

# Proyecto

## Fase IV: Demonstración de Operación de PPU para SPARC

### Tarea:

En esta fase demostrarán la operación del microprocesador mediante una simulación en Verilog del microprocesador ejecutando dos programas de prueba en código SPARC compatible con las especificaciones arquitecturales.

### Demostración:

Para la demostración de esta fase se les proveerán dos programas de prueba en un archivo de puro texto cada uno que deben utilizar para pre cargar la dos memorias del PPU. Cada archivo contiene bytes (en binario) separados por espacios o CR. El primer byte corresponde a la localización 0, el segundo a la 1, el tercero a la 2 y así sucesivamente hasta un eof (como los archivos de pre carga de memorias utilizados a las Fases I y III).

La simulación debe comenzar inicializando **Clk** en cero a tiempo cero. Entonces, debe cambiar de estado cada dos unidades de tiempo de manera perpetua. La señal **Reset** debe tener un valor de 1 a tiempo cero y cambiar a 0 en tiempo 1. La simulación debe culminar cuando el programa llegue a un loop infinito.

Su simulador debe mostrar (mediante una instrucción de **\$monitor**), el contenido de lo siguiente: **PC**, el address que recibe la memoria de data y el contenido de **r1, r3, r4, r5, r8, r10, r11 y r12**, todas en decimal. Todas las señales deben estar declaradas en una sola instrucción de **\$monitor**. Sin embargo, cuando sea requerido, su simulador debe tener la opción de mostrar las señales de control en las diferentes etapas y la instrucción que llega a la etapa de ID (todas en binario).

Para que se le facilite el monitoreo de señales deben instanciar e interconectar los componentes del PPU en el módulo de prueba. De esa manera cualquier señal de interconexión estará disponible para mostrarlas con una instrucción de monitor de ser necesario.

Una vez el programa de prueba finalice (llega a un loop infinito) deben imprimir el contenido de la memoria de data en binario. El contenido de la memoria lo deben imprimir en bytes separados por un espacio (cuatro bytes por línea).

### Programa de validación preliminar:

Antes de la demostración final de la operación del PPU es altamente recomendable que ejecuten el programa que se describe a continuación y se adjunta en como documento de esta fase del proyecto. Ya que su comportamiento es conocido, el programa les servirá como una herramienta de “debugging”. La ejecución del mismo no será un elemento de la calificación de esta etapa.

El programa lee un byte de la localización 56, un segundo byte de la localización 57 y un tercer byte de la localización 58. Si el primer byte es positivo el segundo byte y el tercer byte se suman

y el resultado se guarda en la localización 59. En la ejecución del programa se deben detectar cuatro data hazards. Estos hazards son atendidos por el mecanismo para detectar y manejar data hazards, por lo que no se deben perder ciclos por los mismos. El programa debe ejecutar sin pérdida de ciclos, ni siquiera por hazards de control.

Para demostrar que el programa trabaja correctamente deben mostrar (mediante una instrucción de `$monitor`), el contenido de lo siguiente: **PC** y el contenido de **r5, r6, r16, r17 y r18**, todas en decimal.

Una vez el programa de prueba finalice (llega a un loop infinito) deben imprimir el contenido del Word en la localización 56 en binario.

Instrucción	Assembly	Address	Hazard
10001010 00000000 00100000 00111000	add r0,56,r5	0	
11100000 01001001 01000000 00000000	lsb [r5,r0],r16	4	data
11100010 00001001 01100000 00000001	ldub [r5,1],r17	8	data
11100100 00001001 01100000 00000010	ldub [r5,2],r18	12	data
10001100 10000000 00000000 00010000	addcc r0,r16,r6	16	
00011100 10000000 00000000 00000101	bpos 5	20	
00000000 00000000 00000000 00000000	nop	24	
10001100 00100100 10000000 00010001	sub r18,r17,r6	28	
00010000 10000000 00000000 00000011	ba 3	32	
00000000 00000000 00000000 00000000	nop	36	
10001100 00000100 10000000 00010001	add r18,r17,r6	40	
11001100 00101001 01100000 00000011	stb r6, [r5,3]	44	data
00010000 10000000 00000000 00000000	ba 0	48	
00000000 00000000 00000000 00000000	nop	52	
11111100 00010011 00100000 00000000		56	

### Reglas de juego:

El/la estudiante que abandone el grupo en esta etapa recibirá una calificación de cero en la misma.

### Entrega:

- Subir a NEO un archivo comprimido con lo siguiente:
  1. Una portada identificando los/las integrantes del grupo (pdf).
  2. Versión final del diagrama de bloque del microprocesador (pdf).
  3. Código final de la implementación del circuito y su correspondiente módulo de prueba en Verilog
- De manera individual, cada miembro del grupo debe entregar una evaluación de su contribución al proyecto y de la contribución de cada uno de sus compañeros(as). **Sugiera**

**una calificación para cada uno de ellos(as).** Estas evaluaciones deben ser sometidas individualmente.

**Rúbrica de evaluación:**

- Se adjudicarán 36 puntos si se demuestra una operación correcta del primer programa de prueba.
- Se agregarán 9 puntos si se demuestra una operación correcta del segundo programa de prueba, siempre, y cuando el primer programa de prueba haya ejecutado correctamente.
- Se podrían adjudicar puntos parciales dependiendo del nivel de desarrollo del programa de simulación en caso de que uno o ninguno de los programas corra correctamente.