

# DISEÑO E IMPLEMENTACIÓN DE UN MICROPROCESADOR CON ARQUITECTURA SEGMENTADA EN FPGA

José Alberto Díaz García, Adolfo Méndez Madrigal, Miguel Angel Aguilar Ulloa

Escuela de Ingeniería Electrónica / Instituto Tecnológico de Costa Rica

www.ie.itcr.ac.cr

jdiaz@itcr.ac.cr , amendez@ietec.org, maguilar@ietec.org

## SUMARIO

Se presenta el proceso realizado para segmentar un procesador basado en una arquitectura multiciclo desarrollada en el ITCR, para su implementación en un FPGA utilizando lenguaje de descripción de hardware Verilog, con el objetivo de mejorar su rendimiento al ejecutar un conjunto de instrucciones dado.

## 1. INTRODUCCIÓN

Se presenta el diseño y la implementación de un microprocesador con arquitectura segmentada, utilizando el lenguaje de descripción de hardware, Verilog, partiendo de un microprocesador con arquitectura multiciclo (AMM) [1]. Se muestran los pasos seguidos en la evolución de la arquitectura multiciclo, desde el punto de vista del hardware hasta el rediseño de las instrucciones.

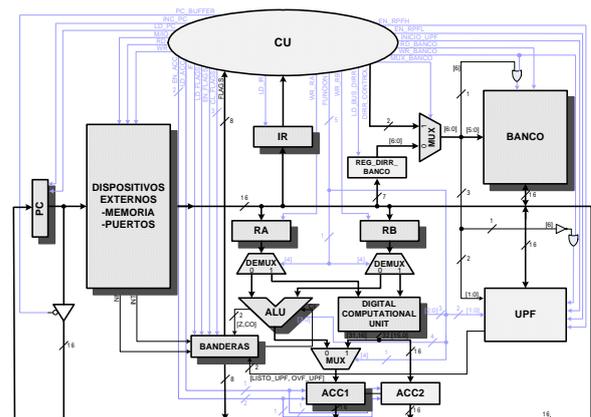
La metodología seguida fue fundamental en el hecho de que se adquirió destrezas en el diseño de arquitecturas para microprocesadores, así como en el desarrollo de este tipo de módulos para formular otras arquitecturas.

## 2. MICROPROCESADOR CON ARQUITECTURA MULTICICLO

Para la realización de la arquitectura segmentada se utilizó como punto de partida la arquitectura multiciclo diseñada en [1], mostrada en la **Fig. 1**. Dicha arquitectura, basada en el modelo de Von Neumann, ejecuta las instrucciones CISC.

Los módulos en el etapa de ejecución son: una unidad lógico aritmética con su correspondiente registro de banderas (ALU), una unidad computacional digital (DCU) que realiza funciones aritméticas de manera combinacional (multiplicación, división, potencia, raíz, logaritmo y antilogaritmo) adicionales a las que ejecuta la ALU, dos registros temporales en la entradas de la ALU para mantener estables los datos (RA y RB), dos registros acumuladores para almacenar temporalmente los resultados de 16 bits cada uno, en donde el segundo se utiliza para manejar los 32 bits provenientes de la unidad computacional digital (ACC1 y ACC2), una unidad de punto flotante (UPF); en la etapa de

almacenamiento y dispositivos externos cuenta con un contador de programa (PC) para el acceso a la memoria principal, un buffer ubicado a la salida del PC que es utilizado para establecer una ruta para que el valor del PC se respalde en el banco de registros sin riesgos de colisión, dispositivos externos tanto una memoria como los puertos de entrada/salida; para el almacenamiento interno la arquitectura cuenta con un banco de 64 registros de 16 bits.



**Fig.1.** Arquitectura de un microprocesador multiciclo.

Todos estos módulos son gobernados por una unidad de control basada en una máquina de estados microprogramada para la decodificación de las instrucciones (CU).

## 3. DESCRIPCIÓN DE LAS INSTRUCCIONES

Para la implementación de esta arquitectura se utilizaron las instrucciones MIPS las cuales se dividen principalmente en dos tipos: R e I. Las instrucciones tipo R son utilizadas para operaciones aritméticas y de movimiento de datos entre registros, las instrucciones tipo I se utilizan para operaciones que involucren la memoria de datos ya que para acceder a la misma se utilizara un direccionamiento indexado, además se utilizaran instrucciones tipo I para operaciones aritméticas en las cuales uno de sus operándos sea una constante.

#### 4. SEGMENTACIÓN DE LA ARQUITECTURA

Con la segmentación se logró aislar entre sí las diferentes etapas que se ocupan para ejecutar instrucciones, para así traslaparlas aprovechando el paralelismo de las instrucciones secuenciales. En este sentido se deben definir las etapas que corresponden a cada parte de la ejecución de las instrucciones. En cada una de las etapas se debe distribuir todo el hardware de la arquitectura multiciclo.

En principio la meta es segmentar la arquitectura multiciclo, para tal efecto se utilizaron los principales bloques y en un primer momento se reordenaron de tal manera que se obtuvo una arquitectura uniciclo. A partir de ésta finalmente se llega a la arquitectura segmentada. En la **Fig. 1** los módulos sombreados indican los que se utilizaron, estos son: el banco de registros, la unidad lógico aritmética, la unidad computacional digital, el registro de instrucción IR, los registros de entrada a la ALU (RA, RB), el acumulador (ACC), el contador de programa (PC), y los dispositivos externos.

La unidad lógico aritmética y la unidad computacional digital permanecen iguales respecto de los utilizados en la arquitectura multiciclo. Los registros de instrucción, de entrada a la ALU y el acumulador se incorporaron dentro de los registros de segmentación. En el caso de los módulos que requieren ciertas modificaciones se encuentran: separación de memoria de datos y de instrucciones, separación del contador de programa de la ALU, descentralización del control, acceso simultáneo en el banco de registros, lógica de control para la ejecución de las MIPS y los registros de segmentación.

##### 4.1. Separación de memoria de datos y de instrucciones

Con respecto a los dispositivos externos se sabe que corresponden a la memoria y a los puertos de entrada/salida. La memoria principal, dentro del contexto de la arquitectura multiciclo, contiene tanto las instrucciones como los datos esto acorde con el modelo Von Neumann, sin embargo, para la implementación de la arquitectura uniciclo y con miras a la segmentada se hace necesario separar la memoria en dos bancos correspondientes a las instrucciones y a los datos, es decir, se busca una arquitectura Harvard. El motivo de este cambio se fundamenta en la necesidad que presenta la arquitectura segmentada en donde se accede a la memoria en busca de instrucciones y datos de manera simultánea.

##### 4.2. Separación del contador de programa de la ALU

Se hizo necesario separar el contador de programa de la ALU para realizar cálculos aritméticos al mismo tiempo que se calcula el valor del contador de programa.

##### 4.3. Descentralización del control

La arquitectura multiciclo usa una unidad de control microprogramada, esta estructura no se puede utilizar para el procesador segmentado debido a que se trata de realizar un proceso del ciclo de ejecución cada ciclo de

reloj, para lograr esto se optó por dividir el CU en 3 módulos que son:

- Control: se encarga de decodificar las señales de control necesarias para la ejecución de la instrucción se ubica en la etapa de decodificación de instrucción (ID).
- Branch: unidad encargada de evaluar las condiciones de salto, esta ubicada en la etapa acceso a memoria (MEM)
- Control\_ex: Se ubica en la etapa de ejecución (EX) y es el encargado de generar las señales de control para la ALU y la DCU.

Estas modificaciones hechas al CU permiten la ejecución de varias etapas de las instrucciones al mismo tiempo ya que las señales de control para cada etapa de la ejecución de la instrucción se generan en la etapa de ID y se propagan a través del microprocesador, además señales extra necesarias para la ejecución de las instrucciones se generan en las etapas donde son necesarias.

##### 4.4. Acceso simultáneo en el banco de registros.

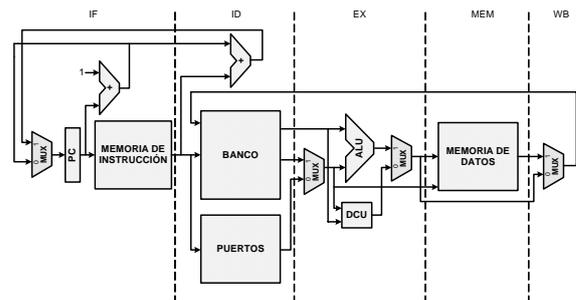
El banco de registros se debe modificar de tal manera que permita la lectura de dos registros y escritura de uno de manera simultánea, esto, debido a que el banco que utiliza la arquitectura multiciclo permite solo un acceso a la vez a cada uno de los 64 registros. Esta modificación se hace necesaria debido a que en la arquitectura segmentada se debe buscar cada operando de manera simultánea y separada, además paralelamente se pueden estar escribiendo en otros registros.

##### 4.5. Lógica de control para la ejecución de MIPS.

Debido a la utilización de MIPS para este microprocesador es necesario agregar en la etapa de decodificación de instrucción una lógica de control, ya que los bits del operando y destino pueden variar de posición además se necesita un módulo de extensión de signo debido a que en las MIPS las constantes son de 16 bits y la ruta de control es de 32 bits.

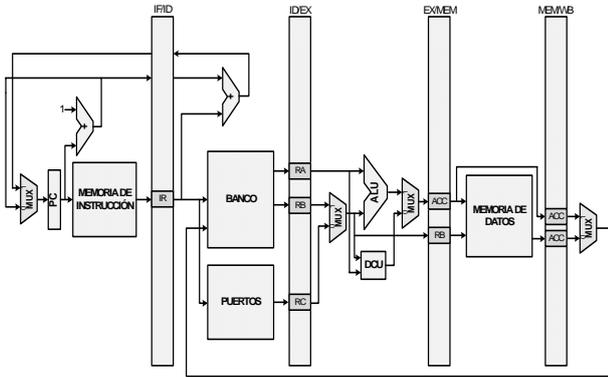
##### 4.6. Registros de segmentación.

Durante la ejecución de las instrucciones en cada una de las etapas se hace necesario mantener datos de cada segmento de la instrucción, esto se logra mediante la ubicación de registros entre cada una de las etapas ubicados en las líneas punteadas de la **Fig. 2**. Las instrucciones avanzan de etapa en etapa en cada un ciclo de reloj de un registro de segmentación al siguiente.



**Fig.2.** Arquitectura uniciclo

El hardware que se encuentra incorporado en cada etapa es combinacional y el máximo tiempo de retardo determinará la frecuencia máxima de operación de la arquitectura. En la **Fig. 3** se muestra la arquitectura segmentada básica con los correspondientes registros de segmentación, en los cuales se incorporan los registros tomados de la arquitectura multiciclo que son: el registro de instrucción (IR) los registros temporales de los operandos (RA, RB) y el registro acumulador (ACC).



**Fig. 3.** Incorporación de registros de segmentación

## 5. ARQUITECTURA RESULTANTE

Una vez realizadas las modificaciones del procesador multiciclo se obtiene una estructura uniciclo, posteriormente una arquitectura segmentada mostrada en la **Fig. 4**, la cual constituye la base para la arquitectura segmentada diseñada y a continuación se describe este proceso.

### 5.1. Segmentación de la arquitectura uniciclo

Con los módulos obtenidos de la arquitectura multiciclo y el ajuste correspondiente a los mismos se llegó a la arquitectura uniciclo que se muestra en la **Fig. 2**, esta representa una primera aproximación hacia la arquitectura segmentada.

En esta arquitectura se muestra como se separó la memoria de instrucciones así como la de datos, además en ella se señala cada una de las etapas de la segmentación, y como en cada una se agrupó el hardware correspondiente que se tomó de la arquitectura multiciclo, el mismo se distribuyó de la siguiente manera:

- En la etapa de búsqueda de instrucción (IF) se colocó el contador de programa, este direcciona la memoria de instrucción, su valor puede ser incrementado o modificado por un salto.
- En la etapa de decodificación de instrucción, búsqueda de operando y calculo de dirección (ID) se ubicó el banco de registros y los puertos, además se agregó un sumador el cual determina la dirección de salto.
- En la etapa de ejecución (EX) se ubicó la unidad lógico aritmética así como la unidad computacional digital, además de la lógica para variar la ruta de datos hacia las unidades matemáticas.

- En la etapa de almacenamiento (MEM) se incorpora la memoria de datos la cual es direccionada por el resultado proveniente de la ALU o bien del DCU.
- Retroalimentación (WB) en esta etapa se guardan los datos provenientes de la unidad de ejecución o de la de almacenamiento en el banco de registros.

En la **Fig. 3** se muestra la incorporación de los registros de segmentación a la arquitectura multiciclo, en donde cada uno de estos registros separa las diferentes etapas de segmentación.

La unidad de control utilizada es básicamente un decodificador ubicado en el segmento de decodificación de instrucción, a partir de ahí las señales se propagan a las otras etapas de la arquitectura, además se cuenta con una unidad de control para generar las señales de la ALU y DCU y para evaluar las condiciones de salto ubicadas en la etapa de ejecución y escritura de memoria respectivamente.

## 6. RIESGOS DE LA ARQUITECTURA

Al utilizar segmentación de la arquitectura ofrece sus ventajas en la ejecución de un conjunto de instrucciones, no así en una única instrucción, por lo que se debe tomar en cuenta los riesgos que conlleva la ejecución al mismo tiempo de varias instrucciones utilizando los diferentes segmentos de la arquitectura del microprocesador.

Los principales conflictos se presentan al ejecutar instrucciones con operandos en común y con aquellas relacionadas con saltos condicionales e incondicionales.

### 6.1. Riesgos estructurales

Los riesgos estructurales se presentan cuando el hardware no puede manejar la combinación de instrucciones que deben ejecutarse en el mismo ciclo de reloj [2] (suponiendo que se cuenta con una única memoria). Por ejemplo: si una instrucción se encuentra en la etapa de búsqueda de instrucción y otra diferente en la etapa de almacenamiento no podrán ejecutarse en el mismo ciclo ya que ambas necesitan acceder datos de la memoria. También se presenta un problema con el banco de registros, cuando un dato debe ser leído y escrito en el mismo instante al correr una secuencia de instrucciones. Para solucionar este problema el banco de registros se escribe en la primera mitad del ciclo de reloj y la lectura se da en la segunda mitad. Además se separó la memoria de programa de la memoria de datos, esto debido a que la memoria de programa es continuamente leída, si la memoria contiene datos e instrucciones se tendría que introducir una burbuja en la ejecución del programa cada vez que se desea escribir o leer datos de la memoria lo cual es ineficiente.

### 6.2. Riesgo en la ruta de los datos

Los riesgos de datos se presentan cuando se desea utilizar un dato existente en un registro que fue modificado en la instrucción anterior, este problema se presenta cuando cualquiera de los operandos de la instrucción siguiente a ejecutar es igual al destino de la instrucción que se esta ejecutando y la señal de escritura del banco de registros

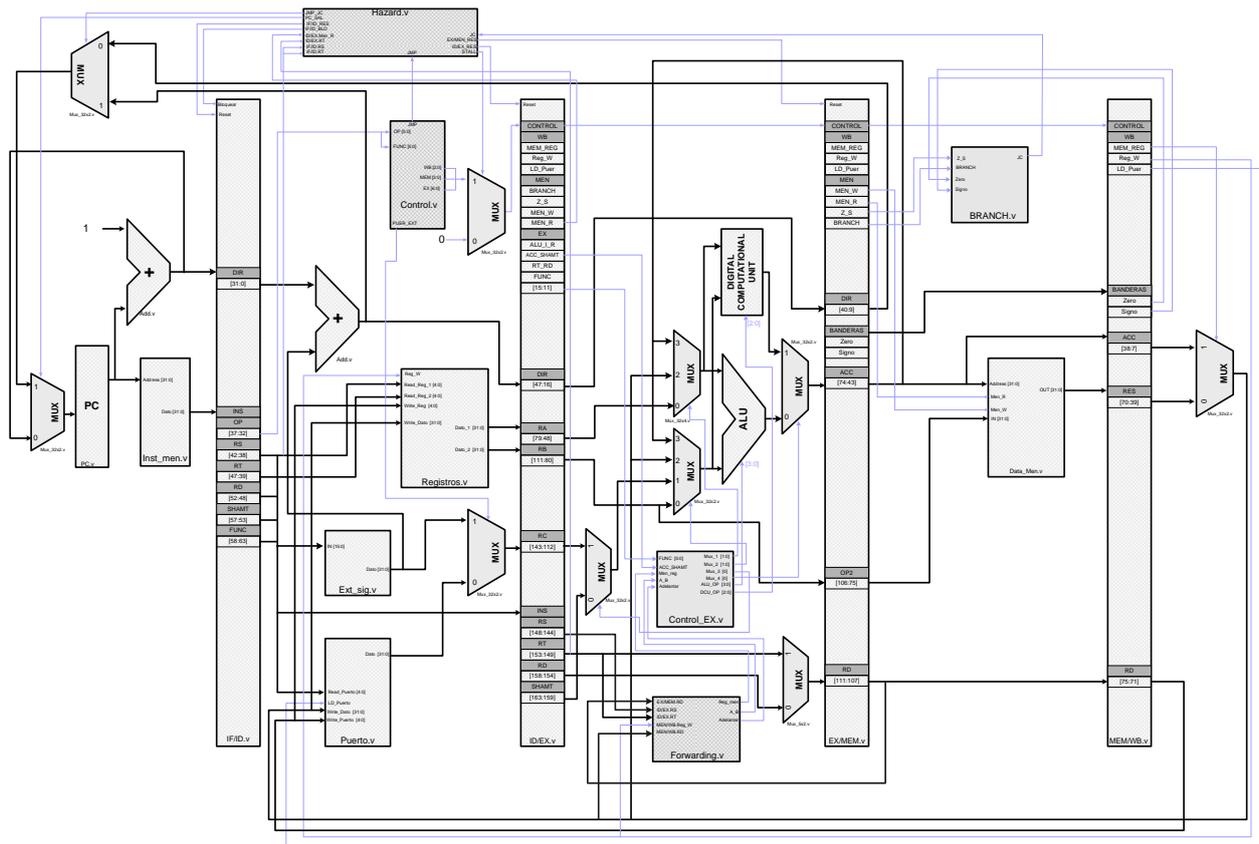


Fig.4. Arquitectura segmentada

se activa. Para la solución de este problema se optó por el adelantamiento de datos. Este consiste en utilizar un multiplexor en la entrada de la ALU para utilizar el dato que se calculó en el ciclo de reloj pasado como operando en lugar del dato obtenido del registro.

Al realizar operaciones de memoria en una arquitectura segmentada también se corre el riesgo de realizar un acceso a memoria y que se utilice el valor leído de memoria en la próxima instrucción este problema se presenta cuando uno de los operandos de las dos próximas instrucciones está al registro destino de la operación de lectura de memoria. Para estos casos existen dos opciones, si la instrucción con problemas de acceso a memoria es la tercera en ejecutarse se debe adelantar el dato de la etapa de escritura de registro a la etapa de ejecución. Para el caso de que las dos instrucciones estén continuas se debe parar un ciclo de reloj la ejecución de la instrucción y además se debe adelantar el dato de la etapa de escritura de registro. Para detener la ejecución de una instrucción por un ciclo de reloj se realizó la modificación a la arquitectura que consiste en un multiplexor en la salida de la unidad de control conectado en la segunda entrada a un cero lógico que permite hacer una “burbuja” en la ejecución del programa, además de un bloqueo del PC y del primer registro de segmentación.

### 6.3. Riesgo en la ruta de control

Al ejecutar instrucciones de salto se debe tomar en cuenta que las condiciones de salto solo pueden ser evaluadas hasta la etapa de ejecución por lo tanto para el

momento en que se sabe si se debe saltar o no ya hay dos instrucciones ejecutándose en el procesador.

Para enfrentar este problema se asumió que la condición de salto no se dará y se continuarán ejecutando las instrucciones, en el caso de que se presente la condición de salto se enviará una señal desde la etapa de ejecución al módulo de control de salto para limpiar los registros de segmentación y cargar la dirección correcta en el PC. En el caso de los saltos incondicionales, si el CU identifica una instrucción de salto enviara una señal de control a la unidad de control de salto que cargara la dirección de salto incondicional en el PC.

## 7. IMPLEMENTACIÓN

Con base en la arquitectura segmentada desarrollada en la etapa de diseño se implementó utilizando el lenguaje de programación orientado al hardware Verilog, mediante la herramienta ISE V8.2i de Xilinx, además por tratarse de un esquema segmentado la programación se facilitó porque cada etapa se implementó independientemente. Para la simulación y depuración se realizó utilizando la herramienta ModelSim SE V6.1d. Una vez depurada y sintetizada la arquitectura se procedió a su implementación en una FPGA Spartan 3 XC3S1000, mediante la herramienta de Xilinx iMPACT V8.2i.

En la tabla 1 se muestra el conjunto de instrucción que se implementó en la presente arquitectura. Se indica tanto

el tipo como la distribución de los bits en cada instrucción.

**Tabla 1.** Instrucciones MIPS

Nombre	[31:26]	[25:21]	[20:16]	[15:11]	[10:6]	[5:0]
<b>Generales (Tipo R)</b>						
NOP	000000	00000	00000	00000	00000	000000
CMP	000001	RF 1	RF 2	00000	00000	000010
<b>Salto (Tipo I)</b>						
JMP	100000	00000	00000	CONSTANTE		
JS	100001	00000	00000	CONSTANTE		
JZ	100010	00000	00000	CONSTANTE		
<b>Direccionamiento Directo (Tipo R)</b>						
ADD	000010	RF 1	RF 2	DES	00000	000001
SUB	000010	RF 1	RF 2	DES	00000	000010
AND	000010	RF 1	RF 2	DES	00000	000011
OR	000010	RF 1	RF 2	DES	00000	000100
NOT	000010	RF 1	00000	DES	00000	000101
<b>Puerto (Tipo R)</b>						
LP	000011	RF 1	00000	PUER	00000	000000
SP	000100	PUER	00000	DES	00000	000000
<b>Direccionamiento Inmediato (Tipo I)</b>						
ADDI	100000	RF 1	DES	CONSTANTE		
SUBI	100000	RF 1	DES	CONSTANTE		
ANDI	100000	RF 1	DES	CONSTANTE		
ORI	100000	RF 1	DES	CONSTANTE		
<b>Memoria (Tipo I)</b>						
LW	100000	RF 1	DES	CONSTANTE		
SW	100000	RF 1	DES	CONSTANTE		

Por otra parte en la figura 5 se muestra el reporte final, sobre la utilización de recursos en el FPGA, la cual refleja que el consumo fue reducido. En cuanto al reloj del microprocesador se obtuvo una frecuencia cercana a 100MHz, lo cual refleja que se mejoró el rendimiento comparando con el procesador multiciclo[1].

Device utilization summary:

```
-----
Selected Device : 3s1000ft256-4
Number of Slices:          1568 out of 7680  20%
Number of Slice Flip Flop  1180 out of 15360  8%
Number of 4 input LUTs:    967 out of 15360  6%
Number of bonded IOBs:     113 out of 173   65%
Number of GCLKs:           1 out of 8      12%
```

Timing Summary:

-----  
Speed Grade: 4

```
Minimum period: No path found
Minimum input arrival time before clock: 4.994ns
Maximum output required time after clock: 9.776ns
Maximum combinational path delay: 11.400ns
```

**Fig. 5.** Reporte final de síntesis provisto por Project Navigator™ V7.1i

## 8. CONCLUSIONES

La arquitectura multiciclo constituye la base de la arquitectura segmentada del presente proyecto; esto se logró mediante la utilización y modificación de varios módulos de la arquitectura multiciclo, con los cuáles se implementó en principio una arquitectura uniciclo y a partir de la misma se llegó finalmente a la arquitectura segmentada.

La segmentación de la arquitectura multiciclo se llevo a cabo en cinco etapas: búsqueda de instrucción (IF), decodificación de la instrucción y búsqueda de operandos (ID), ejecución (EX), acceso a memoria (MEM) y retroalimentación de resultado (WB).

Las principales modificaciones que se realizaron a la arquitectura multiciclo para segmentación de la misma fueron: separación de memoria de datos y de instrucciones, acceso simultáneo en el banco de registros, independizar el PC de la ALU, modificaciones para uso de instrucciones MIPS.

En la ejecución de instrucciones de la arquitectura segmentada se encontró los siguientes riesgos: estructurales, en la ruta de datos, en la ruta de control, en el acceso a memoria. Para solventar esto implementó las unidades de adelantamiento y de predicción.

Con la segmentación se logró mejorar el rendimiento de la arquitectura multiciclo[1].

## 9. REFERENCIAS

- [1] Díaz García J. A., Salazar Brenes E., Quiros Rojas L., Diseño de un filtro digital (IIR) con microprocesador de Arquitectura Multiciclo en FPGA *Fpga Based Systems*, pp. 191–198, marzo 2006.
- [2] L. C. Rosales Alpízar, J. Bermúdez, R. Céspedes, E. Morales Rodríguez, R. Pereira Arroyo, A. Chacón Rodríguez., Diseño en VERILOG de una Arquitectura Segmentada de Microprocesador RISC. ITCR, marzo 2006.
- [3] Patterson David A. y Hennessy John L., Computer Organization and Design, The Hardware / Software Interface, 3 ed. Morgan Kaufmann Publishers, San Francisco, California, USA., pp. 284-353, 2005.
- [4] Patterson David A. y Hennessy John L. Arquitectura de computadoras: Un enfoque cuantitativo McGraw Hill. 1993.
- [5] Tsai Chi Huang, Sudhakar Yalamanchili, Roy W. Melton, Philip R. Bingham, Cecil O. Alford, "Teaching Pipelining and Concurrency using Hardware Description Languages," *mse*, p. 55, IEEE International Conference on Microelectronic Systems Education, 1999.