

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELETRÔNICA

PROJETO RTL: TRABALHO FINAL
LABORATÓRIO DE SISTEMAS DIGITAIS

Felipe Freitas
Sergio Bicalho

Belo Horizonte, setembro de 2020

INTRODUÇÃO

O seguinte relatório tem como objetivo descrever o sistema de controle de acessos a um estacionamento, mostrar o seu diagrama conceitual de Máquina de Estados de Alto Nível, o desenho do Caminho de Dados.

DESCRIÇÃO DO SISTEMA

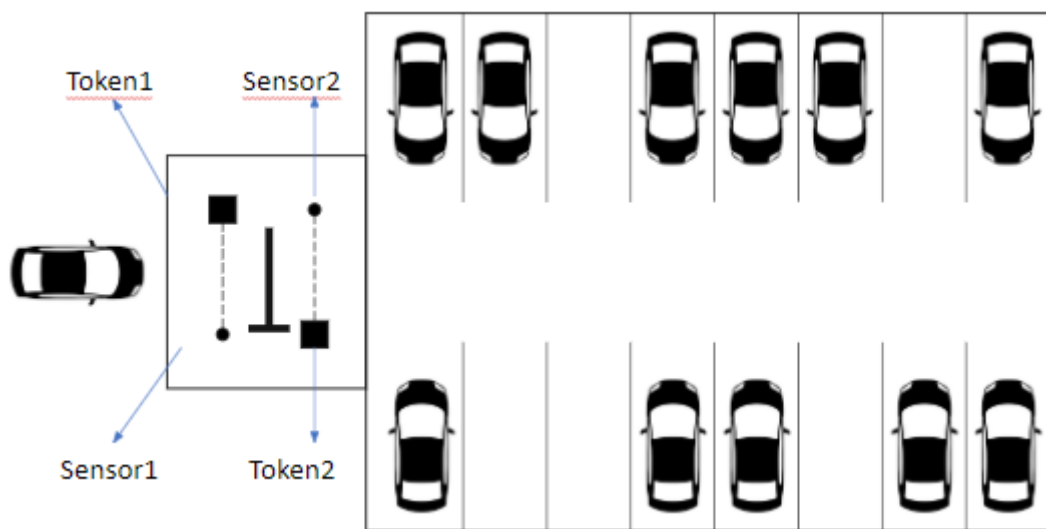


Figura 1: Sistema de Controle de Acessos a um estacionamento com 16 vagas.

Em um estacionamento privado, o controle de entrada e saída é realizado por uma única via. O sistema deve identificar automaticamente o veículo e sua intenção, seja de ingressar ou sair do estabelecimento.

Ao se posicionar em frente ao sensor1 o sistema deverá imprimir um ticket, que possui um código baseado na placa do veículo, que deverá ser obrigatoriamente retirado pelo motorista e só assim a cancela deverá ser liberada e será fechada somente após o veículo passar completamente pelo sensor2. Caso o estacionamento esteja em sua capacidade máxima de lotação o sistema não poderá permitir o acesso de outros veículos ao estabelecimento.

Caso a intenção do motorista seja sair do estacionamento ele deverá colocar um ticket validado para liberar a cancela, após o veículo passar completamente pelo sensor1 a cancela deverá ser fechada.

Máquina de Estados de Alto Nivel

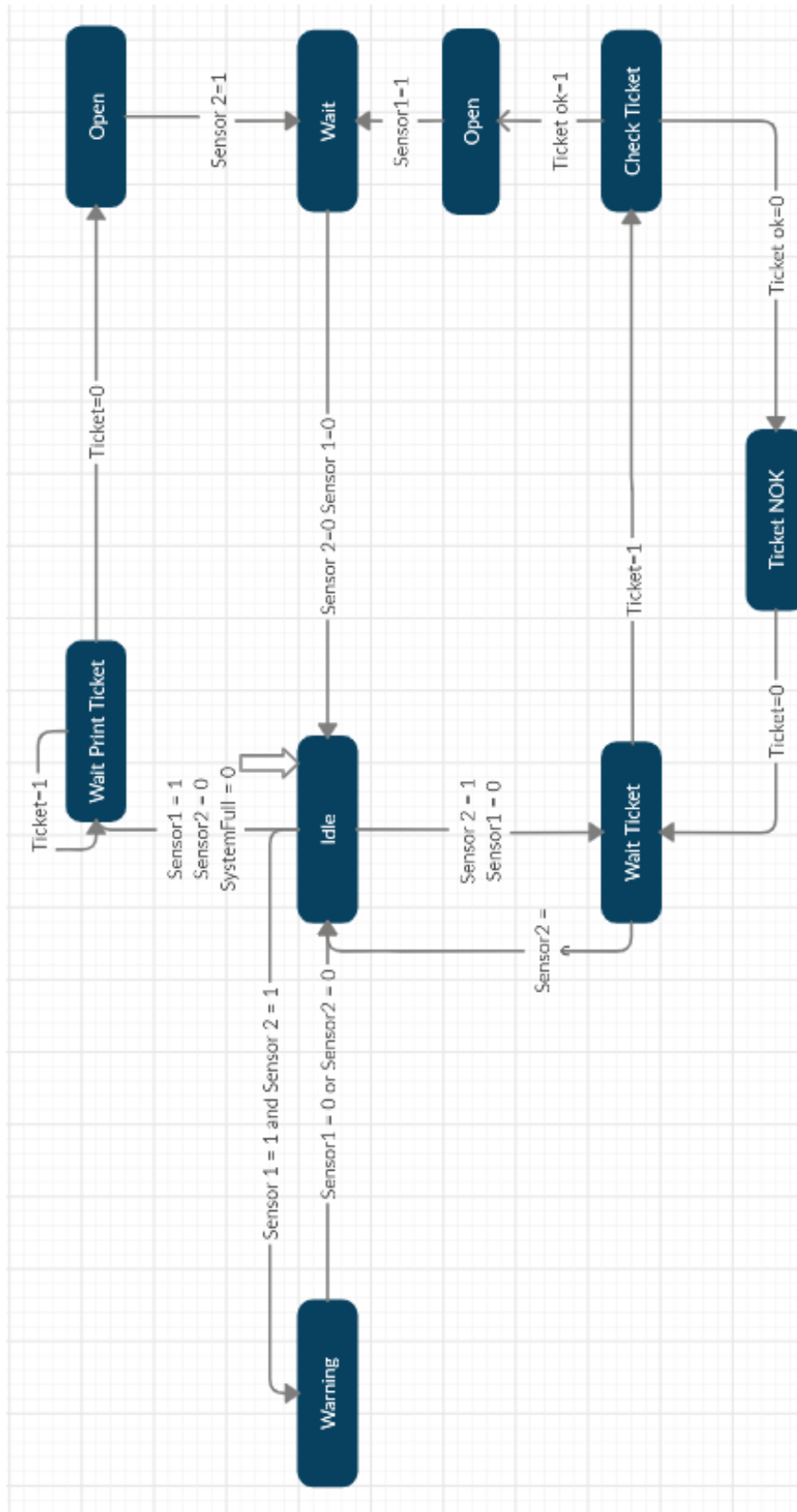
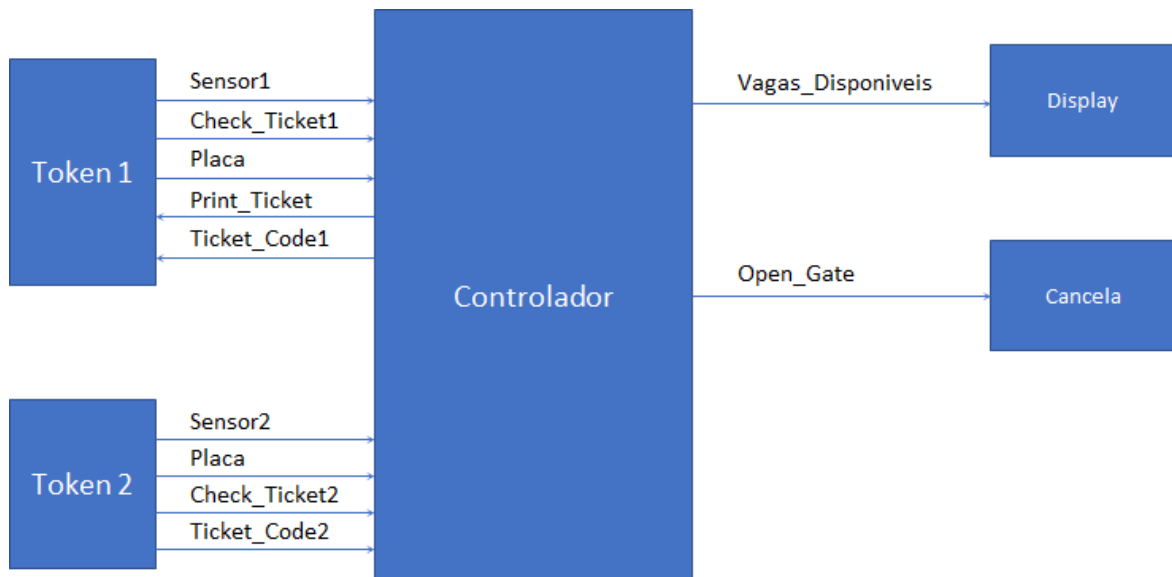


Diagrama de blocos



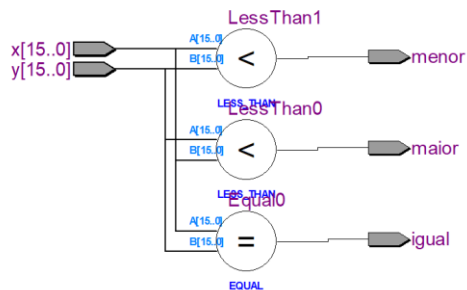
Entradas e Saídas

Entradas	Sensor1	1 bit
	Sensor2	1 bit
	Check_Ticket1	1 bit
	Check_Ticket2	1 bit
	Placa_Read1	16 bits
	Placa_Read2	16 bits
	Ticket_Code1	16 bits

Saídas	Open_Gate	1 bit
	Ticket_Code2	16 bits
	Vagas_disponiveis	4 bits
	Mensagem	16 bits

Componentes do Caminho de dados

Comparador



entity comparador is

port

(

x : in std_logic_vector (15 downto 0);

y : in std_logic_vector (15 downto 0);

igual, maior, menor : out std_logic

);

end entity;

architecture rtl of comparador is

begin

igual <= '1' when x = y else '0' ;

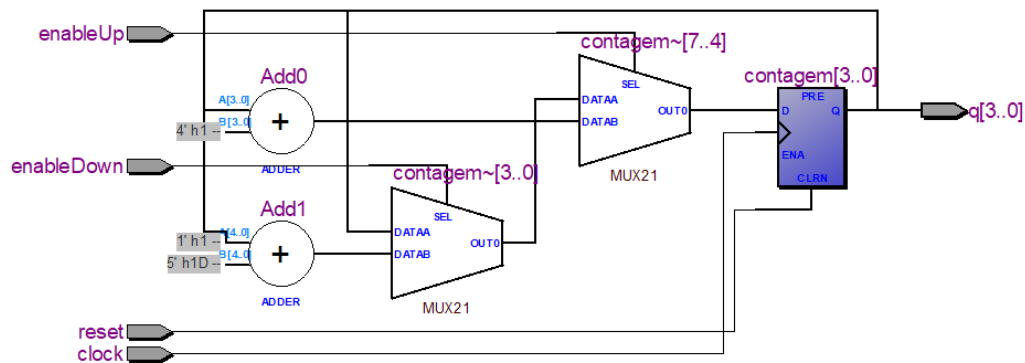
maior <= '1' when x > y else '0' ;

menor <= '1' when x < y else '0' ;

end rtl;

Msgs					
/tb_comparador/A	-No Data-	52428	13107	8738	
/tb_comparador/B	-No Data-	0	17476	13107	65535
/tb_comparador/big...	-No Data-				
/tb_comparador/sm...	-No Data-				
/tb_comparador/equal	-No Data-				

Contador



entity contador is

port (

clock, reset, enableUp, enableDown : in std_logic;
q : out std_logic_vector (3 downto 0));

end contador;

architecture arch of contador is

begin

process (clock, reset)

variable contagem : integer range 0 to 15;

begin

if reset = '1' then

contagem := 0;

elsif clock'event and clock='1' then

if enableUp='1' then

contagem := contagem+1;

elsif enableDown='1' then

contagem := contagem-1;

end if;

end if;

q <= conv_std_logic_vector(contagem, 4);

end process;

end arch;

