**TALLER PARCIAL**

**I[0]-i[7] rg[27] rg[31] entrada**

**l[0]-l[7] rg[16] rg[23] locales**

**out[0], out[7] rg[8] ,reg[15] salida**

**g[0] ,g[7] global**

**2)Flynn se basa en el numero de instrucciones y de la secuencia de datos que la compuatdora utiliza para procesar informacion**

**Hay 4 tipo de computadores**

**SISD sigle instrucción sigle datis**

**SIMD sigle instrucción multiples datos**

**MISD multiple datos 1 instrucción**

**MIMD multiple datos multiple instrucción**

**3)los 4 principios son**

**1) la simplificida favorece la regularidad**

**2) entre mas pequeño mas rápido**

**3) hacer el caso común mas rápido**

**4) buenos diseños demandan grandes compromisos**

**4) los formatos**

**El formato uno se utiliza para funciones de llamados disp. 30 almacena el numero de saltos o desplazmaientos**

**El formato dos se utiliza para SETHI Y BRANCH en disp. 22 se almacena el numero de saltos o el numero**

**El formato 3 para instrucciones aritmetologicas**

**4)Explique como incializar un valor grande**

**1) se pone el # en binario se completan los 32 bits con 0 (si es negativo se invierte)**

**2)se suma el numero 1**

**3)al resultado se toman los 22 bits mas significativos**

**5)como se puede reescribir el mov**

**Or variable ,valor valriable**

**Mov valor,registro**

**6)**

**int main(){**

**int a = 8;**

**int b = ­16800;**

**int c = 33;**

**if((a+b)<=b\*32){**

**c=a+(b\*2);**

**}**

**else{**

**return b;**

**}**

**return a+c;**

**}**

**mov 8, %l0**

**sethi 17,%l1**

**or %l1,608,%l1**

**mov 33,%l2**

**add %l0,%l1,%l3**

**sll %l1,32,%l4**

**cmp %l3,%l4,%l5**

**bg a, else**

**sll %l1, 2,%l6**

**add %l0,%l6,%l2**

**ba a , exit**

**mov %l1,%o0**

**add %l0,%l2,%o1**

**-16800**

**00000000000000000100000110100000**

**11111111111111111011111001011111**

**1**

**11111111111111111011111001100000=608**

**1111111111111111101111**

**0000000000000000010000**

**1**

**0000000000000000010001**

**-17**

**7)** **mov 8, %l0**

**Sethi 17,%l1**

**Or %l1,608,%l1**

**Mov 33,%l2**

**Add %l0,%l1,%l3**

**Sll %l1,32,%l4**

**Cmp %l3,%l4,%l5**

**Bg a, else**

**Sll %l1, 2,%l6**

**Add %l0,%l6,%l2**

**Ba a , exit**

**Mov %l1,%o0**

**Add %l0,%l2,%o1**

**B.**

**Int main(){**

**Int a = 8;**

**Int b = ­10;**

**If(a!=b){**

**Return c/8;**

**}**

**Else{**

**Return b;**

**}**

**}**

**Mov 8,%l0**

**Mov -10,%l1**

**Cmp %l0,%l1**

**Be a,else**

**Srl%l2,8,l2**

**Ba exit**

**Mov %l1,%o0**

**Nop**

**C.**

**Int main(){**

**Int a = ­21180;**

**Return a;**

**}**

**-21180**

**101001010111100**

**11111111111111111010110101000011**

**1**

**11111111111111111010110101000100=324**

**1111111111111111101011**

**0000000000000000010100**

**1**

**0000000000000000010101**

**-21**

**Sheti -21,%l0**

**Or %l0,324,%l0**

**Mov %l0,o0**

**Int main(){**

**Int a = 6; int b=4;**

**For(int i=0; i<=3; i++){**

**C=(a+b)/8;**

**}**

**Return c;**

**}**

**Mov 6,%l0**

**Mov 4,%l1**

**Mov 0,%l2**

**-for cmp %l2,3**

**Bge a return**

**Add %l2,1,%l2**

**Add %l0,%l1,%l0**

**Srl %l0,8,%l3**

**Ba for**

**Mov %l3,o0**

**Nop**

**8.**

**Int ejemplo(int x, int y, int z){**

**Int a;**

**A = x ­ y + z\*8;**

**Return a + 2;**

**}**

**Int main(){**

**Int x = 4, y = 2, z = ­128;**

**Int c= 0;**

**Int c = ejemplo(x,y,z);**

**Return c + 45;**

**}**

**ejemplo{**

**sll %i2,8,%l1**

**sub %i0,%i1,%l2**

**add %l1,%l2,o0**

**jmp %o7,8,%g0**

**add o0,2,o1}**

**main{**

**mov 4,%i0**

**mov 2,%i1**

**mov -128,%i2**

**mov 0,%l0**

**coll ejemplo**

**add %l0,45,o0**

**}**

**9)int mul (int a,int b)**

**Cont=a;**

**For (int i=1 ; i<b;i++)**

**Cont =cont+a**

**Return cont**

**Main**

**Int x=1**

**Int z**

**Int y=2**

**If y=0**

**Return 1**

**Else**

**For (int i=1 ,i<=y,i++)[**

**Z=mult(x,y)**

**]]**

**Return z**

1. **Mov 1,%i0**
2. **Mov 2,%i2**
3. **Cmp %i2,0**
4. **BNE a else**
5. **Mov 1,%o0**
6. **Ba a EXIT**
7. **Else**
8. **Mov 1,%lo**
9. **Cmp %lo,i2**
10. **Bg a exit**
11. **Add %lo ,1,%lo**
12. **Call mul**
13. **//lamo function**
14. **Mov %i1,%l1**
15. **Mov 1,%l2**
16. **Cmp %l2,%i2**
17. **Bge a exit**
18. **Add %l1 1,%l2**
19. **Add %l1 %i1,%l1**
20. **Ba a exit**
21. **Exit**
22. **Jmpl %07,8,%go**
23. **Mov %l1,%03**

**10. int mul( int a, int b){ a=i0**

**int z=0; b=i1**

**for(int i=0; i<b; i++){ i=l1**

**z+=a z=l0**

**} x=i2**

**y=i3**

**return z; w=l3**

**} j=l4**

**int pot (int x, int y){**

**int w=x,y;**

**if(y==0){**

**return 1;**

**}else{**

**for(int j=1; j<=y; j++){**

**int w=pot(w,x);**

**}**

**return w;**

**}**

**}**

**int main(){**

**int a=2;**

**int b=2;**

**int x= pot (a,b)**

**return x;**

**}**

**mul**

1. **mov 0,%l0**
2. **mov 0,%l1**
3. **forcmp %l1,%i1**
4. **bg a, return**
5. **add %l1,1,%l1**
6. **add %l0,%i0,%l0**
7. **ba a, for**
8. **jmp %o7,8,%g0**
9. **mov %l0,%o0**
10. **pot**
11. **mov %i2,%l3**
12. **cmp %i3,0**
13. **bne, a else**
14. **mov %o7,%o6**
15. **jmp %o6,8,%g1**
16. **mov 1,%o1**
17. **else**
18. **for**
19. **mov 1 ,%l4**
20. **cmp%l4,%i3**
21. **bg, a nit**
22. **add %l4,1%l4**
23. **call pot**
24. **ba, a for**
25. **mov %o7,%o6**
26. **jmp %o6,8,%g0**
27. **mov %l3,%o2**
28. **main**
29. **mov 2,%i0**
30. **call pot**
31. **mov 2,%i1**
32. **mov %i2,%o3**
33. **nop**