



RELATÓRIO

Introdução à Arquitetura de Computadores

2014/2015 - 1º Semestre

TRON

Grupo: 17

Turno: Quinta-feira, 08:00H – 09:30H

André Mendonça nº 82304

Alice Dourado nº 81205

Gonçalo Ribeiro nº 82303

Introdução

Para este projeto, o objetivo foi criar o jogo TRON usando a linguagem de programação Assembly e executando esse mesmo jogo na placa do processador P3.

Organização e estrutura do Jogo

O código está dividido em 4 partes: definição de constantes, variáveis, interrupções e o programa principal. Neste relatório vamos falar sobre as 2 últimas.

Interrupções do programa:

Para este programa foi necessário definir as seguintes interrupções:

<i>INTERRUPÇÕES</i>	<i>DESCRIÇÃO</i>
INT_1	Está associada à rotina de interrupção START, tem como objetivo incrementar a variável GAME_ON e permite iniciar o jogo.
INT_0	Está associada à rotina de interrupção X_esq, tem como objetivo incrementar a variável Next_x_esq e permite que o jogador 1 mude a sua direção para a esquerda.
INT_B	Está associada à rotina de interrupção X_dir, tem como objetivo incrementar a variável Next_x_dir e permite que o jogador 1 mude a sua direção para a direita
INT_7	Está associada à rotina de interrupção #_esq, tem como objetivo incrementar a variável Next_Y_esq e permite que o jogador 2 mude a sua direção para a esquerda
INT_9	Está associada à rotina de interrupção #_dir, tem como objetivo incrementar a variável Next_Y_dir e permite que o jogador 2 mude a sua direção para a direita
INT_15	Está associada à rotina de interrupção temporizador, que por sua vez chama a Rotina_temp (inicia o temporizador e define o valor da "variável" 0)

Programa principal

Mensagem inicial e LCD

Na primeira fase, faz-se a escrita das mensagens iniciais. O programa principal começa na etiqueta "Inicio" na qual começa por inicializar a pilha e ativar interrupções, depois chama a rotina "cursor_on", que ativa o cursor, depois é chamada a rotina "Limpar" que apaga todo o ecrã, escrevendo um espaço em branco (" ") em todas as coordenadas. De seguida escreve-se as frases pretendidas com recurso à rotina "Escreve", que utiliza as coordenadas em R3 e a string em R1 para escrever na janela de texto, assim fica lá escrito as seguintes frases "Bem-vindo ao TRON" e "Pressione I1 para começar". Depois escreve-se no LCD a informação do tempo máximo e pontuações, pondo a string em R1 e as coordenadas em R2 e chamando a rotina "Esc_lcd" que faz a escrita no LCD.

Jogo

Depois de aparecer a mensagem inicial entra-se num ciclo infinito, na etiqueta "on" até que seja premido I1, depois move-se a tabela "coord_anterior" para R7, pois este registo vai ser usado durante todo o programa para colocar valores de coordenadas na tabela e verificar colisões. Depois chama-se a rotina "Moldura" que tem como função, escrever o espaço de jogo e os seus limites, esta começa por chamar a função "Limpar" e depois usando coordenadas definidas na 1ª parte do programa que indicam os cantos da moldura, escreve os seus caracteres, começando pelas linhas horizontais e por fim faz as verticais, durante este processo usa a rotina "esc_car" que escreve um caracter na coordenada em R2.

De seguida apaga-se os LEDs, o que é desnecessário no primeiro jogo mas quando se jogar novamente já importa, e coloca-se o valor de coordenadas iniciais em variáveis, depois inicia-se chama-se a rotina "rotina_temp" que está relacionada com a iniciação do temporizador.

Depois entra-se no ciclo principal de jogo, este chama a rotina "Contas", que mete o número que se encontra no Contador no R1 e depois faz as divisões necessárias para converter de hexadecimal para decimal, com cada algarismo num registo diferente, depois escreve esses algarismos no display de 7 segmentos, com exceção do menos significativo pois o número do contador está em décimos de segundo, e ignorando esse algarismo o número que aparece no temporizador fica em segundos. A próxima rotina é a "espera", que é um ciclo infinito que só avança quando passar uma decima de segundo no temporizador. Depois vai-se para "muda_nivel", que verifica se o tempo atual se encontra nos 10, 20, 40 ou 60 segundos e se for esse o caso muda a velocidade e acende os LEDs adequados. Quando essa rotina acaba há um salto para "wait_move" que verifica se já passou o tempo definido pela velocidade, se não tiver volta para "ciclo" se já tiver continua o movimento e chama-se a rotina "move".

A rotina "move" é aquela que realiza o movimento dos jogadores, começa por pôr os valores de variáveis específicas para o jogador 1 em variáveis genéricas e depois chama a rotina "mexe", depois volta a guardar os valores das variáveis específicas, depois faz o mesmo procedimento para o jogador 2.

A rotina "mexe" verifica qual era a direção anterior do jogador, dependendo desta pode saltar para 4 etiquetas diferentes que verificam qual é a próxima direção comparando com as variáveis que são incrementadas pelas interrupções se forem premidos os botões, depois pode saltar para uma das 4 etiquetas que realizam o movimento ("vai_pa_baixo", "vai_pa_cima", "vai_pa_esq", "vai_pa_dir") estas fazem a alteração necessária ao registo(R5) que tem a coordenada atual no depois salta para "verifica_colisoes" que verifica se a coordenada atual é igual a alguma anteriores, para isso move "coord_anterior" para um registo(R4) e compara R5 com todas as coordenadas da tabela até que R4 seja igual a R7, pois este é o registo que indica a ultima posição em que se armazenou coordenadas na tabela. Se houverem colisões e verifica-se quem é que perdeu, incrementando uma das variáveis "loser_x" ou "loser_y", se houver empate incrementa-se as 2 variáveis loser, depois de se ter verificado quem perdeu, há um salto para a etiqueta "final" para o jogo acabar. Se não houverem colisões salta-se para a etiqueta "cont_mov" e aí mete-se a coordenada atual na memória de R7, ficando assim na tabela de "coord_anterior" e escreve-se o caracter na coordenada atual da janela de texto.

Fim do jogo

Quando o jogo acaba há um jump para a etiqueta "final", onde o temporizador é desligado, e se compara o tempo do jogo que está na variável "Contador" com o que se encontra na variável "tempo_max", e o maior destes fica em R1

De seguida é chamada a rotina "update_LCD1", escreve o tempo máximo no LCD, utilizando o valor que se encontra em R1. Primeiro dividir o numero por 10d para não contar decimas de segundo, depois faz mais algumas divisões que vão converter o numero para decimal com cada algarismo num registo, de seguida adiciona-se 48d a cada um desses registos, pois esse numero representa o caracter "0" em ASCII e assim pode-se escrever os algarismos no LCD, para tal move-se a "coord_LCD" para R6 e escreve-se os números, sendo necessário ainda chamar a rotina "next_nu" antes de se escrever algarismos diferentes, pois esta incrementa o pointer do LCD.

Depois é chamada a rotina "escreve_fim" que escreve algumas strings na janela de texto (com recurso a rotina "Escreve"), com a mensagem de fim de jogo, que vai incluir a string "Empate", mesmo que esse não tenha sido o resultado do jogo, pois se alguém ganhar isso vai ser escrito nas mesmas coordenadas e apaga a palavra anterior.

A seguir verifica se o jogador 1 ganhou, e se sim chama a rotina "winner_x", e depois verifica se o jogador 2 ganhou, o que chama a rotina "winner_y". Estas rotinas funcionam de forma semelhante, aumentam a variável que tem a pontuação do jogador, escrevem na janela de texto uma mensagem que diz que ele ganhou, convertem o número de hexadecimal para decimal e depois para ASCII, depois faz-se um jump para a etiqueta "score" que se encontra dentro da rotina "update_LCD" e escreve o número no LCD.

Finalmente são postos novamente os valores iniciais em algumas variáveis, para que o programa funcione adequadamente quando o jogo recomeçar, ativa-se as interrupções e faz-se um jump para a etiqueta "on", que é a parte do programa em que se está num ciclo infinito até que seja premido I1 para o jogo começar novamente.

Aspetos a destacar

Durante o programa as variáveis relacionadas com o jogador 2 são referiadas formas diferentes, as vezes com Y outras com #, pois inicialmente a intenção era que todas tivessem o caracter '#' mas havia erro de sintaxe quando se dava um nome do género de "Next_#_esq" por isso alguns dos nomes ficaram com a letra Y em vez do caracter que queríamos colocar.

-Diferenças entre o enunciado e código:

Neste programa implementamos apenas uma funcionalidade extra, quando o jogo acaba para além das mensagens que são pedidas no enunciado do projeto para colocar na janela de texto, também colocamos uma frase que indica quem ganhou ou se houve um empate, neste caso nenhum dos jogadores ganha pontos.

Conclusão

Neste projeto conseguimos implementar o jogo do TRON em Assembly para jogar na placa do processador P3, usando os conhecimentos lecionados na disciplina de Introdução à Arquitetura de Computadores.