**Projeto Assembly Parte 2**

**Grupo 95**

**João Morais Nº 86876**

**Pedro Abreu Nº 92541**

No âmbito da cadeira de Introdução à Arquitetura de Computadores foi-nos pedido que desenvolvêssemos um programa que simula o movimento de uma bola num plano inclinado em Assembly P4. Para tal, foram criadas rotinas que utilizam o Terminal do Simulador do P4 e o Temporizador.

A simulação resume-se a um loop controlado pelo temporizador, neste loop é chamada a “Main” do programa.

Os valores que o utilizador pode alterar são: a constante VALORTEMPO(1 = 100ms), que determina o intervalo de tempo entre atualizações da posição e velocidade da bola, e as variáveis V\_INI, POSICAO\_INI, INTERVALO\_TEMPO(conversão do VALORTEMPO através da funcao CONV\_TEMPO).

O programa começa por “desenhar” no terminal a bola e o plano com auxílio da função TEXT\_WRITE. De seguida é ativado o temporizador que por sua vez faz com que a Main seja chamada. Na Main é chamada a função APAGA\_POS que apaga a posição anterior, de seguida é chamada a função POSICAOX que calcula a posição da bola através da velocidade inicial, posição inicial e aceleração. O resultado é guardado na POSICAO\_INI, pois esta é a nova posição inicial. De seguida, a função VELOCIDADEX calcula a velocidade da bola e guarda este valor na V\_INI.

A aceleração é calculada pela fórmula a=x\*(g/255), onde x representa o valor lido no acelerómetro.

No final da main é atualizada a posição da bola no terminal através da função ADDPOSTERMINAL. Em caso de colisão a função calcula a nova velocidade e a posição da bola, as constantes V\_INI e POSICAO\_INI também são atualizadas.

Qualquer multiplicação feita no programa recorre à função PRODUTO, que é uma alteração da função dada pelos docentes, que permite multiplicar números representados em vírgula fixa. Neste projeto usamos os 8 bits mais significativos para a parte inteira e os restantes 8 bits para a parte fracionária.

Para a segunda Parte do projecto o PRODUTO foi alterado de forma a dar o resultado de multiplicações negativas, através de uma flag (N\_FLAG) que é levantada quando utilizado um número negativo na multiplicação. No final caso a flag esteja levantada e adicionado 1 bit mais significativo de forma a representar o número negativo. A função não está preparada para a multiplicação de 2 negativos pois este caso nunca acontece no programa.