

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Licenciatura em Engenharia Informática e de Computadores



PROJETO FINAL

Arquitetura de Computadores, Ano Letivo 2023/2024

LEIC 23D, Grupo 12
Gonalo Carvalho, 49219
Joo Martins, 50055

Docente:
Joo Patriarca

Índice

Elementos relevantes para a compreensão do trabalho realizado.....	1
Máquina de estados.....	2
Solução adotada para ligar o circuito pTC à placa SDP16.....	4
Cálculos realizados para determinar as temporizações envolvidas.....	4
Latência máxima no atendimento dos pedidos de interrupção.....	5
Pior tempo de atendimento às interrupções externas.....	5
Conclusões.....	6

Elementos relevantes para a compreensão do trabalho realizado

Pretendia-se a implementação da simulação de um sistema que simulasse um par de semáforos, um para automóveis e outro para pedestres, e que fizesse uso de um botão para estes últimos a fim de permitir a mudança de estado dos semáforos de modo a permitir a passagem dos peões. Esta simulação foi implementada com recurso à placa SDP16 usada ao longo do semestre e respectivos periféricos, bem como do módulo LAPI.

Este projeto fez então uso da SDP16 e respectivos portos de entrada e saída, bem como o periférico pico Timer, implementado através de um circuito integrado PAL ATF750C.

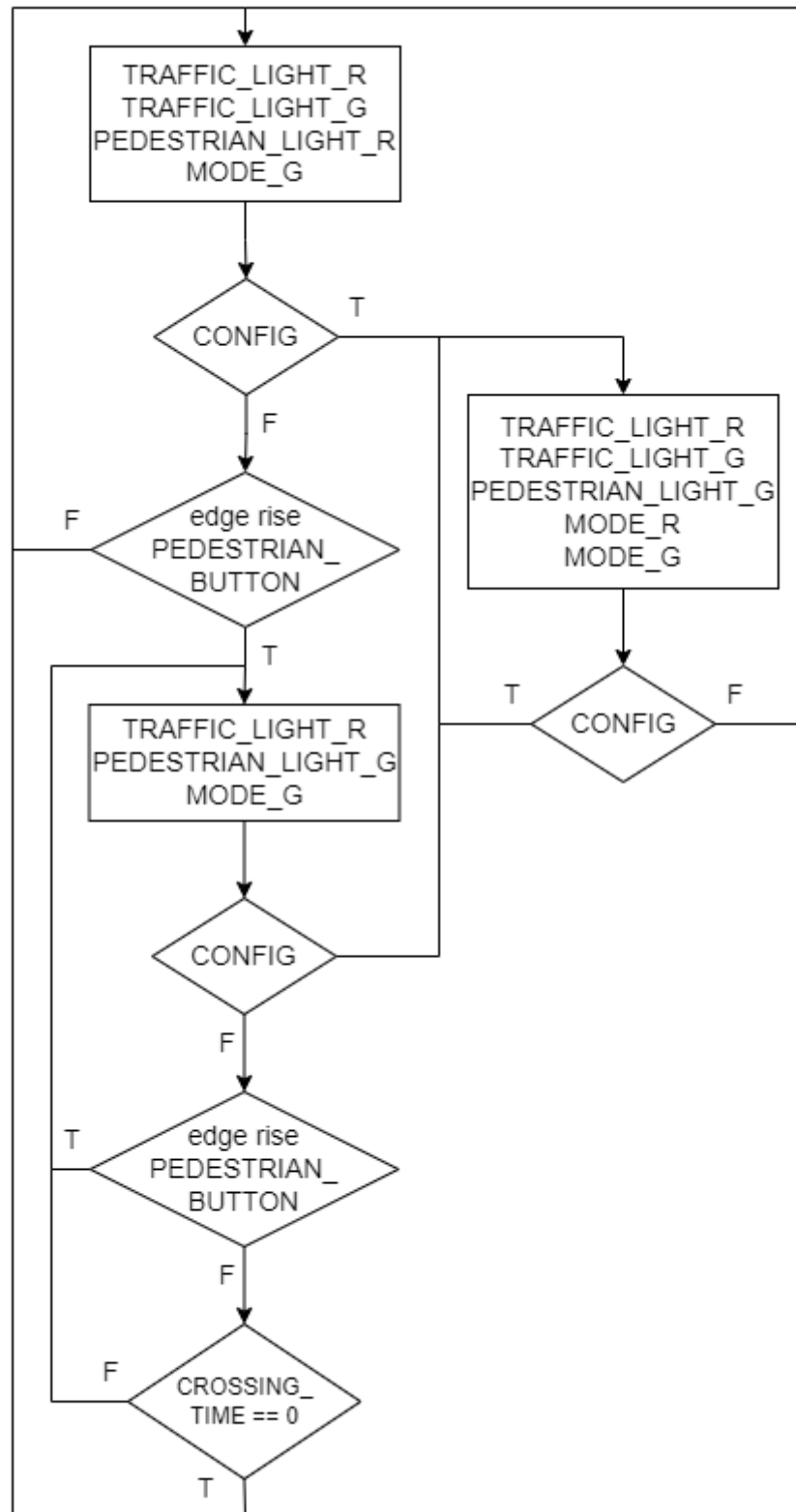
A interação com o porto de entrada permite alternar entre o modo de execução e o modo de configuração. No primeiro é possível interagir com o semáforo de modo a permitir a passagem de peões premindo um botão. No modo de configuração é possível usar dip-switches para determinar o tempo durante o qual o sinal de travessia de peões deve estar aberto

O porto de saída vai interagir com o módulo LAPI de modo a que os seus LED simulem a luz a exibir em cada semáforo.

O circuito pTC vai ser utilizado para toda a temporização do sistema.

Máquina de estados

A análise do enunciado permitiu-nos chegar à seguinte máquina de estados, que serviu como base lógica do funcionamento do programa desenvolvido:



Em suma, o programa pode encontrar-se em modo de execução ou em modo de configuração.

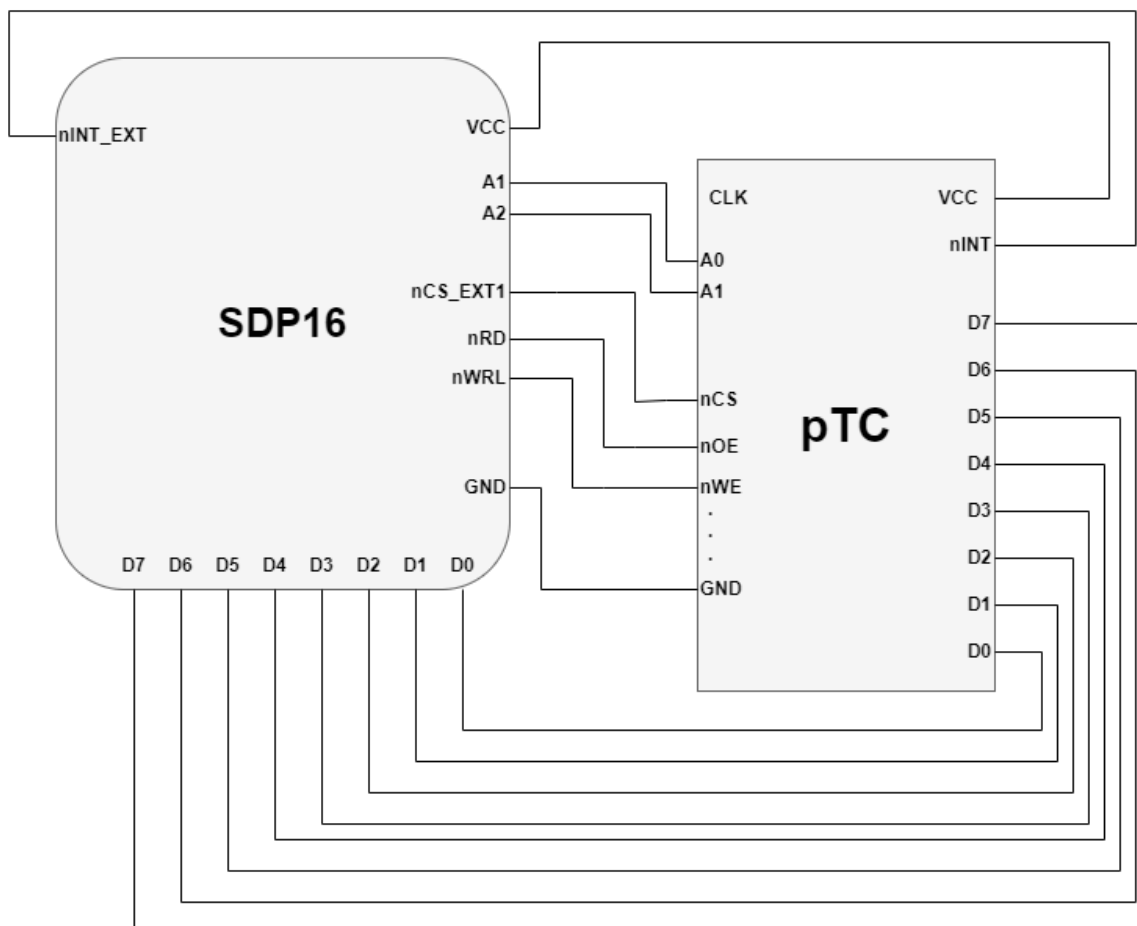
No modo de execução, existem dois estados distintos. No primeiro destes estados, os sinais `TRAFFIC_LIGHT_RED` e `TRAFFIC_LIGHT_GREEN` estão ambos ativos, levando ao piscar em amarelo do LED representante do semáforo automóvel. Os sinais `PEDESTRIAN_LIGHT_RED` e `MODE_G` encontram-se também ativos, exibindo nos LED correspondentes as cores vermelha e verde, respetivamente. O programa sai deste estado para o próximo estado do modo de execução quando é detetado um flanco ascendente no botão de pedestres ou para o modo de configuração quando o dip-switch que controla o modo passa para o valor lógico '1'.

No segundo estado do modo de execução os sinais `TRAFFIC_LIGHT_GREEN` e `PEDESTRIAN_LIGHT_RED` são desativados e o sinal `PEDESTRIAN_LIGHT_GREEN` é ativado. Ao mesmo tempo, é iniciado um timer (cuja duração pode ser configurada pelo uso de dip-switches no modo config). A cada vez que volta a ser detetado um flanco ascendente no botão de pedestres o timer é resetado, voltando então a máquina de estados ao início deste estado. Quando o timer chega a 0 a máquina de estados volta ao estado inicial. A qualquer momento pode ser ativo o sinal de '*config*', passando então para o estado do modo de configuração.

No modo de configuração estão ativos os sinais `TRAFFIC_LIGHT_RED`, `TRAFFIC_LIGHT_GREEN`, `PEDESTRIAN_LIGHT_GREEN`, `MODE_RED` e `MODE_GREEN`. O LED controlados pelos sinais `MODE_GREEN` e `MODE_RED` deve estar permanentemente aceso e os restantes intermitentes. O programa sai deste estado de volta para o estado inicial quando o sinal de '*config*' volta a tomar de novo o valor lógico '0'.

Solução adotada para ligar o circuito pTC à placa SDP16

Para a ligação entre o circuito pTC e a placa SDP16 foram utilizadas as mesmas ligações já utilizadas anteriormente no âmbito do lab 05.



É de realçar que, de forma a facilitar a representação, a posição de cada entrada no diagrama pode não corresponder à sua posição física nos dispositivos reais. A entrada CLK, correspondente ao relógio, deve também estar conectada, não ao SDP16, mas à saída do oscilador da placa ATB (neste caso, na linha correspondente a 1 kHz), de forma a que este possa ser usado como base do relógio interno a usar.

Cálculos realizados para determinar as temporizações envolvidas

Para base das temporizações envolvidas no trabalho foi usado o oscilador com ritmo de 1kHz da placa ATB, com o registo Timer Match Register do circuito pTC a 99. Assim sendo, há um incremento da variável global sysclock por nós implementado a cada 100

ciclos da ATB, ou seja, 100 milissegundos. Isso permite definir os 0.5 segundos de intermitência dos LED do periféricos LAPI, através de 5 incrementos da variável sysclock.

De forma semelhante, guardando num *array* os valores [100, 200, 300, 400, 500], podemos definir os tempos de espera como 10, 20, 30, 40 ou 50 segundos respetivamente. Os valores guardados foram obtidos com base na frequência de 100ms definida para a geração de pedidos de interrupção por parte do circuito pTC.

Latência máxima no atendimento dos pedidos de interrupção

O sistema irá ter latência máxima no atendimento de um pedido de interrupção gerado pelo pTC se esse pedido de interrupção se der no início de uma instrução que execute acessos à memória, uma vez que são estas instruções que requerem um maior número de ciclos de relógio interno para serem completadas.

O relógio interno da placa SDP16 funciona a 50 kHz, pelo que cada ciclo tem 0.02 ms de duração. Como uma instrução de acesso à memória (leitura/escrita em memória/*stack*) executa em 6 ciclos de relógio, é possível concluir que a latência máxima nos pedidos de interrupção gerados pelo circuito pTC é de $0.02ms * 6 = 0.12ms$.

Pior tempo de atendimento às interrupções externas

O relógio interno da placa SDP16 funciona através de um sinal de onda quadrada com frequência igual a 50 kHz.

A rotina de interrupção implementada conta com 2 instruções sem acesso à memória e 9 instruções com acesso à memória. Cada instrução com acesso à memória demora 2 ciclos-máquina a executar e cada instrução sem acesso à memória demora 1 ciclo-máquina a executar. Uma vez que cada ciclo-máquina corresponde a três ciclos de relógio temos que $T_{exec} = \frac{60 \text{ ciclos}}{50 \text{ kHz}} = 60 \text{ ciclos} * 20 \text{ ns} = 1.2ms$.

A rotina de interrupção não conta com qualquer salto, pelo que executa as mesmas instruções apenas uma vez. Desta forma, conclui-se que será sempre este tempo que será necessário para o atendimento de uma interrupção externa.

Conclusões

O desenvolvimento deste trabalho permitiu pôr em prática a vasta maioria dos conteúdos programáticos lecionados durante o semestre no âmbito da Unidade Curricular de Arquitetura de Computadores.

Tendo por base a escrita de um programa na linguagem Assembly do P16 foi possível aplicar as convenções usadas no desenvolvimento deste software, tais como o planeamento de uma máquina de estados como suporte lógico e a organização do programa em rotinas, a preservação dos registos convencionados, etc.. O seu desenvolvimento requereu também a interação com portos de entrada e de saída, bem como de temporizações, que por sua vez requerem o atendimento de interrupções externas, controladas através da interação com um outro periférico.