

Práctica de Laboratorio: Microprocesador MIPS

El objetivo de esta práctica es implementar el microprocesador MIPS (en su versión Multiciclo) en VHDL, cuyos detalles se pueden encontrar en: "Computer Organization and Design: The Hardware/Software Interface 3rd Edition", por David A. Patterson y John L. Hennessy. En concreto, se sigue el modelo detallado en el capítulo 5.

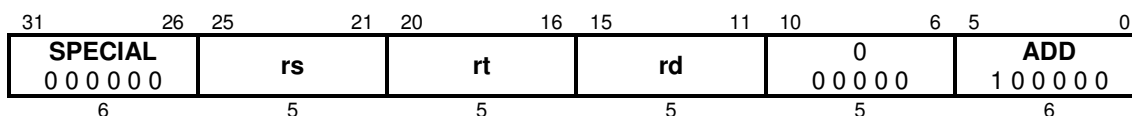
El modelo de las memorias de datos y programas proporcionado (archivo memory.vhd) no introduce ciclos de espera y responde en el mismo ciclo. Además, utiliza dos archivos separados para el contenido inicial de cada memoria, archivo llamado "program1" para memoria de instrucciones y "data" para memoria de datos. Se proporcionan un archivo de ejemplo (program1.s) para utilizar junto con el testbench de la práctica, si bien se pueden generar otros archivos correspondientes a otros códigos para hacer más pruebas.

Se pide implementar el microprocesador MIPS en su versión multiciclo.

No se pide que el microprocesador soporte todo el juego de instrucciones completo, sino solamente las siguientes instrucciones: *add*, *sub*, *and*, *or*, *lw*, *sw*, *slt*, y *beq*.

Se recomienda utilizar el esquema del libro, reflejado en la siguiente figura 1. Todos los registros deben resetearse asincrónicamente y funcionar por flanco de subida del reloj. Las instrucciones a implementar tienen los siguientes códigos de operación. Los códigos de operación se pueden encontrar también en el apéndice A del libro (volumen 3).

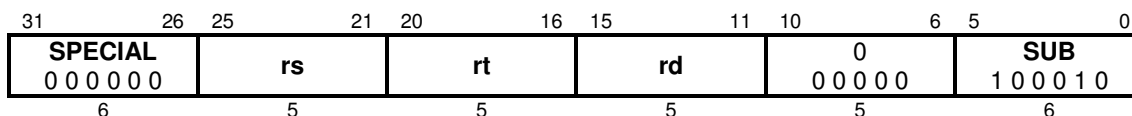
ADD (Add Word)



Formato: ADD rd, rs, rt

Descripción: $rd \leftarrow rs + rt$

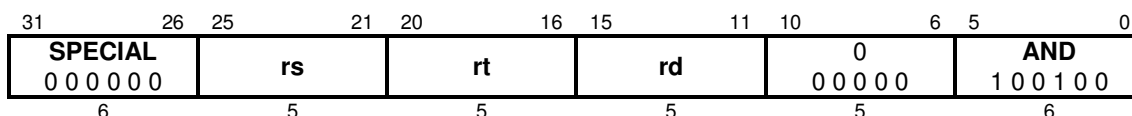
SUB (Subtract Word)



Formato: SUB rd, rs, rt

Descripción: $rd \leftarrow rs - rt$

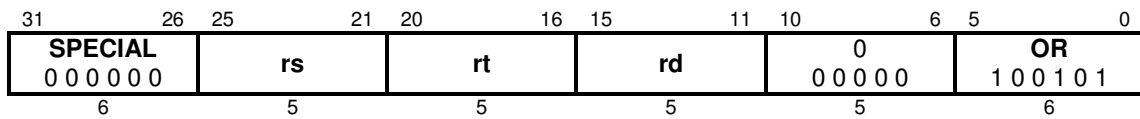
AND



Formato: AND rd, rs, rt

Descripción: $rd \leftarrow rs \text{ AND } rt$

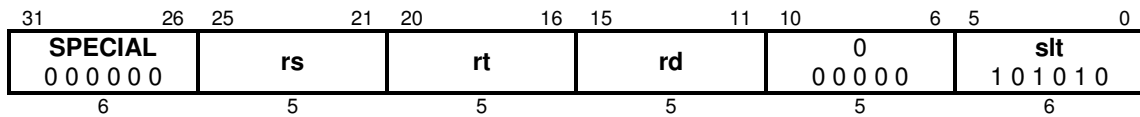
OR



Formato: OR rd, rs, rt

Descripción: $rd \leftarrow rs \text{ OR } rt$

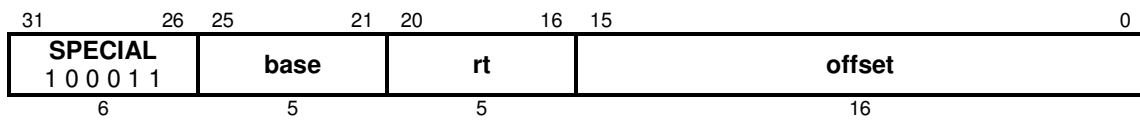
SLT (Set on Less Than)



Formato: SLT rd, rs, rt

Descripción: $rd \leftarrow (rs < rt)$

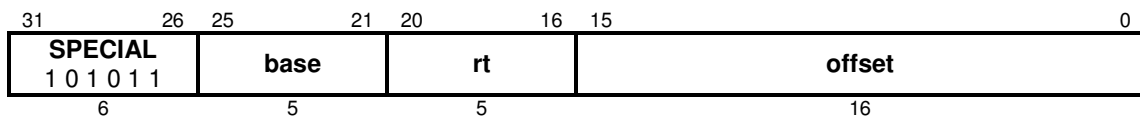
LW (Load Word)



Formato: LW rt, offset(base)

Descripción: $rt \leftarrow \text{memory}[\text{base} + \text{offset}]$

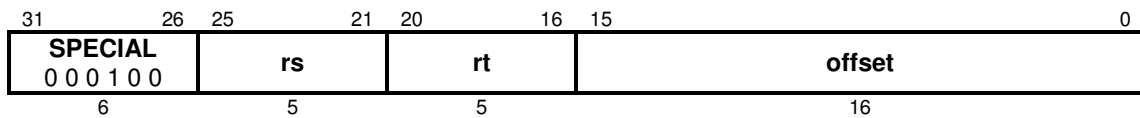
SW (Store Word)



Formato: ST rt, offset(base)

Descripción: $\text{memory}[\text{base} + \text{offset}] \leftarrow rt$

BEQ (Branch on Equal)



Formato: BEQ rs, rt, offset

Descripción: if $(rs = rt)$ then $PC \leftarrow PC + (\text{offset} \ll 2)$

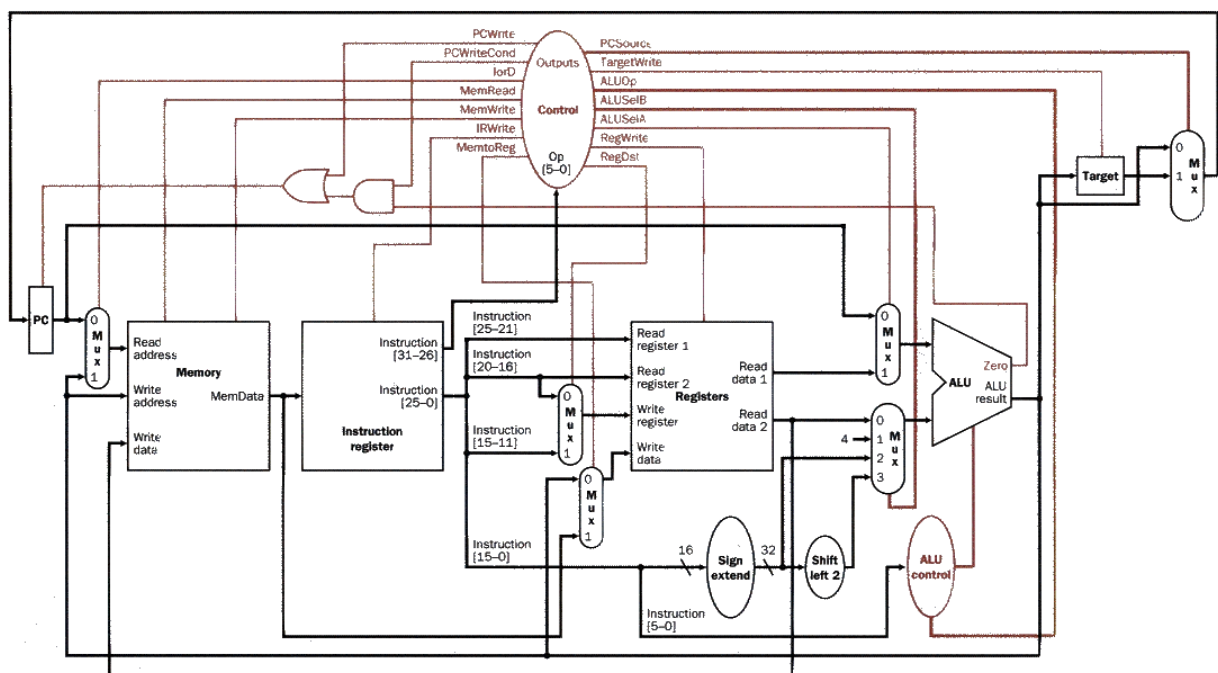


Figura 1. Modelo de microprocesador multiciclo.

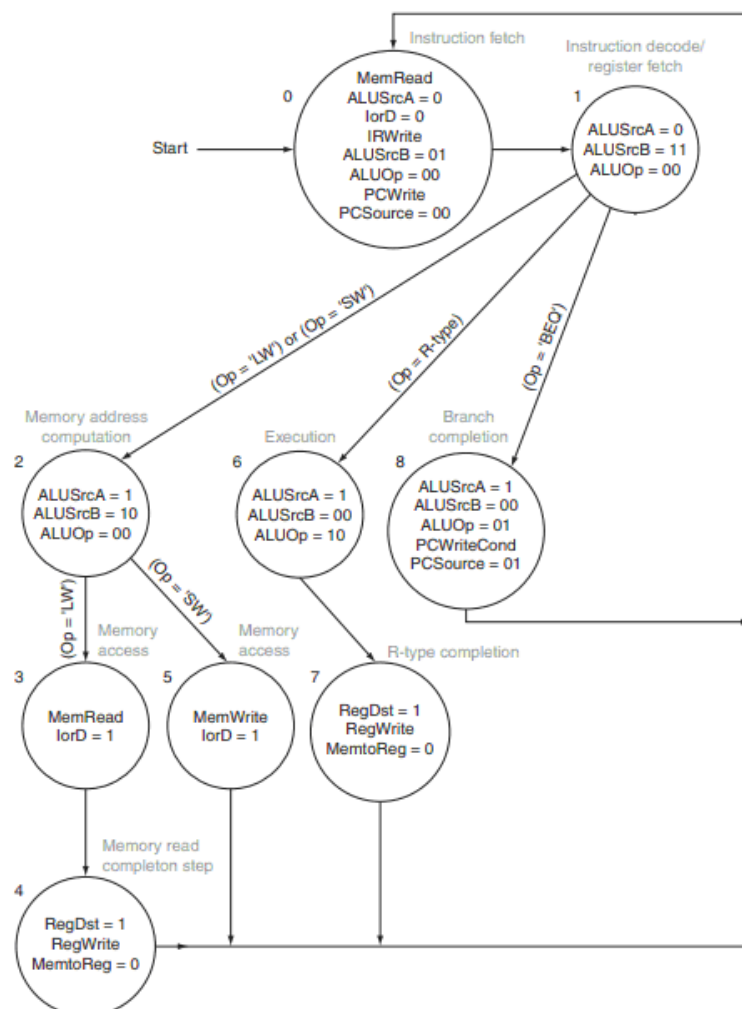


Figura 2. Unidad de control de microprocesador multiciclo.

Ejercicios

1. Implementar el diseño del procesador multiciclo, de acuerdo a la ruta de datos presentada en la Figura 1, y cuya unidad de control se corresponda con la Figura 2
2. Verificar el funcionamiento del diseño usando como programa de entrada "Program1" y el archivo de datos "data" proporcionado por la cátedra
3. Verificar el programa utilizando la versión del MIPS segmentado provista por la cátedra.
4. Comparar ambas soluciones, respecto: frecuencia máxima de operación, CPI estático y dinámico para el programa dado, tiempo de ejecución total de program1, y ocupación de area.

Material a entregar

- Archivos VHDL
- Informe conteniendo decisiones de diseño en la implementación propuesta, y los resultados de la comparación entre ambas versiones del procesador MIPS
- Archivos de memoria de instrucción y de datos

Ayudas y avisos

- Los archivos "programa" y "datos" que contienen las memorias de instrucciones y datos respectivamente deben estar en el directorio de trabajo. Si no es así, en el archivo procesador_TB.vhd se puede dar la ruta completa de dichos archivos cambiando las líneas de código:
- Si se implementa una arquitectura Von Neumann, se debe establecer las rutas de memoria para los 2 archivos (instrucción y datos), en sus valores genéricos C_ELF_FILENAME_LOW y C_ELF_FILENAME_HIGH. El segundo de los archivos comenzará a cargarse a partir de la dirección establecida en el parámetro genérico C_MEM_HIGH_OFFSET. En el caso que la arquitectura sea Harvard, el nombre del archivo a cargar en la parte alta debe descartarse (C_ELF_FILENAME_HIGH = "");
- El programa de prueba "program1.s" proporcionado en la práctica no incluye riesgos y prueba todas las instrucciones del ejercicio básico. El archivo "programa" es el resultado del ensamblado del "program.s" que se usará para probar el ejercicio básico. El archivo "datos" contiene los datos que se usaran por el "programa" en el ejercicio básico.
- El archivo *memory.vhd* contiene la memoria que se usará en el ejercicio básico. El archivo *Pipelined_MIPS.vhd* contiene la descripción completa de la entidad del procesador en su versión segmentada, y el archivo *Pipelined_MIPS_tb.vhd* contiene el testbench del mismo.
- El archivo *Multicycle_MIPS.vhd* debe completarse con la descripción de la arquitectura solicitada, y el archivo *Multicycle_MIPS_tb.vhd* contiene toda la descripción necesaria para realizar la verificación del procesador en su versión multiciclo.
- La tabla contenida en el archivo "registers.html" proporcionada en la práctica muestra la traducción de los nombres de registros usados en ensamblador al número de registro en el micro, del 0 al 31.