



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE NOVA DE LISBOA

**UNIVERSIDADE NOVA DE LISBOA**

Faculdade de Ciências e Tecnologia

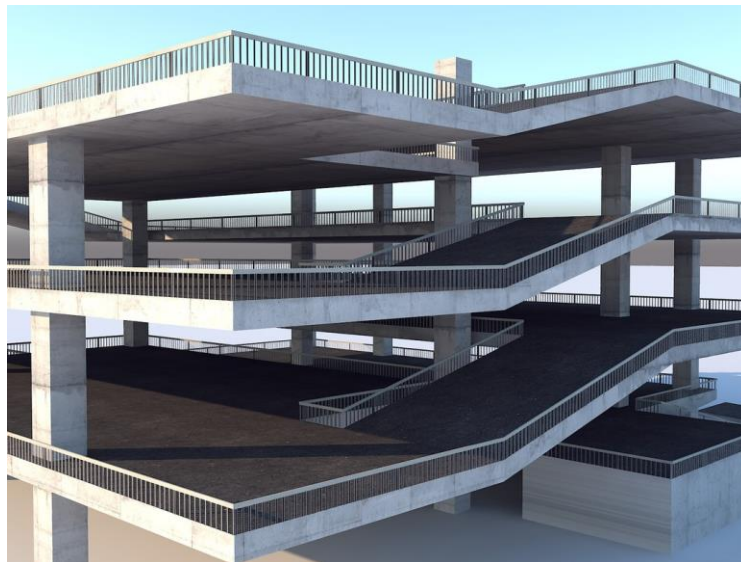
Conceção de Sistemas Digitais

## ***Sistema de controlo de um Parque de Estacionamento -Conceção de Sistemas Digitais***

## ***Sistema de controlo de um Parque de Estacionamento -Conceção de Sistemas Digitais***

## **2º Trabalho Laboratorial**

# **Sistema de Controlo de um Parque de Estacionamento**



P4

André Ferreira 54436  
Eduardo Dias 53579  
Filipe Marques 52755

## Contents

1. Introdução .....	6
2. Descrição do sistema .....	6
3. Controlador do parque .....	7
3.1 PetriNet .....	7
3.2 Entradas no parque.....	8
3.3 Saídas do parque .....	9
3.4 Rampas .....	10
.....	10
3.5 Lotação.....	11
3.5.1 Primeiro piso.....	11
3.6 Controlador do incrementador .....	13
.....	13
4 Dados.....	14
4.1 Incrementador .....	14
4.2 Interface .....	14
4.3 Relógio .....	15
4.4 Conv_display .....	15
4.5 Functions .....	16
5 Implementação FPGA.....	17
5.1 Modo parque.....	17
5.2 Modo relógio.....	19
6 Teste .....	20
7 Conclusão .....	22

## **1. Introdução**

Pretende-se a implementação de um controlador para um parque de estacionamento cujas características irão ser definidas pelos implementadores. Este, por sua vez, terá associado um relógio digital, que irá ser uma adaptação do relógio digital desenvolvido para o primeiro trabalho, que irá permitir estabelecer a hora de entrada e/ou de saída do parque e outras características adicionais associadas a este tendo como recurso 4-digital display.

Este documento tem como objetivo o relato dos requisitos necessários para a especificação geral do projeto e divide-se em 6 capítulos (sem contar com a introdução) que inclui, a descrição geral do sistema bem como as especificações consideradas para a sua implementação, o controlo onde é descrita a petri net bem como o controlador do relógio utilizado, os dados onde é especificada a forma com que os valores de minutos totais no parque por dia foram calculados e guardados, simulação onde o grupo testa o funcionamento da rede de petri e do relógio antes de passar para a FPGA, a FPGA onde é explicado como é feita a atribuição dos botões aos sensores do parque e finalmente a conclusão onde é feita uma retrospectiva ao projeto enumerando as dificuldades e os objetivos concluídos com sucesso.

## **2. Descrição do sistema**

O parque de estacionamento tem como horário de abertura e fecho, 8h e 19h respetivamente. Este, terá 2 andares( floor 0 e 1) . Assim sendo, haverá 2 rampas (and\_12 e and\_21) que fazem a ligação entre o primeiro e segundo andar e entre o segundo e o primeiro andar. Quanto ao número de entradas , o parque está munido de 3 entradas uma delas localizada no primeiro piso, floor 0 e as outras 2 no floor 1. Em relação ao número de saídas, estas serão também 3 e estarão uma delas no primeiro piso, floor 0 e as outras duas no floor 1.

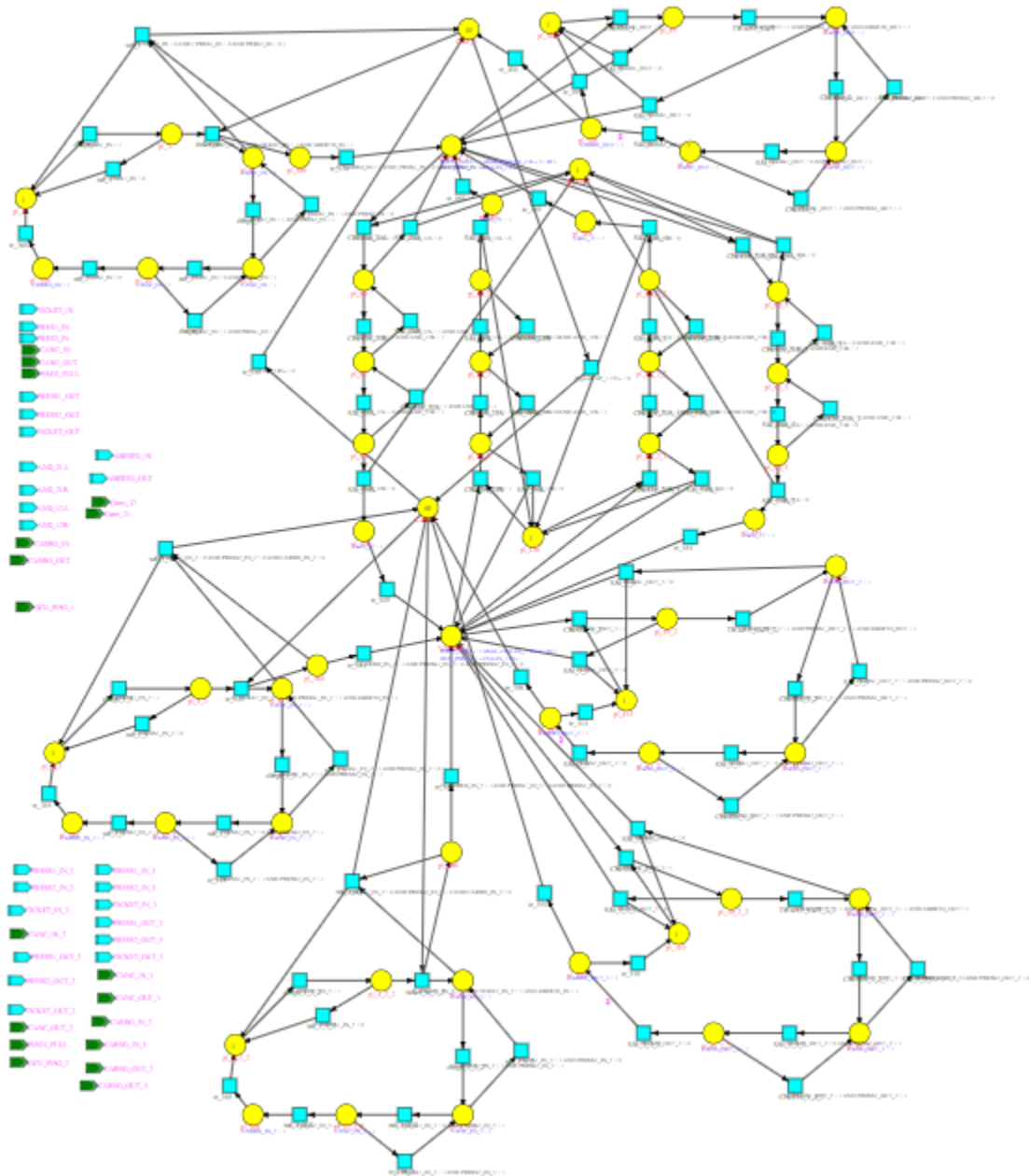
A lotação total do parque de estacionamento será 100 lugares, no primeiro andar serão admitidos 40 carros e no segundo andar serão admitidos apenas 60.

Acerca dos sensores, sabe-se que cada entrada possui 3 sensores e que o mesmo se verifica para cada saída. Em relação á rampa, cada uma possui 2 sensores (and\_XY\_A e and\_XY\_B). Contabilizando o número total de entradas, saídas e rampas do parque de estacionamento, tem-se, no total, 22 sensores, em que:

- Total de sensores nas entradas:  $3 * (3 \text{ entradas}) = 9 \text{ sensores}$ ;
- Total de sensores nas saídas:  $3 * (3 \text{ saídas}) = 9 \text{ sensores}$ ;
- Total de sensores nas rampas:  $2 * (2 \text{ rampas}) = 4 \text{ sensores}$

### 3. Controlador do parque

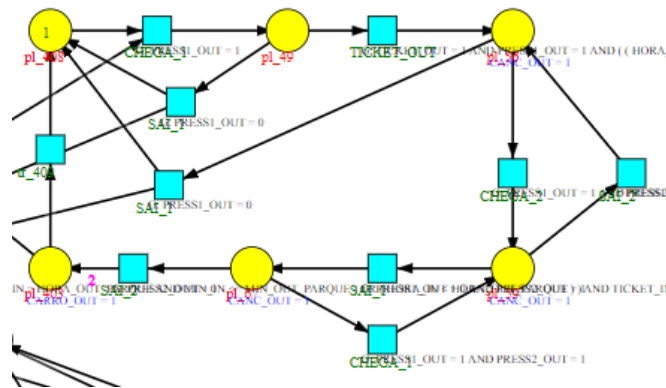
#### 3.1 PetriNet







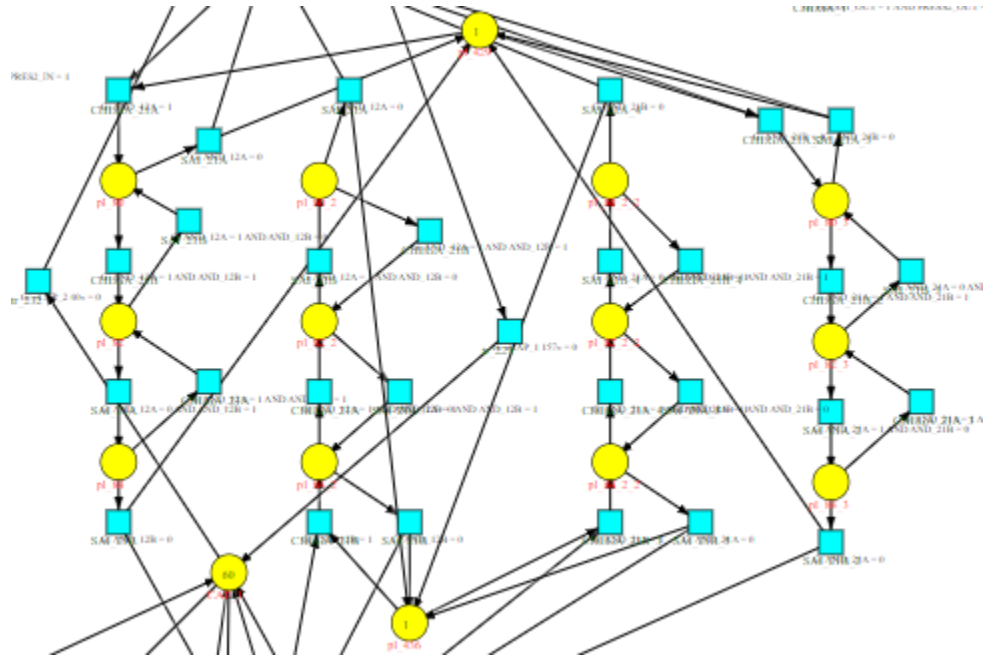
### 3.3 Saídas do parque



Todas as 3 saídas do parque são semelhantes pelo que é apenas apresentada a rede de Petri da primeira saída. Quando um automóvel pretende abandonar o parque, terá que efetuar o pagamento ativando assim o sensor *TICKET\_OUT* que irá permitir a ocupação da saída por este automóvel, levando à ativação do sensor *PRES1\_OUT* ( importante referir que a transição *chega\_1* só fica ativa se houver pelo menos uma marca no place dos lugares ocupados). Nesta transição é verificado o estado das horas e dos minutos do parque e comparados com aqueles que são recebidos do relógio de forma a verificar se o condutor se encontra dentro do tempo de funcionamento do parque. De seguida, o condutor pressiona o botão que ativa o sensor *TICKET\_OUT* que irá permitir abertura da cancela, autorizando a saída do automóvel, sendo então ativado o sensor *PRES2\_OUT*, que permite detetar se o condutor optou, efetivamente, pelo abandono do parque. Após verificar a desativação de ambos os sensores, *PRES1\_OUT* e *PRES2\_OUT*, a cancela irá fechar e voltar ao estado inicial completanto o loop de saída no parque.

A cada place (em cada passo de saída do parque), o condutor pode optar por não sair do parque enquanto espera, pela receção do ticket ou por outros fatores, desocupando a entrada do parque. Se isto se verificar desta forma, o sensores *PRES1\_OUTX* e *PRES2\_OUTX* desativam, e volta-se ao estado inicial.

### 3.4 Rampas

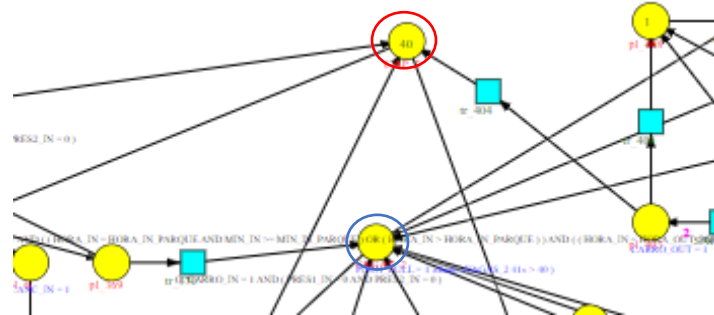


Para que haja a ativação de um sensor *AND\_XYA* ou *B*, é necessária a deteção de presença de um automóvel no início da rampa. Assim que o carro pretende avançar, é ativado, o sensor *AND\_XYB*, que permite detetar se o automóvel avançou com a mudança de piso. Em qualquer altura que o carro se encontre no início da rampa (ou seja com o *AND\_XYA* ativo), o automóvel pode fazer marcha atrás colocando assim a marca de volta no place referente á lotação do andar onde se encontra. Quando ambos sensores A e B se encontram ativos, sabe-se que o automóvel está a subir a rampa, mas não quer dizer que tenha de concluir a sua intensão, e portanto são criadas transições complementares que retiram a marca do place atual e colocam no anterior (isto representa o retrocesso nos passos de subida e descida da rampa). Assim que o automóvel transita para o piso que pretendia, ambos os sensores *AND\_XYA* e *AND\_XYB* - iram ser desativados e a rampa irá estar desocupada.

É neste último estágio do processo de subida ou descida que se sabe efetivamente que o carro concluiu a sua intensão original de transitar de piso, e portanto só aqui é que é adicionada uma marca ao place que representa o número de lugares ocupados em cada piso.

## 3.5 Lotação

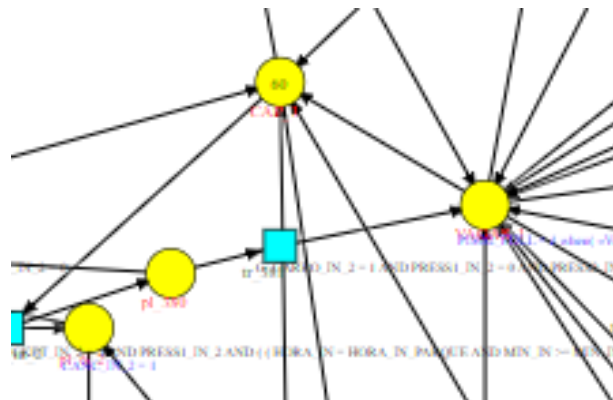
### 3.5.1 Primeiro piso



Para o primeiro piso foi definido o place, assinalado a vermelho, que representa a lotação total que vai ser admitida nesse andar. As marcas só irão sair deste place se o TICKET\_IN estiver ativo para permitir a entrada do carro, quando este completou com sucesso a sua intenção de entrar no parque este valor irá ser decrementado um valor unitário ao place até que eventualmente chegue a 0 (a partir deste momento o loop associado à entrada no primeiro piso irá ser bloqueado até que sejam repostas parcialmente as marcas neste place).

Quando se verifica que o carro efetivamente entrou será adicionada uma marca ao place, assinalado a azul, que corresponde ao número de lugares ocupados no parque.

### 3.5.1 Segundo piso

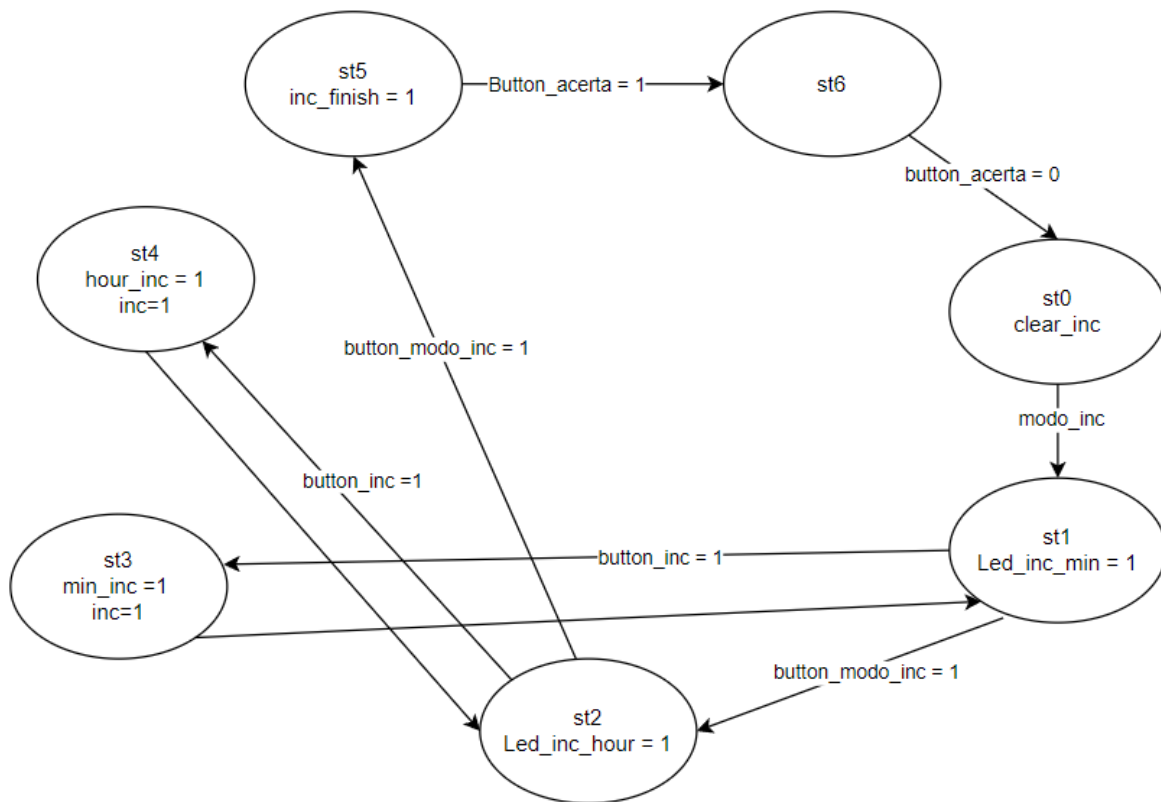


O segundo piso é muito semelhante ao primeiro pois também existe um place , assinalado a vermelho, que representa a lotação total que vai ser admitida no segundo andar. As marcas só irão sair deste place se o TICKET\_IN (de qualquer uma das entradas do segundo andar) estiver ativo para permitir a entrada do carro. Quando se verifica que a intenção do

## Sistema de controlo de um Parque de Estacionamento -Conceção de Sistemas Digitais

condutor foi bem sucedida, irá ser decrementado um valor unitário ao place até que este eventualmente chegue a 0 (a partir deste momento o loop associado à entrada no primeiro piso irá ser bloqueado até que sejam repostas parcialmente as marcas neste place). Será então adicionada uma marca ao place, assinalado a azul, que corresponde ao número de lugares ocupados no parque. Quando o primeiro piso se encontra cheio, mas o segundo não, o parque ainda deixa entrar condutores pois o segundo ainda não atingiu lotação total. Para tal efeito quando o carro entra, é retirado uma marca ao place dos lugares do segundo andar e recolocado no primeiro de forma a que o parque não bloqueie.

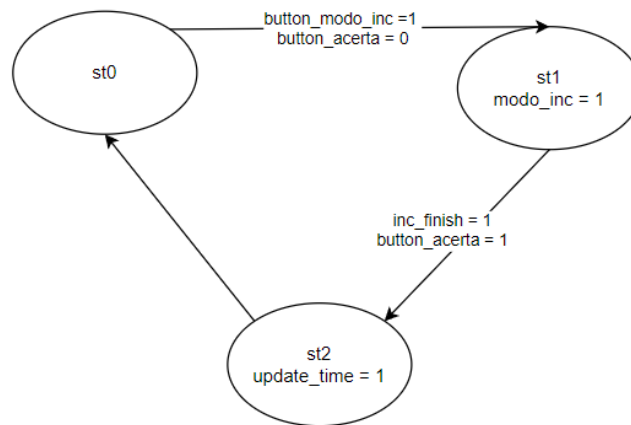
### 3.5 Controlador do incrementador



Este módulo é responsável pelo controlo do relógio. No estado 0, gerado o sinal clear\_inc que faz um clear aos valores do incrementador. Neste estado, se o sinal modo\_inc for ativado, passamos para o estado 1 no qual é feita a incrementação dos minutos através do estado 3. Sempre que o button\_inc fica ativo, vai-se para o estado 3 onde são gerados sinais

de controlo min\_inc e inc, para ser possível visualizar os passos da incrementação. A mesma lógica é aplicada para o estado 2 que é responsável pelo incremento das horas. Depois do utilizador estar satisfeito com o incremento realizado, pressiona o button que vai gerar o sinal button\_modoin e vai fazer com que se chegue ao estado 5. No estado 5 é gerado o sinal inc\_finish para sinalizar o sistema que os incrementos estão completos e seguidamente se o button\_acerta referente ao switch 6 ficar ativo, iremos para o estado 6 onde este espera que o switch volte para a posição 0 e o sistema está pronto para recomeçar a máquina de estados.

### 3.6 Controlador do incrementador



Este módulo é responsável pelo controlo do incrementador. Este módulo começa no estado 0 e quando recebe um input button\_modoin irá para o estado 1 onde emitirá um sinal de controlo para sinalizar que está no modo de incrementação. Após a incrementação estar finalizada é recebido um sinal inc\_finish que em combinação com o button\_acerta ativo, farão com que o sistema atualize a hora entrando no estado 2 e emitindo o sinal update\_time.

## **4 Dados**

### **4.1 Incrementador**

Este módulo é encarregue de fazer a contagem “manual” das unidades temporais descritas anteriormente. Consoante os sinais que recebe, este contador vai incrementando cada unidade isoladamente e começa sempre com o valor 0. Para este efeito foram utilizadas variáveis auxiliares como o `clear_inc` que serve como sinal para reiniciar o incrementador colocando o `cont_X_U` e o `cont_X_D` a 0. Para incrementar as unidades pretendidas é utilizada uma variável `X_inc` (no caso dos minutos é `min_inc`), quando esta está a 1 o sistema sabe que se encontra na incrementação das unidades corretas e verifica se as unidades e as dezenas já chegaram ao valor limite que podem tomar. Em caso afirmativo, estes dois valores fazem “reset” (colocados a 0) e posteriormente incrementados através da variável auxiliar `cont_X_U` e `cont_X_D` para começar um novo ciclo. Depois de cada ciclo de incrementação os novos valores das duas variáveis auxiliares supracitadas, irão ser colocados nas saídas do módulo `new_X_U` e `new_X_D`.

### **4.2 Interface**

Neste módulo é definido o comportamento dos botões e dos leds consoante o modo de parque ou relógio onde este se encontra. Quando o sistema está no modo parque (switch 7 a 0) o primeiro botão de pressão (button 0) irá mudar o andar do parque. Se estiver no primeiro piso e for ativado o terceiro botão servirá para retirar o ticket de entrada, e o segundo botão servirá para o ticket de saída.

Se estiver no segundo piso e o switch 0 desativado o terceiro botão (button 2) serve para o retirar um ticket na primeira entrada do segundo piso, mas quando o switch 0 está a 1 este terceiro botão já servirá para retirar o ticket para a segunda entrada do segundo piso. Este comportamento é idêntico para o retirar o ticket na saída, só que nesta o botão pressionado será o segundo (button 1). O mapeamento dos restantes botões e leds feito neste módulo é mostrado no capítulo 5 deste trabalho.

### **4.3 Relógio**

Neste módulo são definidas as variáveis `aberto_in` e `aberto_out` que servem para dividir o dia em 3 partes: quando o `aberto_in` se encontra ativo, significa que o parque se encontra dentro do horário de funcionamento normal do parque (desde as 8h às 19h). Quando o `aberto_out` está ativo e o `aberto_in` a 0, significa que o parque está no horário em que só deixa os carros sair, mas já não deixa entrar. Quando os dois sinais se encontram a 0 o parque está fechado e portanto não deixará carros entrar nem sair. Estes valores são atribuídos comparando os valores que o módulo recebe do relógio, com as horas definidas de fecho e abertura do parque.

O relógio funcionará normalmente para as horas e minutos, o que quer dizer que a cada tick do clock interno da FPGA, cuja frequência é 50MHz, adiciona um segundo à sua contagem, e quando atinge os 60 segundos, adiciona um minuto, e assim sucessivamente. Este módulo recebe uma variável `update_time` que indica se o utilizador quer acertar o relógio. Se assim for, o `cont_seg_u`, que é a variável auxiliar usada para incrementação das unidades do segundo fica a 0, pois na funcionalidade deste parque não é permitida a atualização dos segundos. Caso contrário, se a variável `update_time` estiver com o valor 0, a cada evento do clock o valor do `cont_seg_u` irá incrementar, isto se não tiver chegado ainda a 9. No caso dos minutos e das horas, caso o sinal `update_time` esteja ativo o relógio irá atualizar as suas horas para o horário definido pelo utilizador, caso contrário as variáveis correspondentes a cada unidade e dezena das horas e dos minutos são incrementadas até chegarem ao seu limite 23:59.

### **4.4 Conv\_display**

Neste módulo é feita a conversão e seleção do que o display apresenta para cada modo do nosso parque. Caso estejamos no modo relógio, ou seja caso a variável `modo_relogio` for '1', o output do display será as horas e minutos atuais. Continuando o `modo_relogio` a '1', caso seja ativado o modo de incrementação, ou seja a variável `modo_inc` for '1', então irá ser displayed os valores a ser incrementados pelo utilizador. Se este `modo_inc` for '0', então, se o `sw0`, o `sw1`, o `sw2`, ou o `sw3` ficarem a '1', sendo estas variáveis referentes aos switches da placa correspondentes, então irá

ser displayed para o sw0, a quantidade de carros que entraram no piso 1, para o sw1, a quantidade de carros que saíram no piso 1, para o sw2 a quantidade de carros que entraram no piso 2, e para o sw3 a quantidade de carros que saíram do piso 2. O grupo constatou que quando se encontra no modo\_relogio e é feito o display da de quantos carros entraram e saíram do parque os switches da direita têm prioridade sobre os switches da esquerda. Isto significa que se o switch 3 estiver a dar display do carros que saíram, se o primeiro switch for ativado o display muda sem que o switch 3 fique a 0. Isto poderia ser resolvido com recurso a condições if, mas faltou tempo.

Caso o modo\_relogio apresentar o valor '0', então o nosso display irá apresentar a lotação do parque atual de acordo com o piso selecionado.

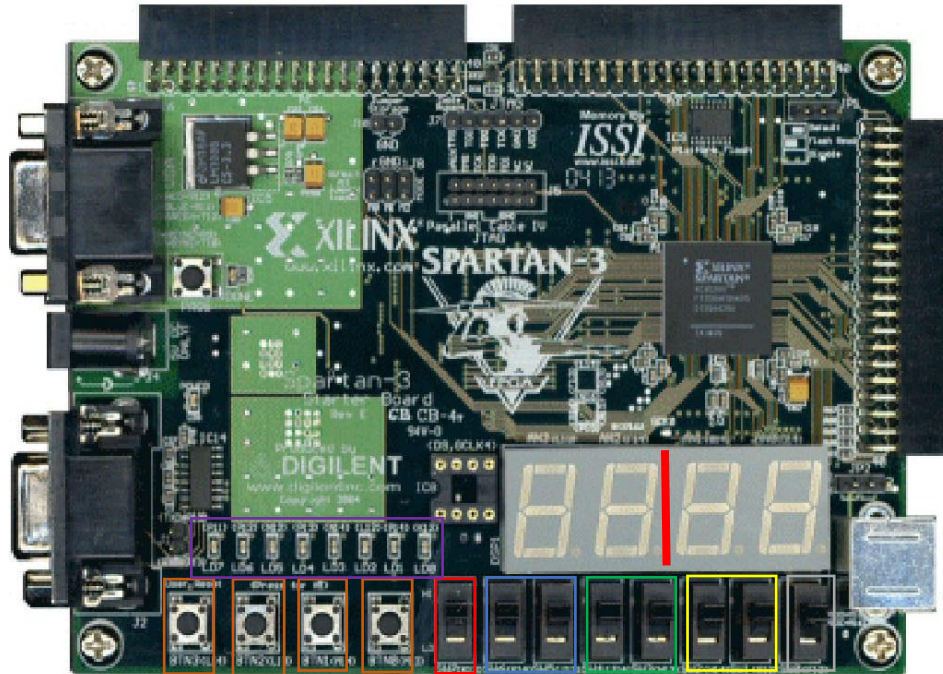
## **4.5 Functions**

Neste módulo foram desenvolvidas as funções do parque solicitadas pelos docentes nomeadamente quantos carros entraram no total e a lotação atual. Para tal é feito um processo que recebe um sinal da rede de petri referente à entrada bem sucedida de um carro, mais especificamente CARRO\_IN, CARRO\_IN\_2, CARRO\_IN\_3, que cada vez que fica ativo incrementa o número de carros que entraram no piso respectivo. O mesmo acontece para as saídas em que sempre que recebe o sinal da petri net, CARRO\_OUT, CARRO\_OUT\_2 ou CARRO\_OUT\_3, irá decrementar o número de carros que saiu do respectivo piso. Além desta funcionalidade também é implementada a funcionalidade para ver a lotação atual do parque para cada piso, sendo que esta retorna o valor do número de carros atualmente no parque para o piso 1 ou para o piso 2. Isto é feito através dos sinais referidos anteriormente e os sinais CARRO\_12 e CARRO\_21 referentes a quando um carro usa com sucesso a rampa para subir ou descer de piso, ou seja, se houver um CARRO\_12, a lotação irá incrementar no piso 2 e decrementar no piso 1, e o mesmo acontece para a situação contrária. Na mesma perspectiva, se houver um CARRO\_IN, irá incrementar a lotação no respectivo piso em que a entrada se localiza e se ouver um CARRO\_OUT irá decrementar a lotação no respectivo piso a que a saída pertence.



## 5 Implementação FPGA

### 5.1 Modo parque



Switch:

Floor 0:

- switch 0 – muda entre as duas saídas/entradas do piso 1
- switch 1 – manipula o sensor *PRES1\_IN*;
- switch 2 – manipula o sensor *PRES2\_IN*;
- switch 3 – manipula o sensor *PRES1\_OUT*;
- switch 4 – manipula o sensor *PRES2\_OUT*;
- switch 5 – manipula o sensor *AND\_12A*;
- switch 6 – manipula o sensor *AND\_12B*;
- Switch 7 – muda entre modo parque e modo relógio;

Floor 1:

- switch 0 – muda entre as duas saídas/entradas do piso 1
- switch 1 – manipula o sensor *PRES1\_IN*;
- switch 2 – manipula o sensor *PRES2\_IN*;
- switch 3 – manipula o sensor *PRES1\_OUT*;
- switch 4 – manipula o sensor *PRES2\_OUT*;
- switch 5 – manipula o sensor *AND\_12A*;
- switch 6 – manipula o sensor *AND\_12B*;
- Switch 7 – muda entre modo parque e modo relógio;

## ***Sistema de controlo de um Parque de Estacionamento -Conceção de Sistemas Digitais***

Buttons:

- *Button 0* – muda de andar
- *Button 1* – TICKET\_OUT
- *Button 2* – TICKET\_IN
- *Button 3* – reset

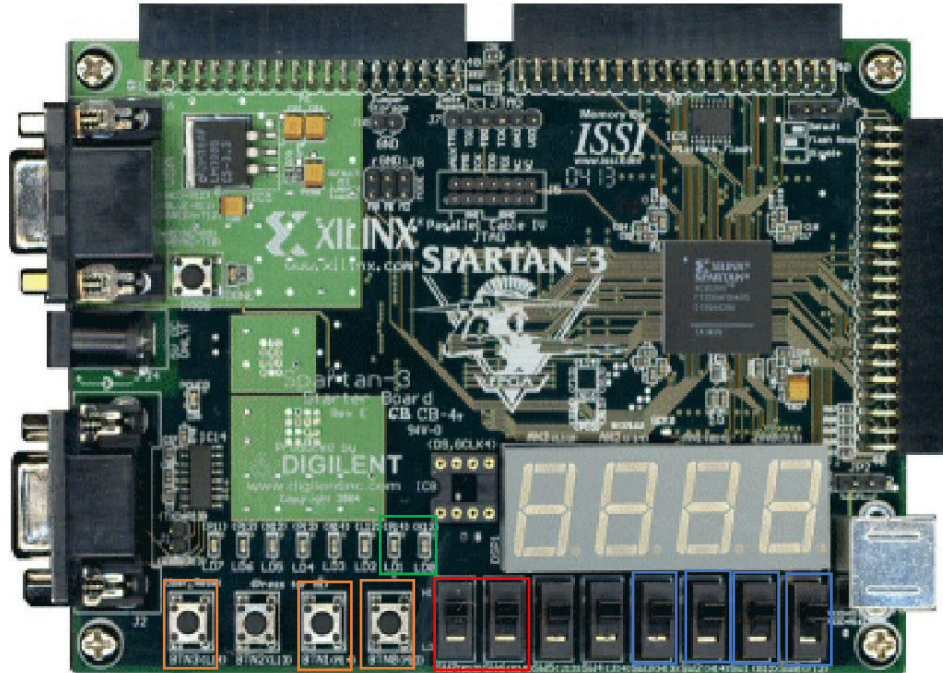
No modo parque o display da placa irá apresentar no lado esquerdo os lugares totais do piso em questão e do lado direito os lugares ocupados.

Led:

- *Led 0* – CANCEL\_IN
- *Led 1* – CANCEL\_OUT
- *Led 2* – piso 1
- *Led 3* – piso 2
- *Led 4* – mudança de entrada/saída
- *Led 5* – piso full
- *Led 6* – piso muita lotação
- *Led 7* – piso pouca lotação

É importante reforçar que, estando o utilizador no piso 1, após ser ligado o switch 0, o switch 1 fará referência ao pres\_1\_in da segunda entrada do piso 1 e o switch 2 fará referência ao pres\_2\_in da segunda entrada. Se o switch 0 estiver desativo estes mesmos switches farão referência aos mesmos sensores mas da primeira entrada. O mesmo acontece para as saídas. A versatilidade dos botões deve-se ao facto do grupo ter definido 3 entradas e três saídas, o que fará com que tenha mais sensores que botões na placa.

## 5.2 Modo relógio



Switch:

- *switch 0* – carros que entraram no piso 1;
- *switch 1* – carros que saíram do piso 1;
- *switch 2* – carros que entraram no piso 2;
- *switch 3* – carros que saíram do piso 2;
- *switch 6* – update time;
- *switch 7* – muda entre modo parque e modo relógio;

Buttons:

- *Button 0* - muda entre incremento de minutos e incremento de horas;
- *Button 1* - incrementador de minutos/horas
- *Button 3* - reset

Leds:

- *Led 0* – led do incremento de minutos
- *Led 1* – led do incremento de horas

Em modo relógio, o display irá apresentar informação referente às horas e aos minutos.

## Sistema de controlo de um Parque de Estacionamento -Conceção de Sistemas Digitais

```
1 NET "clk" LOC = "T9";
2 NET "reset" LOC = "L14";
3 NET "button_1" LOC = "M13";
4 NET "button_2" LOC = "M14";
5 NET "button_3" LOC = "L13";
6 NET "led0" LOC = "K12";
7 NET "led1" LOC = "P14";
8 NET "led2" LOC = "L12";
9 NET "led3" LOC = "N14";
10 NET "led4" LOC = "P13";
11 NET "led5" LOC = "N12";
12 NET "led6" LOC = "P12";
13 NET "led7" LOC = "P11";
14 NET "sw0" LOC = "K13";
15 NET "sw1" LOC = "K14";
16 NET "sw2" LOC = "J13";
17 NET "sw3" LOC = "J14";
18 NET "sw4" LOC = "H13";
19 NET "sw5" LOC = "H14";
20 NET "sw6" LOC = "G12";
21 NET "sw7" LOC = "F12";
22 NET "algarismo<0>" LOC = "N16";
23 NET "algarismo<1>" LOC = "F13";
24 NET "algarismo<2>" LOC = "R16";
25 NET "algarismo<3>" LOC = "P15";
26 NET "algarismo<4>" LOC = "N15";
27 NET "algarismo<5>" LOC = "G13";
28 NET "algarismo<6>" LOC = "E14";
29 NET "algarismo<7>" LOC = "P16";
30 NET "an0" LOC = "D14";
31 NET "an1" LOC = "G14";
32 NET "an2" LOC = "F14";
33 NET "an3" LOC = "E13";
```

## 6 Teste

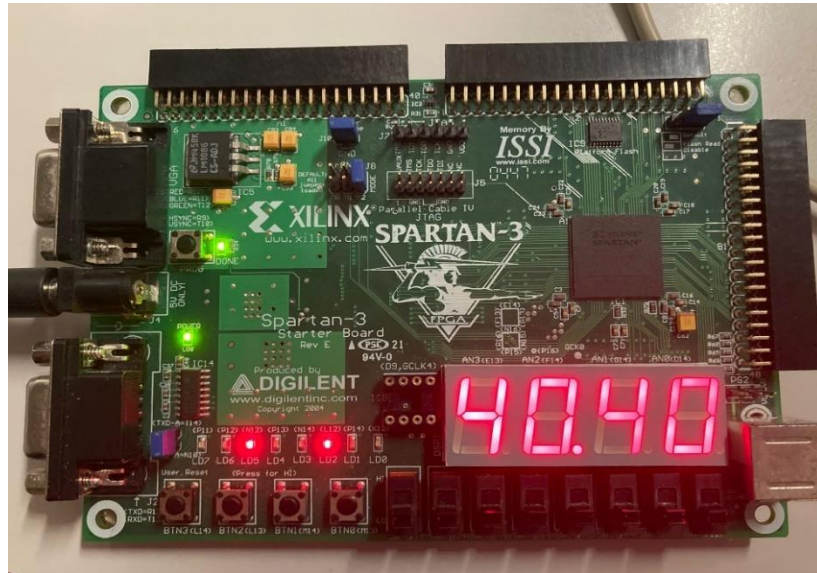


Para demonstrar o bom funcionamento do parque implementado, foi feito um teste FPGA capturados os resultados. Na foto acima, pode constatar-se que, como referido nos capítulos anteriores, o número de lugares possíveis de cada piso encontra-se no lado esquerdo do display enquanto a lotação atual do piso, no lado direito. Pode-se observar também na foto

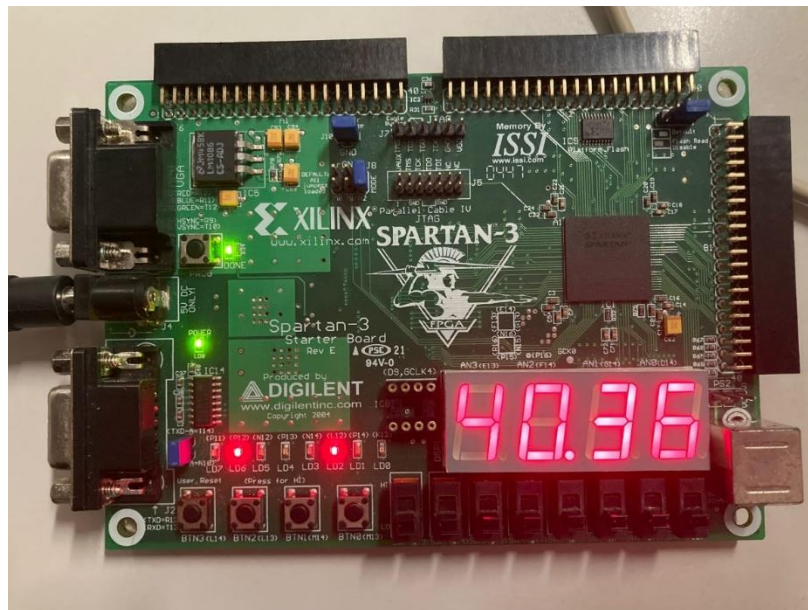


## Sistema de controlo de um Parque de Estacionamento -Conceção de Sistemas Digitais

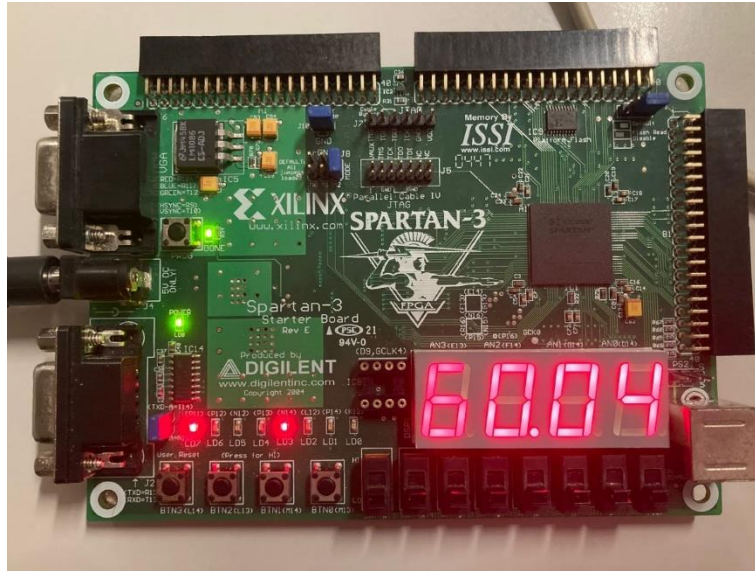
que o piso em que o utilizador se encontra é o 0 pois é o led 2 que se encontra aceso e também que o parque está muito lotado pois o led 6 se encontra também aceso.



Nesta segunda foto verifica-se que o parque está cheio, devido ao led 5 estar ativo, e também que o utilizador ainda se encontra no piso 0.



Na terceira foto, verifica-se que foram retirados ao piso 0, 4 lugares ocupados, e isto foi feito via rampas de acesso ao segundo andar. Assim que os lugares do piso 0 ficaram vagos o led referente á lotação total do parque (led 5) desativou e acendeu de novo o led 6 que indica que o parque está muito cheio.



Nesta quarta e última foto constata-se que o utilizador está no segundo piso (led 3) e que neste se encontram os quatro carros que saíram do primeiro piso pelas rampas. Note-se que agora o led 7 está de novo aceso pois a lotação do segundo piso é apenas 4 lugares o que é um valor baixo.

## **7 Conclusão**

Em suma, o objetivo de implementação e desenvolvimento do controlador do parque de estacionamento para o seu bom funcionamento foi devidamente cumprido. Os requisitos propostos foram cumpridos como proposto e o grupo sente que um dos maiores problemas enfrentados foi na integração do relógio digital do trabalho anterior para determinar a hora de entrada e a hora de saída do parque.

O grupo não conseguiu implementar a mudança das horas de entrada e saída do parque, os minutos totais dos carros no parque e só existe informação atual do parque, não foi implementado com sucesso o histórico das informações do parque.

Uma das dificuldades sentidas pelo grupo foi não conseguir separar os números que vinham das horas em dezenas e unidades para facilitar o cálculo do horário de funcionamento do parque e foi este um dos fatores que levou a que não fosse implementada a funcionalidade de mudar o horário de funcionamento.

A conceção do sistema de forma a possuir distinções entre os módulos dados e controlo, permite uma fácil integração dos componentes que se possui ou a outros eventuais projetos, sendo este considerado um ponto forte do projeto. Em relação aos pontos fracos, pode-se considerar a modelagem do projeto complexa devido ao xilinx, e a compreensão do mesmo árdua.

## ***Sistema de controlo de um Parque de Estacionamento -Conceção de Sistemas Digitais***