

Diseño Basado en Microprocesadores

Práctica 4

Programación de la FPU

Índice

1. Ejercicios	1
1.1. Ejercicio 1	1
1.2. Ejercicio 2	2

1. Ejercicios

1.1. Ejercicio 1

Escribe una función en ensamblador para ser enlazada con C que use instrucciones de la FPU para calcular la media aritmética de un vector de valores de tipo real doble. El prototipo de la función será

```
double media(double *vector, unsigned int numdatos);
```

El argumento `vector` señala al primero de los elementos del vector. El argumento `numdatos` indica el número de elementos en el vector.

La función retorna la media aritmética de los elementos del vector. Si el argumento `vector` es un puntero nulo o si el argumento `numdatos` es 0 la función debe retornar 0.0.

Escribe un pequeño programa en C en el que se llame a `media` y comprueba su funcionamiento.

INDICACIONES:

El tipo `double` tiene un tamaño de 64 bits. Para cargarlo desde la memoria con `FLD` o sumarlo desde la memoria con `FADD` hay que utilizar el especificador de tamaño `qword` en el operando de memoria.

Para dividir directamente `ST0` entre un dato entero almacenado en la memoria puede usarse la instrucción `FIDIV` (consulta el resumen de instrucciones de la FPU). Como queremos dividir entre un `unsigned int` (que tiene 32 bits) habrá que utilizar el especificador `dword` para indicar el tamaño del dato por el que queremos dividir.

1.2. Ejercicio 2

Escribe una función en ensamblador que use instrucciones de la FPU para evaluar un polinomio de grado n :

$$P_n(x) = c_n x^n + c_{n-1} x^{n-1} + \dots + c_1 x + c_0$$

para un valor dado de x . El prototipo de la función será

```
double eval_poli(double *coefs, double x, unsigned int grado);
```

El argumento `coef` señala al vector con los coeficientes del polinomio que están almacenados empezando por c_n y terminando por c_0 . El argumento `x` es el valor de x para el que debe evaluarse el polinomio. El argumento `grado` indica el grado del polinomio. El número de coeficientes almacenados a partir de la posición señalada por `coef` es igual al grado del polinomio más 1.

La función retorna el resultado de evaluar el polinomio. Si el argumento `coef` es un puntero nulo la función debe retornar 0.0.

INDICACIONES:

La FPU no permite realizar la operación x^y directamente para valores arbitrarios de x e y . El cálculo de las sucesivas potencias de x puede realizarse multiplicando reiteradamente por x pero es mejor reorganizar la expresión de $P_n(x)$ según el método de Horner:

$$P_n(x) = c_n x^n + c_{n-1} x^{n-1} + \dots + c_1 x + c_0 = (\dots((c_n x + c_{n-1})x + c_{n-2})x + \dots)x + c_0$$

Por ejemplo, para un polinomio de grado 4

$$P_4(x) = c_4 x^4 + c_3 x^3 + c_2 x^2 + c_1 x + c_0 = (((c_4 x + c_3)x + c_2)x + c_1)x + c_0$$

De esta forma, evaluar un polinomio de grado n requiere sólo n multiplicaciones y n sumas.

Escribe un pequeño programa en C que use la función `eval_poli`. Ejecuta el programa para comprobar su funcionamiento.