

FINAL 14.02.2013

1_ Escriba una rutina en Assembly ARC que recibe a través de pila 2 valores enteros en complemento a 2. Luego verifica si su suma excede el rango representable en complemento a 2. Si la suma es representable, esta rutina devuelve un 0 por Stack y termina. En caso contrario, invoca otra rutina y luego termina devolviendo un 1.

Esta segunda rutina debe ser declarada en otro módulo y cumple la función de completar con 0's las 12 palabras siguientes a la dirección 9A0000h.

2_ Un programa en Assembly ARC es ejecutado en un momento en que la memoria de usuario se encuentra disponible a partir de la dirección 4000h. Un fragmento de ese programa es:

```
.org 2048
```

```
Y      .equ 10
```

```
Z      .equ 5
```

```
sumaParcial: 15
```

```
Inicio: sethi Y, %R1
```

```
        srl %R1, 10, %R1
```

Indique el contenido en binario de las posiciones de memoria a partir de 4000h, luego de que el programa inicie su ejecución.

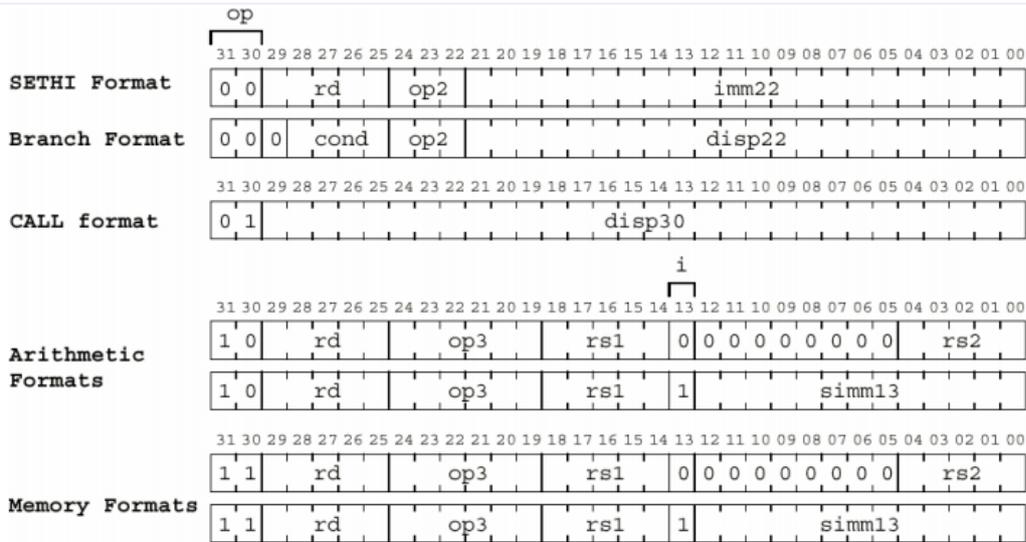
3_ Un proceso ARC está ejecutando la instrucción:

```
st %R2, %R10
```

Indique los componentes de la micro arquitectura que están involucrados y de qué manera la instrucción es ejecutada.

4_ Muchas rutinas de micro código se encuentran duplicadas en la memoria de control (Control Store). Justifíquelo. Indique de que manera esta repetición de código puede ser evitado.

La siguiente página contiene una imagen de ayuda, útil para el ejercicio 3.



| op | Format | op2 | Inst. | op3 (op=10) | op3 (op=11) | cond | branch |
|----|--------------|-----|--------|--------------|-------------|------|--------|
| 00 | SETHI/Branch | 010 | branch | 010000 addcc | 000000 ld | 0001 | be |
| 01 | CALL | 100 | sethi | 010001 andcc | 000100 st | 0101 | bcs |
| 10 | Arithmetic | | | 010010 orcc | | 0110 | bneg |
| 11 | Memory | | | 010110 orncc | | 0111 | bvs |
| | | | | 100110 srl | | 1000 | ba |
| | | | | 111000 jmp1 | | | |