

Vídeo 1 - Introducció a la Fase 1 i format E9M22

Hola, en aquest vídeo explicaré la primera part de la pràctica de Fundaments de Computadors. Aquesta fase gira al voltant del format de coma flotant personalitzat anomenat E9M22. Aquest format fa servir 1 bit de signe, 9 bits per l'exponent amb biaix de 255, i 22 bits de mantissa amb bit implícit. A diferència del format IEEE754, aquí hem de tractar manualment amb casos especials com el zero, els infinits i els NaN, que tenen una codificació específica.

La primera tasca ha estat implementar diferents operacions aritmètiques bàsicament amb aquest format. Per això, hem utilitzat un conjunt de funcions auxiliars escrites en llenguatge ensamblador ARM. Aquestes funcions són: `E9M22_add`, `E9M22_sub`, `E9M22_neg`, `E9M22_abs`, `E9M22_mul_s`, `E9M22_are_eq` i `E9M22_are_unordered_s`. Totes aquestes rutines treballen sobre enters de 32 bits però simulen el comportament del format flotant E9M22.

Per exemple, la suma i la multiplicació han de gestionar desalineaments d'exponents, normalització de resultats, i detecció de casos com NaN o operands infinits. També cal assegurar-nos que les funcions respecten el comportament esperat quan comparem valors que podrien ser iguals, infinits, o no ordenables.

Per garantir la correcció del nostre codi, hem fet servir una sèrie de fitxers de prova proporcionats, i hem comprovat els resultats amb una versió en C que actua com a oracle. Cada funció s'ha aïllat i provat individualment abans d'integrar-se.

En el proper vídeo parlaré amb més detall del procés de desenvolupament d'aquestes funcions i de les dificultats que han sorgit.

Vídeo 2 - Desenvolupament detallat de les operacions E9M22

Continuant amb la Fase 1 de la pràctica, ara explicaré amb més detall com he desenvolupat les funcions que treballen amb el format E9M22. L'objectiu era entendre a fons com es representa i manipula un nombre en format flotant, tot implementant operacions aritmètiques bàsiques a baix nivell.

La funció `E9M22_add`, per exemple, ha estat una de les més complexes. Ha calgut descomposar els operands en signe, exponent i mantissa, comparar els exponents per alinear les mantisses, fer la suma o resta segons el signe, i finalment normalitzar el resultat. En aquest procés, cal tenir en compte molts detalls: desbordaments, subnormals, i arrodoniments.

També ha estat interessant la funció `E9M22_are_unordered_s`, que verifica si dos valors no són ordenables, és a dir, si almenys un dels dos és NaN. Això és important per complir amb les normes del comportament de nombres especials.

Durant el desenvolupament, una eina fonamental ha estat el debugger, que m'ha permès analitzar pas a pas com es modificaven els registres. També he fet servir la sortida de fitxers de prova per comparar els valors binaris obtinguts amb els esperats.

Finalment, puc dir que aquesta part m'ha ajudat a entendre molt millor el funcionament intern dels nombres en coma flotant i la complexitat d'implementar-los sense suport directe del hardware. En els pròxims vídeos explicaré la segona fase, que tracta sobre el processament de dades de temperatures.

Vídeo 3 - Introducció a la Fase 2 i estructuració del projecte

En aquest tercer vídeo començaré explicant la segona fase de la pràctica. Aquesta part es basa en el tractament de dades reals emmagatzemades en una matriu. Concretament, es tracta d'una matriu de temperatures, on cada fila representa una ciutat i cada columna representa un mes. Les dades estan codificades en format E9M22, i l'objectiu és calcular estadístiques com la mitjana, el màxim i el mínim per ciutat i per mes.

Per fer-ho, es treballa sobre dues funcions en C: `avgmaxmin_city()` i `avgmaxmin_month()`. Aquestes funcions han estat traduïdes a llenguatge ensamblador ARM com `avgmaxmin_city_s` i `avgmaxmin_month_s`, respectivament. La seva estructura ha estat pensada per aprofitar al màxim les funcions auxiliars implementades a la primera fase, especialment les de suma, comparació i producte.

En aquesta fase he seguit una metodologia organitzada, començant per entendre bé el codi original en C, i després fent una traducció pas a pas, mantenint l'estructura de bucles i condicions. També he tingut molta cura a l'hora de passar els paràmetres per registres, gestionar l'stack i conservar els valors entre crides.

Aquest projecte es compila i s'executa dins d'un entorn que proporciona dades de prova, cosa que facilita molt la validació. A més, he afegit instruccions de debug per entendre millor com evoluciona cada valor dins de la matriu.

Al següent vídeo parlaré de l'optimització d'aquestes funcions i dels resultats obtinguts.

Vídeo 4 - Optimització, proves i tancament de la pràctica

En aquest últim vídeo parlaré de l'optimització del codi i de com he validat el funcionament correcte de la segona fase de la pràctica. Després de traduir les funcions `avgmaxmin_city_s` i `avgmaxmin_month_s`, ha calgut revisar acuradament el rendiment i la precisió dels càlculs.

Per assegurar la correcció dels resultats, he utilitzat valors de test que contenen casos especials: valors zero, valors negatius, infinits i NaN. Això ha permès comprovar que les comparacions i càlculs es comporten com cal. Per exemple, en calcular la mitjana, s'ha de descartar qualsevol NaN i dividir només pels valors vàlids. Això es fa utilitzant la funció `E9M22_are_unordered_s`, que detecta si un valor és NaN.

També he optimitzat els bucles per evitar lectures innecessàries de memòria i he aplicat instruccions eficients per operar amb els valors flotants codificats. Un altre punt clau ha estat assegurar la correcta inicialització dels màxims i mínims, especialment en presència de valors especials.

A nivell de presentació final, he documentat bé el codi, indicant clarament el paper de cada rutina i afegint comentaris en català per facilitar la lectura. A més, he gravat aquest vídeo com a part de l'entrega per tal de demostrar el coneixement adquirit.

En resum, aquesta pràctica ha estat molt útil per entendre tant la representació de nombres com el tractament eficient de dades. He après a traduir lògica d'alt nivell a codi de baix nivell i a tenir en compte casos que normalment el compilador gestiona per nosaltres. Gràcies per la vostra atenció!