

Universidad Simón Bolívar.
Arquitectura del Computador II
Actividad a desarrollar:

Responda las siguientes preguntas:

1. El SPARC carece de varias instrucciones que se encuentran generalmente en máquinas CISC. Algunas de ellas se simulan fácilmente usando el registro R0, que siempre vale 0 o un operando constante. Estas instrucciones simuladas se llaman "pseudoinstrucciones" y el compilador del SPARC las reconoce. Muestre como simular las pseudoinstrucciones siguientes con una única instrucción del SPARC. En todas ellas «fuen» y «dest» se refieren a registros. (Pista: un almacenamiento en R0 no tiene efecto)

- a) **MOV** fuen, dest
- b) **COMPARE** fuen1, fuen2
- c) **TEST** fuen1
- d) **NOT** dest
- e) **NEG** dest
- f) **CLR** dest
- g) **NOP**

2. Se tiene un procesador Sparc con 72 registros.

a. Indique cuantos procedimientos es posible anidar simultáneamente sin exceder la capacidad del banco de registros.

b. Cada vez que que el programa pase por el punto A indique el valor de los punteros de ventana en curso y de ventana guardada.

Programa Principal:

```
{  
....  
int n  
k=fact(6);  
....  
}
```

```
Int fact(int n)  
{  
if(n<=1)          /*Punto A*/  
    return(1);  
else  
    return(n*fact(n-1));  
}
```

Resolucion:

1.- Pseudoinstrucciones

a) MOV fuen, dest

dest \leftarrow fuen **OR** Ro
dest \leftarrow fuen **ADD** Ro
dest \leftarrow fuen **SUB** Ro (son válidas otras combinaciones de **OP** con Ro)

b) COMPARE fuen1, fuen2

Ro \leftarrow fuen1 **SUB** fuen 2

c) TEST fuen1

Ro \leftarrow fuen1 **OR** fuen1

d) NOT dest

dest \leftarrow dest **ORN** Ro

e) NEG dest

dest \leftarrow Ro **SUB** dest

f) CLR dest

dest \leftarrow Ro **ADD** Ro
dest \leftarrow Ro **OR** Ro
dest \leftarrow Ro **SUB** Ro

g) NOP

JMPL *siguiente linea*
Ro \leftarrow Ro **ADD** Ro

Son válidas demás operaciones de escritura sobre Ro, lo cual no tiene efecto ya que Ro esta cableado a cero)

2.- SPARC de 72 registros

a) Numero de procesos anidados

- 8 registros globales
- Cada procedimiento usa 24 registros
 - o 8 registros de entrada (salidas del procedimiento anterior)
 - o 8 registros locales
 - o 8 registros de salida (entradas del procedimiento siguiente)

En total se pueden anidar 3 procedimientos sin exceder la capacidad del banco de registros. Quedan sin utilizarse $72 - 64 = 8$ registros

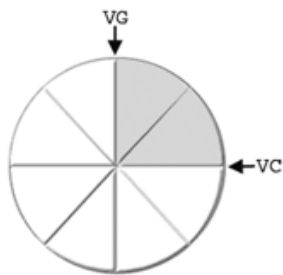


Total usado: $8 + 24 + 16 + 16 = 64$ registros en 3 procedimientos

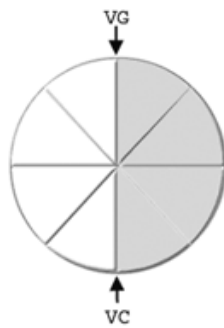
b) Ejecución de Factorial:

El código de la función "fact" es anidado 5 veces. A continuación se muestra la posición de los punteros "Ventana en curso" y "Ventana guardada"

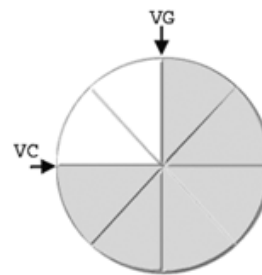
VC: Ventana en curso
VG: Ventana guardada



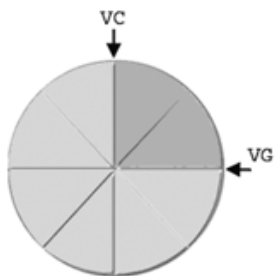
Fact (6)



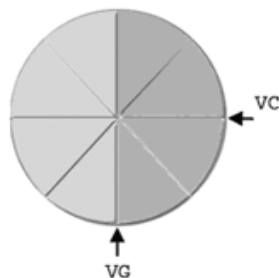
Fact (5)



Fact (4)



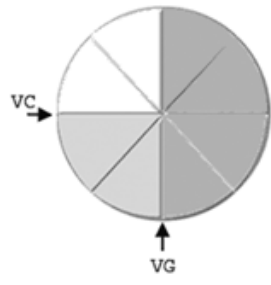
Fact (3)



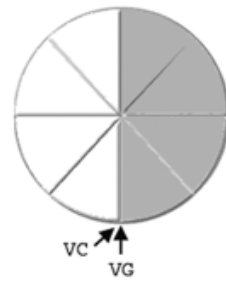
Fact (2)



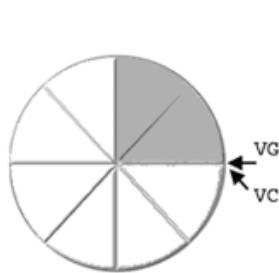
return (2)



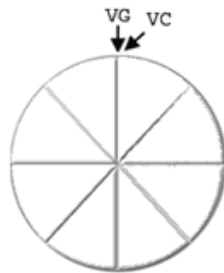
return (3)



return (4)



return (5)

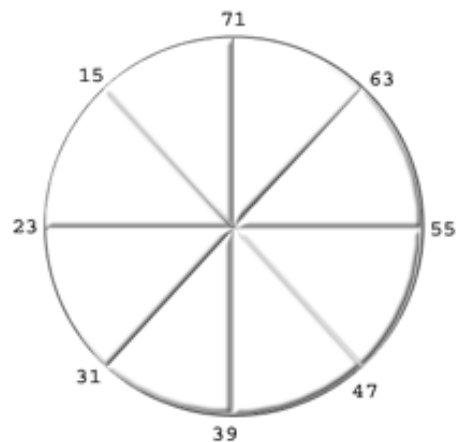


return (6)

Finalmente, para determinar el valor de los apuntadores **Ventana en curso** y **Ventana guardada**, se toma la siguiente referencia:

0 – 7 : registros reservados para variables globales

8 – 71: registros para los procedimientos (salida – locales – entradas)



Dado esto nos queda que los valores de los apuntadores son:

Procedimiento	Ventana en curso	Ventana guardada
fact (6)	55	77
fact (5)	39	77
fact (4)	23	77
fact (3)	71	55
fact (2)	55	39
return (2)	71	39
return (3)	23	39
return (4)	39	39
return (5)	55	55
return (6)	71	71

Nota: solo interesan para la solución las 5 primeras filas (procedimientos anidados)