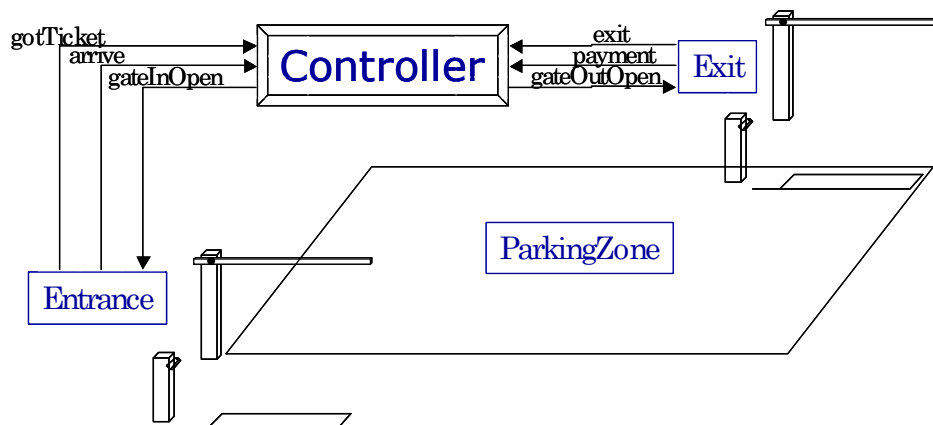


Laboratório 4 - Mini-parque de estacionamento e modelação com Redes de Petri

Objetivo:

Pretende-se realizar o controlador de um mini-parque de estacionamento com capacidade para três lugares. O parque possui uma zona de entrada, uma zona de saída e a zona de estacionamento propriamente dita.



A zona de entrada dispõe de um sensor *arrive* detetando a entrada de um novo carro na zona de receção. Após o condutor recolher o bilhete na entrada, o controlador recebe um sinal *gotTicket*, após o que deverá levantar a cancela atuando a saída *gateInOpen*, caso existam lugares vagos, até que o carro entre completamente (isto é, até que o sinal *arrive* fique não ativo).

A zona de saída dispõe de um sensor *exit* detetando a chegada de um novo carro à zona de saída. Após pagamento, o controlador recebe um sinal *payment*, após o que deverá levantar a cancela atuando a saída *gateOutOpen*, até que o carro permaneça na zona de saída.

Admite-se que os carros não fazem marcha atrás e que a paciência do condutor é infinita (pode esperar por um lugar indefinidamente).

A solução a encontrar deverá ser implementada utilizando o ambiente de desenvolvimento ISE da Xilinx, tendo o kit distribuído contendo a FPGA Spartan3 como plataforma para implementação.

Plano de execução:

Parte I – Modelação através de RdP

Especifique o controlador descrito através de uma rede de Petri (RdP) lugar-transição.

Parte II – Análise do modelo RdP

Recorrendo aos recursos disponibilizados pelas IOPT-Tools no site <http://gres.uninova.pt/IOPT-Tools>, proceda à edição do modelo RdP obtido e obtenha o seu espaço de estados, de forma a determinar as marcações potencialmente observáveis nos vários lugares do modelo. Identifique (pelo menos) uma sequência (de interesse) de eventos ou sinais de entrada (*arrive*, *gotTicket*, *exit* e *payment*) e caracterize o comportamento do modelo em face dessa sequência de eventos.

Faça uso das capacidades de simulação do ambiente, bem como da possibilidade de interligação à ferramenta HIPPO para a obtenção de invariantes e matriz de incidência.

Parte III – Especificação da implementação

Tendo em conta as propriedades do modelo, especifique uma implementação baseada na representação individual de cada nó da RdP (isto é, lugares por flip-flops ou contadores e transições representadas por lógica combinatória).

Codifique o sistema em VHDL e edite a solução encontrada.

Parte IV – Utilização de gerador automático de código

Utilize a capacidade do ambiente IOPT-Tools para geração automática de código VHDL.

Compare o código gerado automaticamente com o que produziu manualmente. Tire conclusões.

Parte V – Simulação da implementação

Recorrendo às facilidades de simulação do ISE proceda à validação do funcionamento do sistema, recorrendo à sequência de eventos ou sinais utilizada na parte II, no sentido de confirmar que os circuitos encontrados nas partes III e IV satisfazem o enunciado.

Parte VI – Teste e verificação

Utilizando o kit distribuído com a FPGA Spartan 3, verifique que o funcionamento do circuito integrado configurado está de acordo com o pretendido.