# Trabajo Práctico 10

Procesador TDA 1819

1. Implemente las siguientes instrucciones propuestas para el procesador “TDA 1819":

|  |  |  |
| --- | --- | --- |
| **Tipo de instrucción** | **Instrucción** | **Comentario** |
| **Aritmética** | dsubi *rd*, *rf*, *N* | Resta el valor inmediato *N* al registro *rf*, dejando el resultado en el registro *rd* (valores con signo) |
| **Lógica** | xnorr *rd*, *rf*, *rg* | Realiza un XNOR entre los registros *rf* y *rg* (bit a bit), dejando el resultado en el registro *rd* |

Para poder implementar estas operaciones primero definí la operación junto a su código único en el archivo repert\_cpu.vhd, ubicado en la carpeta Packages. Para la instrucción dsubi la ubique debajo de la operación dsub en la sección de operaciones aritméticas y le asigne el siguiente numero de operación 2C ya que era el siguiente disponible a la ultima operación aritmetica. Para la xnorr la ubique debajo de la operación xorr en la sección de operaciones logicas. Y le asigne el numero de operación 37 (hexadecimal) por la misma razón que al dsubi.

Lo siguiente que hice fue declarar el nombre y el tamaño asociado a cada código en el archivo const\_ensamblador.vhd, los agregue en las mismas posiciones que en el repertorio y le asigne un tamaño de 8 bytes al dsubi por ser una operación con un inmediato y de 5 al xnorr por ser una operación lógica entre registros.

Luego debo declarar los pasos a realizar al momento de decodificarse la operación en el archivo decode.vhd en la ubicación Usuario\PC\2. CPU\2. Etapas\2. Decode. Para ello utilice como modelo la operación DADDI para el DSUBI, ya que ambos son una operación aritmética que utiliza un valor inmediato. Lo que le hice fue cambiar la operación de alu que se ejecuta de un EX\_ADD a un EX\_SUB. Para que realice una resta en lugar de una suma.

Para la operación xnorr no fue tan sencillo porque la operación xnor no estaba declarada en el repertorio de ejecución de la alu. Primero al igual que en el dsubi utilice otra operación como modelo para el archivo decode.vhd, en este caso XORR ya que era una operación lógica entre registros. Y el cambio fue pasar del EX\_XOR a un EX\_XNOR que por el momento no existía en la cpu. Para agregar esta operación primero fui al archivo const\_cpu.vhd ubicado en la carpeta Packages y agregue la constante EX\_XNOR debajo de EX\_XOR. Luego de esto, para que funcione, tuve que asignarle el valor en orden después de esta (15) y correr todos los valores de las operaciones siguientes.

Por ultimo tuve que declarar el comportamiento de la operación EX\_XNOR en el archivo execute\_alu.vhd ubicado en Usuario\PC\2. CPU\2. Etapas\3. Execute\1. ALU. El comportamiento que declare fue Ures := Uop1 xnor Uop2; cuando la operación es EX\_XNOR.

1. Desarrolle un programa de prueba en el lenguaje Assembler tal que haga uso de las instrucciones previamente implementadas y pueda ser ejecutado en el TDA 1819.

Para ello, en primer lugar, deberá tener en cuenta estas consideraciones:

* 1. Se utilizarán valores con signo de 16 bits cada uno.
  2. A partir de su número de legajo, establezca el máximo valor correspondiente al mismo que resulte representable en este sistema binario. Incluya el dígito verificador en caso de ser posible. Por ejemplo, si su legajo fuera 00615/3, entonces el valor máximo sería 6153. En cambio, si fuera 09365/7, sería 9365.
  3. Aplique sobre su número de documento un mecanismo similar al descrito en el inciso anterior. Aquí el objetivo consiste, más precisamente, en seleccionar y preservar los cuatro o cinco dígitos menos significativos del susodicho documento según corresponda a fin de obtener el máximo valor expresable en este sistema binario. Por ejemplo, si su documento fuera 38283239, entonces el valor máximo sería 3239. Por su parte, si fuera 39431491, sería 31491.
  4. En caso de que los valores obtenidos en los apartados b. y c. resulten iguales entre sí, réstele una unidad al número calculado en el punto b.

A continuación, se presentan las tareas que habrán de realizarse por el programa solicitado en Assembler:

* 1. Calcular la diferencia entre los dos valores determinados en los incisos anteriores. Luego, efectuar nuevamente la operación aritmética intercambiando las posiciones de dichos operandos.
  2. Para cada cálculo llevado a cabo en la tarea previa, si el resultado hubiera sido negativo, realizar la operación lógica XNOR entre este último y el valor binario 10101110. En caso contrario, llevar a cabo esta misma operación utilizando como segundo operando el número hexadecimal B5C9.
  3. Almacenar en la memoria principal de la computadora el resultado de cada operación aritmético/lógica efectuada en los puntos e. y f.

1. Al utilizarse valores con signo de 16 bits con signo el valor máximo posible para representar en el sistema es 32768.
2. Mi numero de legajo es 02887/7. Por lo que el máximo numero representable en este sistema es 28877.
3. Mi numero de documento es 44356213. Utilizando los 4 dígitos menos significativos llego al máximo numero representable en este sistema, 6213. Si tomara el 5to digito el numero resultaría muy grande, 56213.

.data Posición en memoria:

A: .hword 28877 1000

B: .hword 6213 1002

C: .hword 0xB5C9 1004

D: .hword 0xAE 1006

E: .hword 0 1008

F: .hword 0 100A

G: .hword 0 100C

H: .hword 0 100E

.code CodOp: Accion:

lh r1, A(r0) 06 r1 🡨 A =28877 (0x70CD)

lh r3, C(r0) 06 r3 🡨 B =0xB5C9

lh r6, B(r0) 06 r6 🡨 C =6213 (0x1845)

lh r8, D(r0) 06 r8 🡨 D =0xAE

dsubi r2, r1, 6213 2C

dsubi r7, r6, 28877 2C

xnorr r4, r2, r3 37

xnorr r9, r7, r8 37

sh r2, E(r0) 07

sh r7, G(r0) 07

sh r4, F(r0) 07

sh r9, H(r0) 07

halt 81

1. Una vez simulada la ejecución en el TDA 1819 del programa desarrollado en el ejercicio precedente, resuelva las consignas finales seguidamente definidas:
   1. Determine el tiempo de ejecución del programa con la segmentación del cauce del procesador habilitada y deshabilitada.
   2. Identifique las instrucciones del programa y, en particular, la etapa de sus respectivos ciclos de ejecución en las cuales se efectúan en la ALU del TDA 1819 cada una de las operaciones aritmético/lógicas especialmente solicitadas para dicho programa. Indique la duración y los instantes de inicio y fin de la etapa involucrada en cada caso (con la segmentación del cauce del procesador habilitada y deshabilitada).

La entrega del presente trabajo práctico involucra los siguientes requerimientos:

1. Un informe detallado que contenga toda la información que juzgue pertinente para la resolución de los problemas aquí planteados.
2. Los archivos VHDL (extensión “.vhd”) del proyecto “TDA\_1819” actualizados con todos los cambios que le resultaron necesarios para implementar las instrucciones propuestas en el ejercicio 1. No se requiere enviar el espacio de trabajo o workspace completo, alcanza con especificar en el informe los nombres de los ficheros modificados junto con las rutas en las cuales deberán ser ubicados dentro del proyecto original para la correcta simulación del procesador.
3. El nuevo programa desarrollado en el lenguaje Assembler tal como es solicitado en el ejercicio 2 (extensión “.asm”). Se recomienda asimismo en este caso incluir en el informe la ruta del mencionado programa en el proyecto “TDA\_1819” para evitar cualquier inconveniente que pudiera presentarse durante el ensamblaje y/o la ejecución del mismo.



# Fecha límite de entrega por mensajería de IDEAS: viernes 24 de noviembre de 2023 a las 11:59:59 hs. (mediodía)