

Trabajo Práctico

Mini-Vm

R-222 Arquitectura del Computador

Consultas: fedebergero@gmail.com

Introducción

El objetivo de este trabajo es que el alumno aprenda los detalles de ejecución e implementación de una máquina abstracta.

La máquina posee las siguientes instrucciones:

- La máquina posee 8 registros de 4 bytes. Ellos son: ZERO, siempre vale cero. PC indica qué instrucción es la próxima a ejecutar (de la lista de instrucciones). SP Apunta al tope de la pila. R0,3 son registro de uso general y FLAGS contiene bits de estado (resultado de operaciones, comparaciones, etc).
- La máquina posee una única memoria en la cual se puede escribir y leer, mediante instrucciones MOV o POP/PUSH.
- El juego de instrucciones es el siguiente:
 - NOP No realiza ninguna operación.
 - MOV mueve un valor de/hacia registro/memoria/constante.
 - SW guarda el valor de src en la dirección de memoria dst.
 - LW carga en dst el valor de la memoria apuntada por src.
 - PUSH escribe el argumento src en el tope de la pila, decrementando el SP por 4.
 - POP escribe en el argumento dst el tope de la pila incrementando el SP por 4.
 - PRINT imprime por pantalla el entero descripto en el argumento src.
 - READ lee un entero y lo guarda en el argumento destino.
 - ADD/SUB suma/resta los argumentos src y dst y lo guarda en dst.
 - MUL/DIV multiplica/divide los argumentos src y dst y lo guarda en dst.
 - CMP compara src y dst y setea los bits correspondiente en el registro FLAGS.
 - JMP saltan a la instrucción número dst de la lista de instrucciones.
 - JMPE saltan a la instrucción número dst si la bandera de igualdad está seteada.

- JMWPL saltan a la instrucción número dst si la bandera de menor está seteada.
- HLT termina el programa.

Todos sus operando son enteros de 4 bytes

Metodología

Para realizar el trabajo el alumno debe resolver los siguientes puntos:

- Comprender la representación de las instrucciones y de la máquina abstracta.
- Implementar las instrucciones de la máquina abstracta.
- Desarrollar problemas de tests y validar sus resultados. Estos problemas deben ser representando como arreglos de instrucciones en C++. Por ejemplo:

```
Instruction code[3] = {
    Instruction(MOV,Operand(IMM,3),Operand(REG,R0)),
    Instruction(PRINT,Operand(REG,R0)),
    Instruction(HLT) };
```

representan el programa:

```
MOV $3, %R0
PRINT %R0
HLT
```

- Notar que en la página de la materia tiene un código a completar. Código

Problemas de tests

- El `test1` debe implementar un programa que imprima el valor absoluto de un entero leído.
- El `test2` debe implementar un programa que que cuente los bits de un entero leído y lo imprima por pantalla. Para este test deberá incluir una operacion AND que realice el and de bits de sus dos argumentos y lo guarde en dst.
- El `test3` debe implementar un programa que sume los enteros de un arreglo dado como argumento. Aquí puede asumir que los argumentos vienen en la pila, siendo (SP) el inicio del arreglo y 4(SP) el tamaño.

Características adicionales

El alumno puede extender el trabajo con las siguientes características adicionales (debe realizar al menos una)

La inclusión de nuevas instrucciones requieran recompilar el analizador utilizando bison y flex (ver el Makefile).

- Incluir instrucciones de operaciones de bits a la máquina abstracta
- Implementar las instrucciones CALL (que saltan a un número de instrucción guardando el PC en la pila) y RET (que salta al número de instrucción guardado en la pila). Con ellas puede desarrollar alguna función de ejemplo que pase argumentos por pila.

Entrega del Trabajo

El trabajo será evaluado por la cátedra mediante una presentación en computadora. El alumno debe entregar un informe de al menos dos páginas incluyendo datos académicos (integrantes del grupo, legajos, fechas) y reportando problemas y soluciones encontradas durante la realización del trabajo y posibles extensiones al mismo