|  |  |
| --- | --- |
|  | |
| 1. Describa la Taxonomía de Flynn. |
|  |  | |
|  | -SISD (SINGLE INSTRUCTION SINGLE DATA):Dispositivo que utiliza la arquitectura son los microprocesadores | |
|  | -SIMD (SINGLE INSTRUCTION MULTIPLE DATA):Dispositivo que utiliza la arquitectura es la GPU | |
|  | -MIMD (MULTIPLE INSTRUCTION MULTIPLE DATA):Dispositivo que utiliza la arquitectura es los sistemas aéreos | |
|  | -MISD(MULTIPLE INSTRUCTION SINGLE DATA):Dispositivo que utiliza la arquitectura es los sistemas distribuidos | |
|  |  | |
|  | 2. Diga cuales son los 4 principios de diseño. | |
|  |  | |
|  | 1-La simplicidad favorece la regularidad | |
|  | 2-Entre más pequeño es más rápido | |
|  | 3-Hacer el caso común más rápido | |
|  | 4-Buenos diseños de  mandan grandes compromisos | |
|  |  | |
|  | 3. Explique los tres formatos de instrucciones principales existentes en la arquitectura SPARCV8. | |
|  |  | |
|  | -FORMATO #1 Lo utilizan las instrucciones de CALL | |
|  | OP disp30 | |
|  |  | |
|  | -FORMATO #2 Lo utilizan las instrucciones SETHI y BRANCH (Bicc,FBfcc,CBcc) | |
|  | OP RD OP2 imm22 | |
|  | OP a COND OP2 disp22 | |
|  |  | |
|  | -FORMATO #3 Lo utilizan las instrucciones memory instructions,arithmetic.logical,shnct and remaining | |
|  | OP RD OP3 RS1 i UNSUED/ZERO RS2 | |
|  | OP RD OP3 RS1 i UNSUED/ZERO SHNCT | |
|  |  | |
|  | 4. Explique cómo inicializar un valor grande, que ocupe más de 13 bits, en la arquitectura SPARCV8. | |
|  |  | |
|  | -Se inicializa con la instrucción SETHI,la cual permite almacenar hasta 22 bits | |
|  |  | |
|  | 5. Como puedo reescribir la instrucción (OR y SUBcc) cuando inicializo y comparo 2 registros. | |
|  |  | |
|  | -Se puede reescribir con las instrucciones sintéticas MOV para OR y CMP para SUBcc. | |
|  |  | |
|  | 6. Implemente en lenguaje de bajo nivel y lenguaje de maquina el siguiente programa. | |
|  | 7. a. LENGUAJE DE BAJO NIVEL | |
|  | int main(){ | |
|  | int a = 8; MOV 8 , %L0 | |
|  | int c = 33; MOV 33, %L2 | |
|  | int b = -16800; SETHI -17,%L1 | |
|  | OR %L1,608,%L1 | |
|  | if((a+b)<=b\*32){ ADD %L0,%L1,%L3 | |
|  | SLL %L1,32,%L4 | |
|  | CMP %L3,%L4,%L4 | |
|  | BG a SALTO1 | |
|  | c=a+(b\*2); SLL %L1,2,%L6 | |
|  | ADD %L0,%L6,%L2 | |
|  | BA a EXIT | |
|  | } SALTO1 | |
|  | else {return b; MOV %L1,%O0 | |
|  | } EXIT | |
|  | return a+c; ADD %L0,%L2,%O1 | |
|  | } | |
|  |  | |
|  | LENGUAJE DE MÁQUINA | |
|  |  | |
|  | DIRECCIONES OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0000 10 10000 000010 00000 1 0000000001000 | |
|  | OP RD OP2 Imm22 | |
|  | OX0004 00 10001 100 1111111111111111101111 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0008 10 10001 00010 10001 1 0001001100000 | |
|  | OX000C 10 10010 000100 10010 1 0000000100001 | |
|  | OX0010 10 10011 000000 10000 0 00000000 10001 | |
|  | OX0014 10 10100 100101 10001 1 00000000 100000 | |
|  | OX0018 10 10101 010100 10011 0 00000000 10100 | |
|  | OP a cond OP2 disp22 | |
|  | OX001C 00 1 1010 010 0000000000000000000100 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0020 10 10110 100101 10001 1 00000000 00010 | |
|  | OP a cond OP2 disp22 | |
|  | OX0024 00 1 1000 010 0000000000000000000010 | |
|  | OP RD OP3 RS1 i Unusued/zero SHCNT Ó RS2 | |
|  | OX0028 10 01000 100101 00000 0 00000000 10000 | |
|  | OX002C 10 01001 000000 10000 0 00000000 01001 | |
|  |  | |
|  |  | |
|  |  | |
|  | b. | |
|  | int main(){ LENGUAJE DE BAJO NIVEL | |
|  | int a = 8; MOV 8,%L0 | |
|  | int b = -10; SETHI -1,%L1 | |
|  | OR %L1,1014,%L1 | |
|  | int c = 3; MOV 3,%L2 | |
|  |  | |
|  | if(a!=b){ CMP %L0,%L1,%L3 | |
|  | BE a SALTO1 | |
|  | return c/8; SRL %L2,3,%L2 | |
|  | MOV %L2,%O0 | |
|  | } BA a EXIT | |
|  | else{ SALTO1 | |
|  | return b; MOV %L1,%O1 | |
|  | } EXIT | |
|  | NOP | |
|  | } | |
|  | LENGUAJE DE MÁQUINA | |
|  |  | |
|  | DIRECCIONES OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0000 10 10000 000010 00000 1 0000000001000 | |
|  | OP RD OP2 Imm22 | |
|  | OX0004 00 10001 100 1111111111111111111111 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0008 10 10001 000010 10001 1 0001111110110 | |
|  | OX000C 10 10010 000100 00000 1 0000000000011 | |
|  | OX0010 10 10011 100101 10000 0 00000000 10001 | |
|  | OP a cond OP2 disp22 | |
|  | OX0014 00 1 0001 010 0000000000000000000100 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2/SHCNT | |
|  | OX0018 10 10110 100101 10001 1 00000000 00010 | |
|  | OX001C 10 10010 100100 10010 1 00000000 01000 | |
|  | OP a cond OP2 disp22 | |
|  | OX0020 00 1 1000 010 0000000000000000000010 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0024 10 01001 000010 00000 0 00000000 10001 | |
|  | OP RD OP2 Imm22 | |
|  | OX0028 00 00000 100 0000000000000000000000 | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  | c. | |
|  | int main(){ LENGUAJE DE BAJO NIVEL | |
|  | int a = -21180; SETHI -21,%L0 | |
|  | OR %L0,324,%L0 | |
|  | return a; MOV %L0,%O0 | |
|  | } | |
|  |  | |
|  | LENGUAJE DE MÁQUINA | |
|  |  | |
|  | DIRECCIONES OP RD OP2 Imm22 | |
|  | OX0000 00 10000 100 11111111111111111101011 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0004 10 10000 00010 10000 1 0000101000100 | |
|  | OX0008 10 1000 000100 00000 0 00000000 10000 | |
|  |  | |
|  |  | |
|  | d. | |
|  | int main(){ LENGUAJE DE BAJO NIVEL | |
|  | int a = 6; MOV 6,%L0 | |
|  | int b=4; MOV 4,%L1 | |
|  | int c=2; MOV 2,%L2  MOV 0, %L3 | |
|  | for(int i=0; i<=3; i++) FOR | |
|  | CMP %L4,3,%L5 | |
|  | { BG a SALTO1 | |
|  | c=(a+b)/8 | |
|  | SRL %L0,8,%L2 | |
|  | } BA a EXIT | |
|  | SALTO1 | |
|  | return c; MOV %L2,%O0 | |
|  | } EXIT | |
|  | NOP | |
|  |  | |
|  | LENGUAJE DE MÁQUINA | |
|  |  | |
|  | DIRECCIONES OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0000 10 10000 0000100 00000 1 0000000000110 | |
|  | OX0004 10 10001 0000100 00000 1 0000000000100 | |
|  | OX0008 10 10010 0000100 00000 1 0000000000010 | |
|  | OX000C 10 10101 100101 10100 1 0000000000011 | |
|  | OP a cond OP2 disp22 | |
|  | OX0010 00 1 1010 010 0000000000000000000100 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2/SHCNT | |
|  | OX0014 10 10000 000000 10000 0 00000000 10001 | |
|  | OX0018 10 10010 100100 10000 1 00000000 010000 | |
|  | OP a cond OP2 disp22 | |
|  | OX001C 00 1 1000 010 0000000000000000000010 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0020 10 01000 000010 00000 0 00000000 10010 | |
|  | OP RD OP2 Imm22 | |
|  | OX0024 00 00000 100 0000000000000000000000 | |
|  |  | |
|  | 8. Convierta el siguiente código a lenguaje de máquina | |
|  | Asignación de registros | |
|  | X=%i0 | |
|  | y=%i1 | |
|  | z=%i2 | |
|  | c=%L0 | |
|  | a=%L1 Lenguaje ensamblador | |
|  | EJEMPLO | |
|  | int ejemplo(int x, int y, int z){ SUB %i0,%i1,%i3 | |
|  | int a; SLL %i2,8,%i2 | |
|  | a = x - y + z\*8; ADD %i3,%i2,%l1 | |
|  | return a + 2; JMPL %O7,8,%g0 | |
|  | } ADD %L1,2,%O0 | |
|  |  | |
|  | int main(){ MAIN | |
|  | int x = 4, y = 2, z = -128; MOV 4,%i0 | |
|  | MOV 2,%i1 | |
|  | MOV -128,%i2 | |
|  | CALL EJEMPLO | |
|  | int c= 0; MOV 0,%L0 | |
|  | int c = ejemplo(x,y,z); ADD %L0,45,%O1 | |
|  | return c + 45; | |
|  | } | |
|  |  | |
|  | Para ingresar la inicialización del registro %i2 debo realizar complemento a dos y realizar lenguaje ensamblador con las siguientes instrucciones: | |
|  | SETHI 0,%i2 | |
|  | OR %i2,896,%i2 | |
|  |  | |
|  | LENGUAJE MAQUINA | |
|  | DIRECCIONES OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0000 10 11010 000010 11000 0 00000000 11001 | |
|  | OP RD OP3 RS1 i Unusued/zero SHCNT | |
|  | OX0004 10 11001 100101 11001 1 00000000 01000 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0008 10 10001 000000 11011 0 00000000 11010 | |
|  | OX000C 10 00000 111000 01111 1 0000000001000 | |
|  | OX0010 10 10000 000000 10001 1 0000000000010 | |
|  | OX0014 10 11000 000010 00000 1 0000000000100 | |
|  | OX0018 10 11001 000010 00000 1 0000000000010 | |
|  | OP RD OP2 Imm22 | |
|  | OX001C 00 11010 100 11111111111111111111111 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0020 10 11010 000010 11010 1 0001110000000 | |
|  | OP Disp 30 | |
|  | OX0024 01 00000000000000000000000000001000 | |
|  | OP RD OP3 RS1 i Unusued/zero RS2 | |
|  | OX0028 10 10000 000010 00000 1 0000000000000 | |
|  | OX002C 10 11001 000000 10000 1 0000000101101 | |
|  |  | |
|  |  | |
|  | 9.Implemente una función mul en lenguaje de alto nivel, lenguaje de bajo nivel SPARCV8 y lenguaje de máquina SPARCV8 que realice la multiplicación de dos enteros sin signo usando solo sumas. | |
|  |  | |
|  | int mul (int n1, int n2) | |
|  | { | |
|  |  | |
|  | int t = 0; | |
|  |  | |
|  | for (int i = 0; i < n2; i++) | |
|  | { | |
|  | t + = n1; | |
|  | } | |
|  | return t; | |
|  | } | |
|  |  | |
|  |  | |
|  | int main () | |
|  | { | |
|  |  | |
|  | int r=10, x=9, e=6; | |
|  |  | |
|  | While (e > 0) | |
|  |  | |
|  | { | |
|  | r = mul (r, x); | |
|  | e - - ; | |
|  | } | |
|  |  | |
|  | return r; | |
|  | }  Lenguaje de bajo nivel  Mul:  Mov0, %L0  Mov0, %L1  Ciclo:  Add %L0, %i0, %L0  Add %L1, %1, %L1  Cmp %L1, i1  Bl ciclo:  Mov %L0, O0  Ret %15  Main:  Mov 10, %L2  Mov 9, %L3  Mov 6, %L4  Ciclo1:  Mov %L2, %L0  Mov %L3, %L1  JMPL mul, %15  Mov %i0, %L2  Sub %%L4,1, %L4  Cmp %L4, 0  Bg ciclo1  Mov %L2, %O2 | |
|  |  | |
|  | 10.Implemente la función pot en lenguaje de alto nivel, lenguaje de bajo nivel SPARCV8 y lenguaje de máquina SPARCV8 que realice la potencia de dos números enteros sin signo realizando llamados a la función desarrollada en el punto 9. | |
|  |  | |
|  | int mul (int n1, int n2) | |
|  | { | |
|  |  | |
|  | int t = 0; | |
|  |  | |
|  | for (int i = 0; i < n2; i++) | |
|  | { | |
|  | t + = n1; | |
|  | } | |
|  | return t; | |
|  | } | |
|  |  | |
|  |  | |
|  | int suma sucesiva () | |
|  | { | |
|  |  | |
|  | int r=10, x=9, e=6; | |
|  |  | |
|  | While (e > 0) | |
|  |  | |
|  | { | |
|  | r = mul (r, x); | |
|  | e - - ; | |
|  | } | |
|  |  | |
|  | return r; | |
|  | } | |
|  |  | |
|  | int main() | |
|  | { | |
|  | int base=2; | |
|  | int potencia=4; | |
|  | int r=1; | |
|  | int i=0; | |
|  |  | |
|  | if (potencia == 0) | |
|  | { | |
|  | return r; | |
|  | } | |
|  | { | |
|  | while(i<potencia) | |
|  | { | |
|  | resultado = mul(r,base); | |
|  | i = i + 1; | |
|  | } | |
|  | } | |
|  |  | |
|  | return r; | |
|  | } | |
|  |  | |
|  |  | |
|  |  | |
|  | 11.Implemente una función pol en lenguaje de alto nivel, lenguaje de bajo nivel SPARCV8 y lenguaje de máquina SPARCV8 que calcule un polinomio de grado n, en donde los coeficientes del polinomio se encuentran almacenados en un vector k de n+1 posiciones, de la siguientes forma: | |
|  | pol(x,k,n)=i=0nkixi | |
|  |  | |
|  | En donde el ki se encuentra en la posición k[i] del vector k. | |
|  |  | |
|  | 12.Implemente una función fact en lenguaje de alto nivel, lenguaje de bajo nivel SPARCV8 y lenguaje de máquina SPARCV8 que calcule el factorial de un número entero sin signo | |
|  |  | |
|  | 13.Implemente una función divint en lenguaje de alto nivel, lenguaje de bajo nivel SPARCV8 y lenguaje de máquina SPARCV8,que calcule la división entera entre dos números enteros. | |
|  |  | |