk=Q2$ e as saídas na ordem {Q3,Q2,Q1}.

05.) Montar um contador assíncrono com 3 *flip-flops* T em série, com $T1=T2=T3=1$, $T1_clk=clk$ $T2_clk=Q1$ not $T3_clk=Q2$ not e as saídas na ordem {Q3,Q2,Q1}.

Extras

06.) Montar no JFLAP uma máquina de Turing para reconhecer a sequência 1001, com interseção.

07.) Montar no JFLAP uma máquina de Turing para reconhecer a sequência 101, sem interseção.