
Diseño e Implementación de un CPU RISC de 32 bits en un FPGA

Ing. MSc. Aurelio Morales Villanueva

**Facultad de Ingeniería Eléctrica y Electrónica
Universidad Nacional de Ingeniería**

<http://fieee.uni.edu.pe>

Agenda

- Introducción
- CPUs tipos CISC y RISC
- Selección de modelo de CPU RISC de 32 bits
- Herramienta EDA de Diseño Digital
- CPU RISC Monociclo en un FPGA
- CPU RISC Pipeline en un FPGA
- Demostración
- Resumen

Agenda

- **Introducción**

- CPUs tipos CISC y RISC
- Selección de modelo de CPU RISC de 32 bits
- Herramienta EDA de Diseño Digital
- CPU RISC Monociclo en un FPGA
- CPU RISC Pipeline en un FPGA
- Demostración
- Resumen

Introducción

Es muy conocido el auge que han tenido los microprocesadores en las últimas décadas. Hoy en día se cuenta con CPUs multicore.

Para fines de enseñanza e investigación, no se cuenta con una plataforma de hardware para el estudio de CPUs de arquitectura **RISC**.

También es conocido que existen fabricantes de SW y HW orientados a diseño digital especializado. Con el uso de dispositivos versátiles como los **FPGA** (Field Programmable Gate Array), asociados a herramientas **EDA** (Electronic Design Automation), es posible diseñar sistemas muy complejos.

Se buscará diseñar e implementar un CPU RISC de 32 bits del tipo pipeline escalar de 5 etapas en un FPGA.

Agenda

- Introducción
- **CPUs tipos CISC y RISC**
- Selección de modelo de CPU RISC de 32 bits
- Herramienta EDA de Diseño Digital
- CPU RISC Monociclo en un FPGA
- CPU RISC Pipeline en un FPGA
- Demostración
- Resumen

CPUs tipo CISC y RISC

CPU tipo CISC (Complex Instruction Set Computer)

- Formato de instrucción de longitud variable.
- Gran cantidad de instrucciones.
- Algunas instrucciones son muy complejas.
- Algunos formatos de instrucciones son muy poco utilizados.
- Variados modos de direccionamiento.
- Instrucciones de manipulación de datos (aritméticas, lógicas y desplazamiento) pueden usar operandos de memoria.
- Generalmente las instrucciones de manipulación de datos incluyen dos operandos, donde un operando es a la vez fuente y destino.
- No hay restricciones en el uso de instrucciones con operandos de memoria.
- Por lo general tienen una Unidad de Control Microprogramada.

CPUs tipo CISC y RISC (cont.)

CPU tipo RISC (Reduced Instruction Set Computer)

- Formato de instrucción de longitud fija.
- Relativamente poca cantidad de instrucciones.
- Todas las instrucciones son simples.
- Pocos formatos de instrucciones.
- Pocos modos de direccionamiento.
- Instrucciones de manipulación de datos (aritméticas, lógicas y desplazamiento) solo usan operandos de registro.
- Generalmente las instrucciones de manipulación de datos incluyen tres operandos, dos fuentes y un destino.
- Instrucciones con operandos de memoria restringido solamente a carga y almacenamiento.
- Por lo general tienen una Unidad de Control Alambrada.

Agenda

- Introducción
- CPUs tipos CISC y RISC
- **Selección de modelo de CPU RISC de 32 bits**
- Herramienta EDA de Diseño Digital
- CPU RISC Monociclo en un FPGA
- CPU RISC Pipeline en un FPGA
- Demostración
- Resumen

Selección de modelo de CPU RISC

Arquitectura del CPU

a) Arquitectura Von-Neumann

Un único medio de almacenamiento para las instrucciones y datos que maneja el CPU. Es decir una sola memoria.

Ocurre un “cuello de botella” cuando se desea realizar operaciones simultáneas (lectura de instrucción, y lectura/escritura en memoria)

b) Arquitectura Harvard

Dos medios de almacenamiento: uno para las instrucciones y otro para los datos. Es decir, dos memorias separadas.

Se puede trabajar independientemente con cada una de ellas.

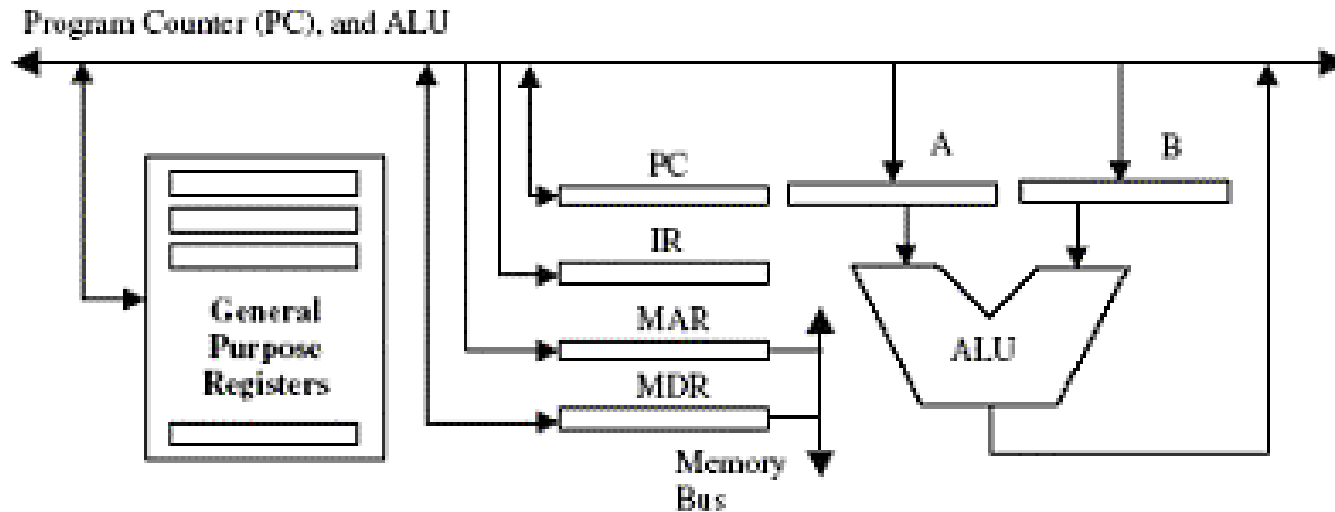
Mejor desempeño que arquitectura Von-Neumann

Tiene relación con memoria cache L1 en los CPUs.

Selección de modelo de CPU RISC (cont.)

Organización del Camino de Datos (Datapath)

a) Organización de un solo Bus

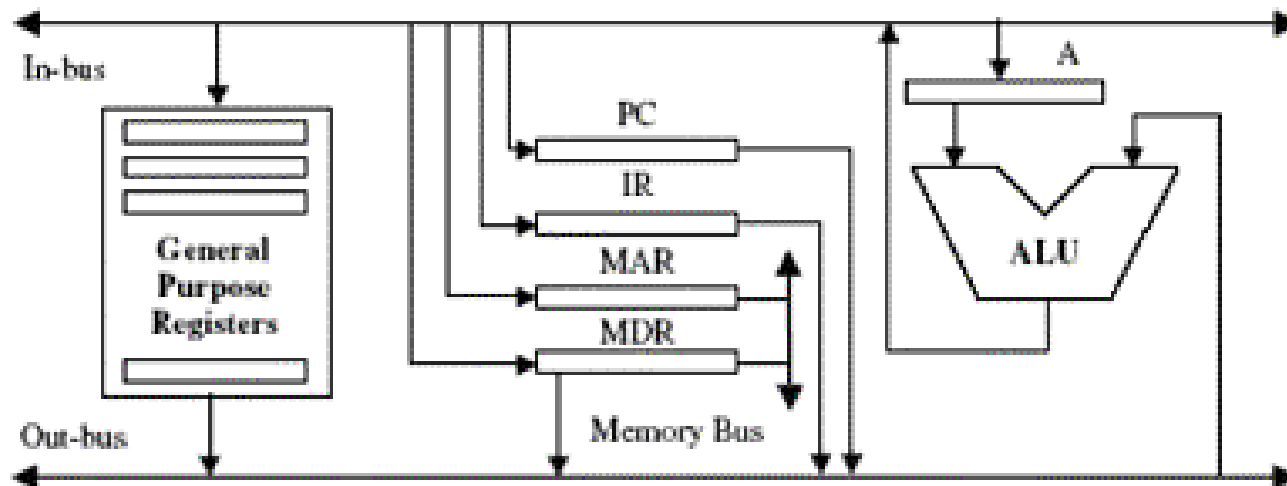


Se necesitan dos registros intermedios para las operaciones aritméticas y lógicas. Como consecuencia, las operaciones son lentas.

Selección de modelo de CPU RISC (cont.)

Organización del Camino de Datos (Datapath)

b) Organización de dos Buses

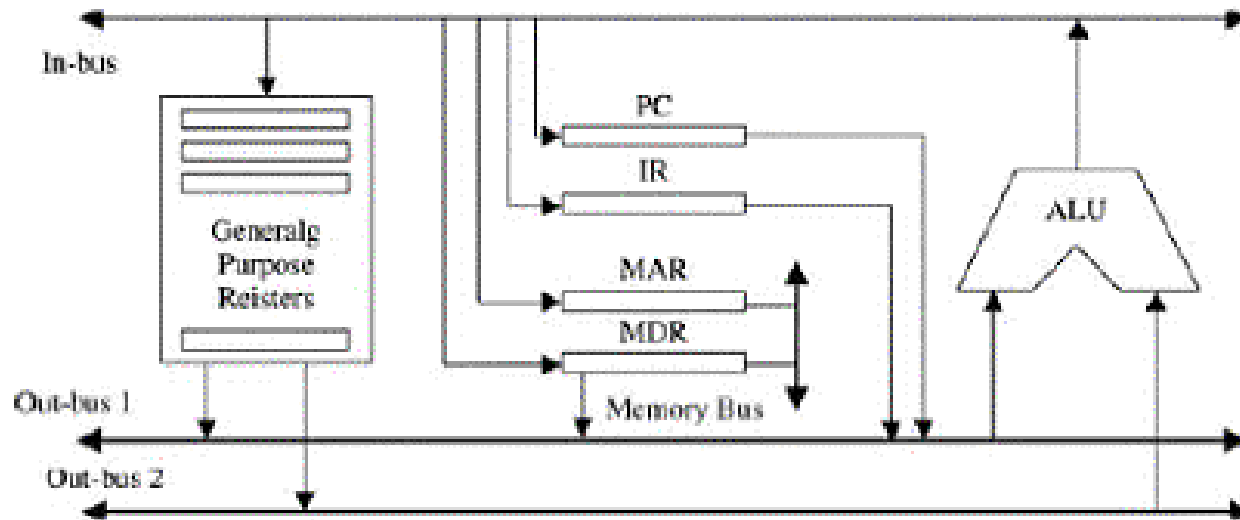


Un bus para lectura de elementos, y un bus para escritura de elementos. Problema de contención en el bus de salida.

Selección de modelo de CPU RISC (cont.)

Organización del Camino de Datos (Datapath)

c) Organización de tres Buses



Se pueden leer dos elementos del “Register File”.
Buscar un balanceo de operandos en los buses de lectura.
Organización preferida para CPU tipo RISC.

Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

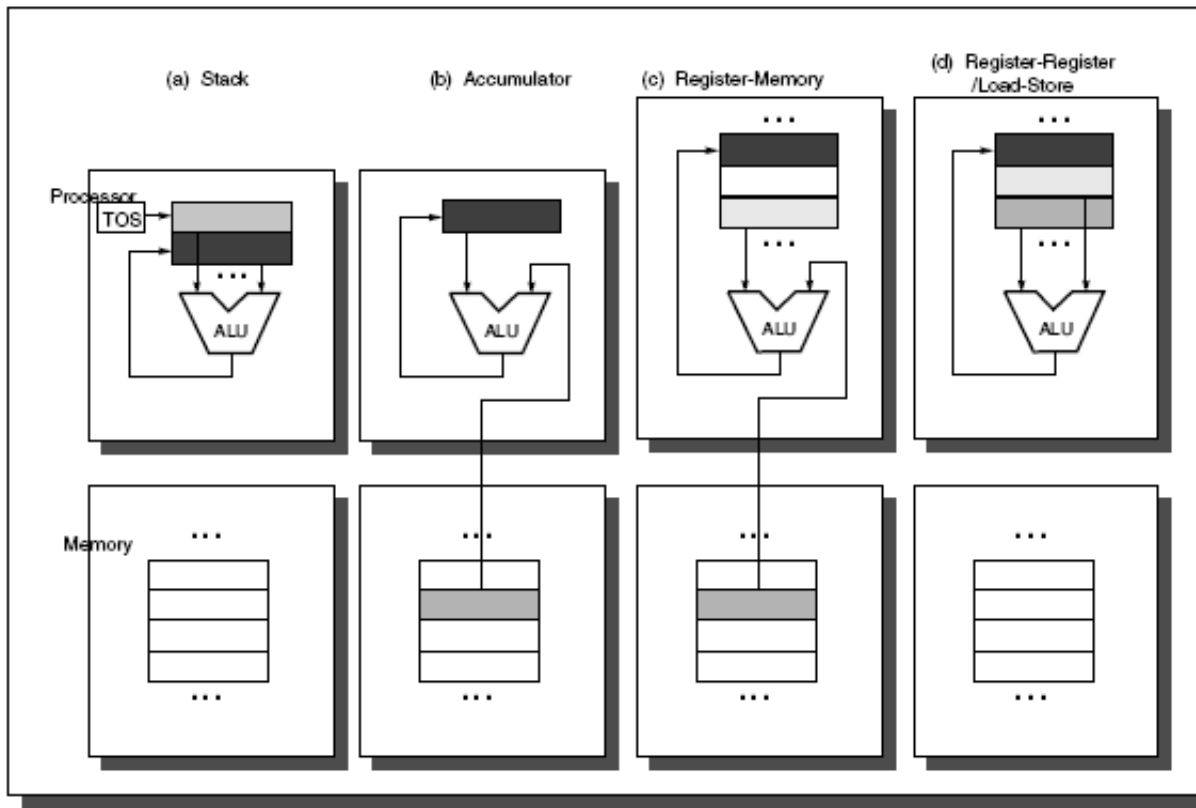
Criterios a tomar en cuenta:

- Ubicación de los operandos
- Direccionamiento de memoria
- Modos de direccionamiento
- Tipo y tamaño de los operandos
- Operaciones del juego de instrucciones
- Instrucciones de control de flujo
- Codificación del juego de instrucciones

Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

- *Ubicación de los operandos*



- a) Arquitectura basada en Pila (Stack)
- b) Arquitectura basada en un Acumulador
- c) Arquitectura Registro-Memoria
- d) Arquitectura Registro-Registro

Opción d) se escoge para CPU RISC

Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

Stack	Accumulator	Register (register-memory)	Register (load-store)
Push A	Load A	Load R1,A	Load R1,A
Push B	Add B	Add R3,R1,B	Load R2,B
Add	Store C	Store R3,C	Add R3,R1,R2
Pop C			Store R3,C

Implementación de $C = A + B$ en las cuatro arquitecturas

A, B, y C son direcciones simbólicas de memoria.

Ciclos *fetch* y *ejecución* se ven afectados por el tipo de instrucción.

Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

- ***Direccionamiento de memoria***

Consideraciones para datos e instrucciones multibyte:

a) Ordenamiento

- Little Endian
- Big Endian

En Little Endian el byte menos significativo ocupa la dirección menos significativa, y en Big Endian ocupa la dirección más significativa.

b) Alineamiento

- Instrucciones y Datos No Alineados
- Instrucciones y Datos Alineados

Instrucciones y Datos No Alineados se usa en CPUs del tipo CISC.

Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

• *Modos de direccionamiento*

- Por registro Ej: add r1, r2 ; $r1 \leftarrow r1 + r2$
- Inmediato Ej: add r4, #200 ; $r4 \leftarrow r4 + 200$
- Con desplazamiento Ej: add r3, 100(r5); $r3 \leftarrow r3 + \text{mem}[100 + r5]$
- Indirecto por Registro Ej: add r6, (r9) ; $r6 \leftarrow r6 + \text{mem}[r9]$
- Directo Ej: add r1, (100) ; $r1 \leftarrow r1 + \text{mem}[100]$
- Indirecto Ej: add r3, @(100); $r3 \leftarrow r3 + \text{mem}[\text{mem}[100]]$
- Autoincremento Ej: add r5, r2+ ; $r5 \leftarrow r5 + r2, r2 \leftarrow r2 + 1$
- Autodecremento Ej: add r6, -r2 ; $r2 \leftarrow r2 - 1, r6 \leftarrow r6 + r2$

Complejidad del modo de direccionamiento afecta al ciclo *fetch* y *ejecución* de las instrucciones.

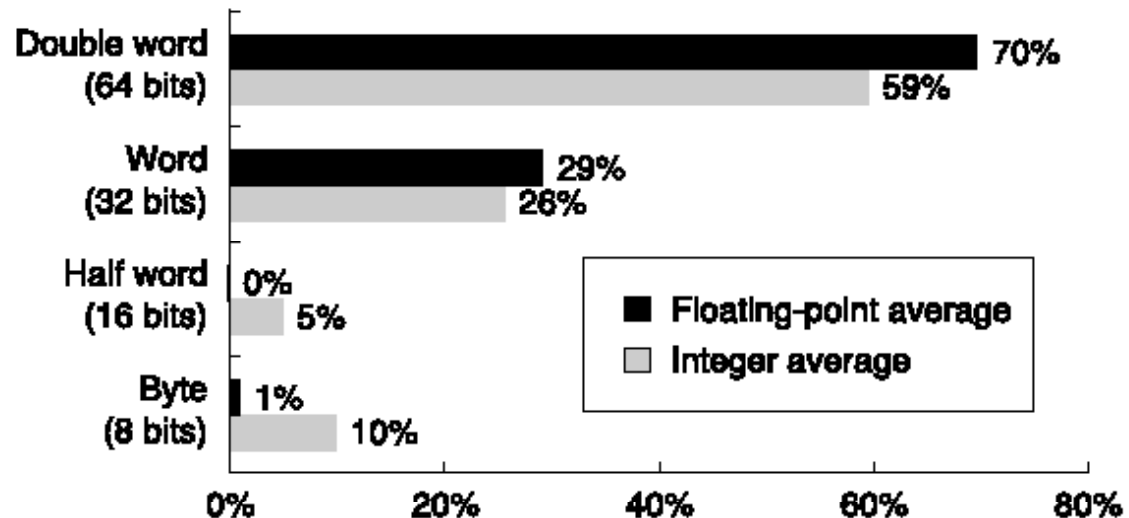
Debe escogerse el tamaño de los valores inmediatos y desplazamientos.

Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

• *Tipo y tamaño de operandos*

- Enteros, en representación de signo complemento a dos, de 8, 16, 32 y 64 bits.
- Caracteres ASCII (8 bits).
- Números en punto flotante de 32 o 64 bits usando el estándar IEEE 754.



Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

- ***Operaciones del juego de instrucciones***

- Operaciones aritméticas, lógicas y desplazamiento:
suma, resta, and, or, multiplicación, división, shl, shr
- Operaciones de transferencia de datos:
carga, almacenamiento
- Operaciones de control:
ramificaciones, saltos, llamado a subrutinas, retorno de subrutinas
- Operaciones del sistema:
llamadas del sistema operativo
- Operaciones de punto flotante:
suma, resta, multiplicación, división

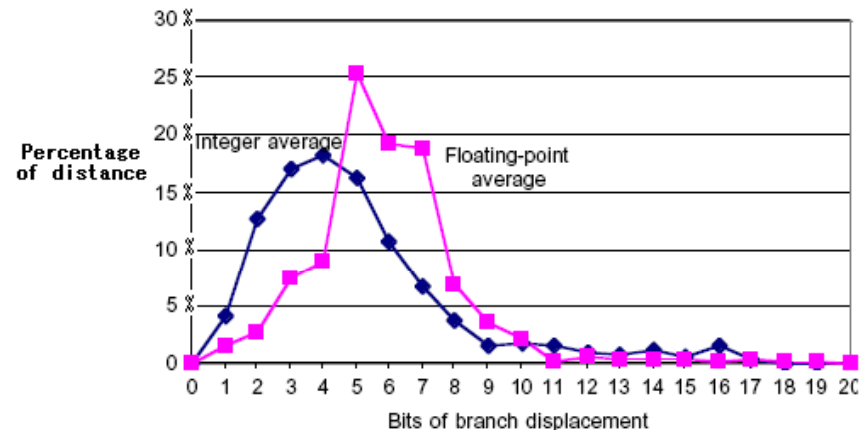
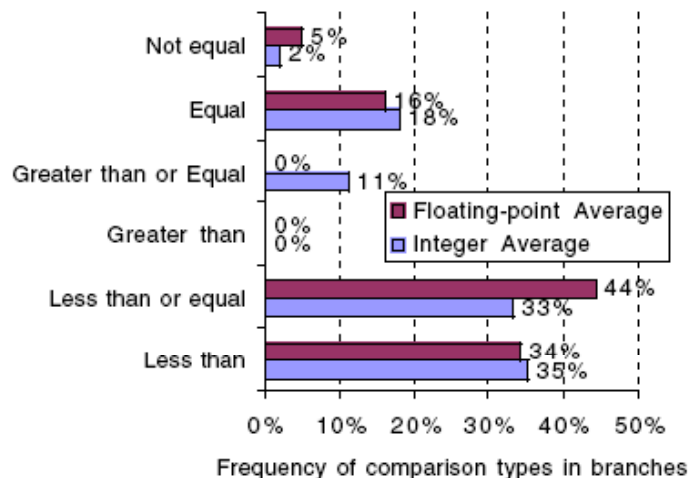
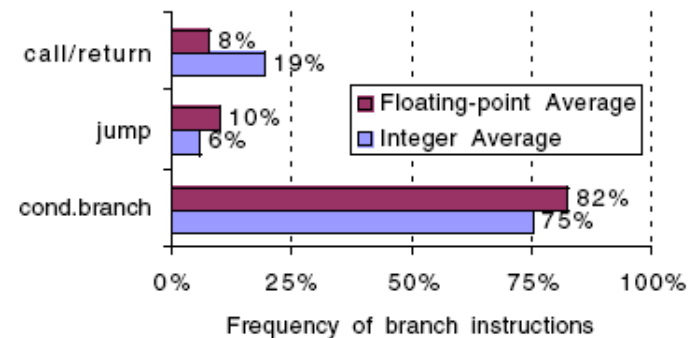
Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

• *Instrucciones de control de flujo*

- ramificaciones condicionales (conditional branches)
- saltos incondicionales (jumps)
- llamados a rutinas (calls)
- retornos de las rutinas (returns)

Debe conocerse el tamaño del desplazamiento relativo al PC.



Selección de modelo de CPU RISC (cont.)

Arquitectura del Juego de Instrucciones (ISA)

- ***Codificación del juego de instrucciones***

Puntos de partida:

- Formato de longitud de instrucción fija de 32 bits
- Register File de 32 registros
- Registros de 32 bits
- Solo instrucciones enteras

Entonces:

- 15 bits para especificar 3 operandos de registro, quedando 17 bits para especificar las operaciones.
- 16 bits para desplazamiento, quedando 16 bits para operación y operandos.
- Definir campos de operación, operandos y valores inmediatos.
- Formatos de instrucción deben favorecer fácil decodificación.

Agenda

- Introducción
- CPUs tipos CISC y RISC
- Selección de modelo de CPU RISC de 32 bits
- **Herramienta EDA de Diseño Digital**
- CPU RISC Monociclo en un FPGA
- CPU RISC Pipeline en un FPGA
- Demostración
- Resumen

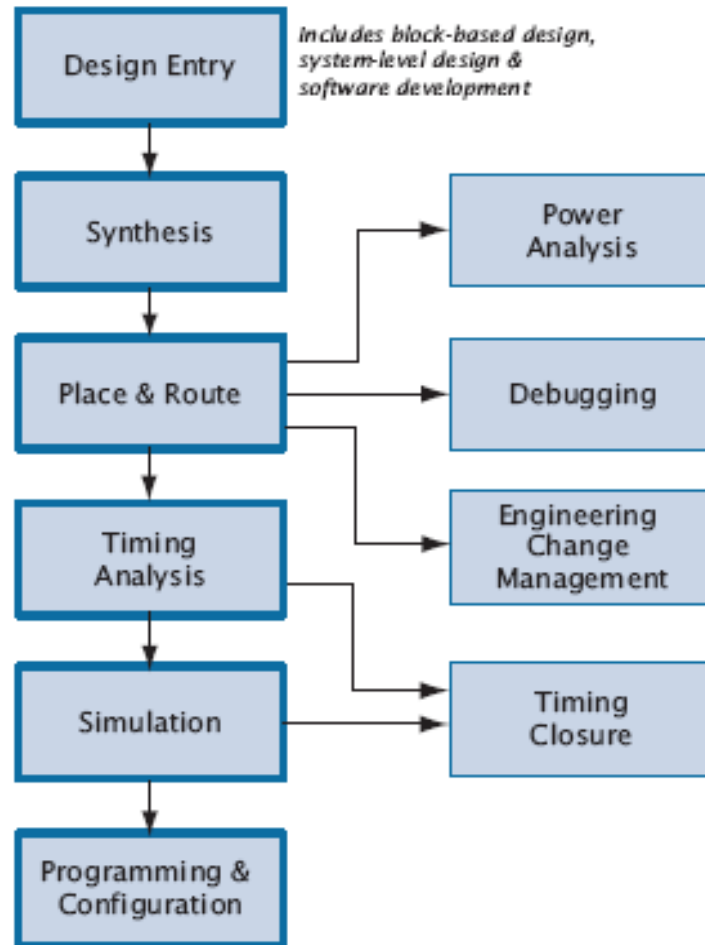
Herramienta EDA de Diseño Digital

Quartus II de Altera Corporation

- Herramienta para diseño e implementación de sistemas digitales.
- Múltiples formas para especificar el diseño:
Gráfica, Texto (lenguajes VHDL, AHDL, Verilog), Diagrama de Bloques.
- Posibilidad de generar sistemas completos en un chip, conocido como **SoPC** (System on Programmable Chip).
- Integración HW y SW usando el **IDE** (Integrated Development Environment).
- Muchas utilidades para depuración de proyectos.
- Soporte de amplia variedad de dispositivos (CPLDs y FPGAs).

Herramienta EDA de Diseño Digital (cont.)

Metodología de Diseño de Quartus II



The screenshot displays the Quartus II software interface for the project "MIPS_pipe - MIPS_pipe - [MIPS_pipe-HDU-FU-beq.vwf]". The top menu bar includes "MAX+PLUS II", "File", "Edit", "View", "Project", "Assignments", "Processing", "Tools", "Window", and "Help". The toolbar contains various icons for file operations, editing, and simulation.

The **Project Navigator** on the left shows the project hierarchy, including the "Entity" list with "MIPS_pipe" selected. The **Entity** list also shows the "Cyclone II: EP2C35F" device. The **Status** window at the bottom left shows the "Module" and "Proc" tabs.

The main window displays the **Timing Diagram** for the "MIPS_pipe-HDU-FU-beq.vwf" project. The **Master Time Bar** shows a time scale from 0 ps to 2.097972 us. The **Pointer** is set to 21.78 ns, and the **Interval** is -2.08 us. The **Name** column lists the signals being monitored:

- reset
- clock
- PC
- Instruction_out
- burbuja
- if_id_write_out
- PCWrite_out
- read_data_1_out
- read_data_2_out
- ALU_result_out
- Jump_out
- Branch_out
- Zero_out
- Regwrite_out
- Memwrite_out
- address_mem_out
- write_data_mem_out
- read_data_mem_out
- write_data_out
- IDcode:ID/register_array[1]
- IDcode:ID/register_array[2]

The timing diagram shows the waveforms for these signals. The **reset** signal is a single pulse at the beginning. The **clock** signal is a periodic square wave. The **PC** signal shows the program counter values. The **Instruction_out** signal shows the instruction values. The **burbuja** signal shows the bubble detection. The **if_id_write_out** signal shows the write enable for the instruction file register. The **PCWrite_out** signal shows the write enable for the program counter. The **read_data_1_out** and **read_data_2_out** signals show the data values read from the data memory. The **ALU_result_out** signal shows the ALU result. The **Jump_out** signal shows the jump enable. The **Branch_out** signal shows the branch enable. The **Zero_out** signal shows the zero flag. The **Regwrite_out**

25 de 38

Agenda

- Introducción
- CPUs tipos CISC y RISC
- Selección de modelo de CPU RISC de 32 bits
- Herramienta EDA de Diseño Digital
- **CPU RISC Monociclo en un FPGA**
- CPU RISC Pipeline en un FPGA
- Demostración
- Resumen

CPU RISC Monociclo en un FPGA

Versión reducida y compatible con CPU RISC de Arquitectura **MIPS** (**M**icroprocessor without **I**nterlock **P**ipeline **S**tages) de 32 bits:

- Formato de Instrucción de longitud fija de 32 bits.
- 32 registros enteros de 32 bits.
- Solo instrucciones enteras.
- Ejecución de instrucciones en 1 pulso de reloj.

campo
posición de bit

op	rs	rt	rd	shamt	func
32-26	25-21	20-16	15-11	10-6	5-0

a) Instrucción de Formato R

campo
posición de bit

op	rs	rt	Address or Immediate #
32-26	25-21	20-16	15-0

b) Instrucción de Formato I

campo
posición de bit

op	Address
32-26	25-0

c) Instrucción de Formato J

CPU RISC Monociclo en un FPGA (cont.)

Modos de direccionamiento

Modos de direccionamiento

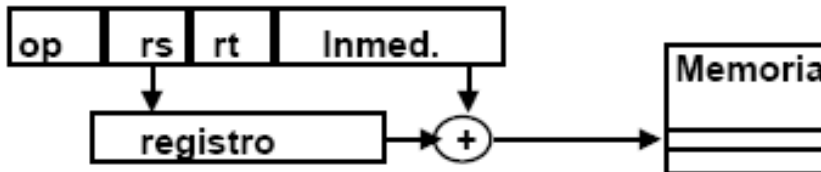
☒ Directo
de registro



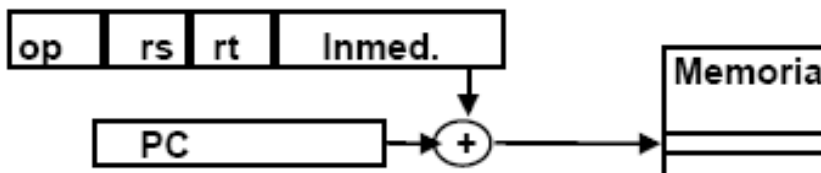
☒ Inmediato



☒ Indirecto con
desplazamiento

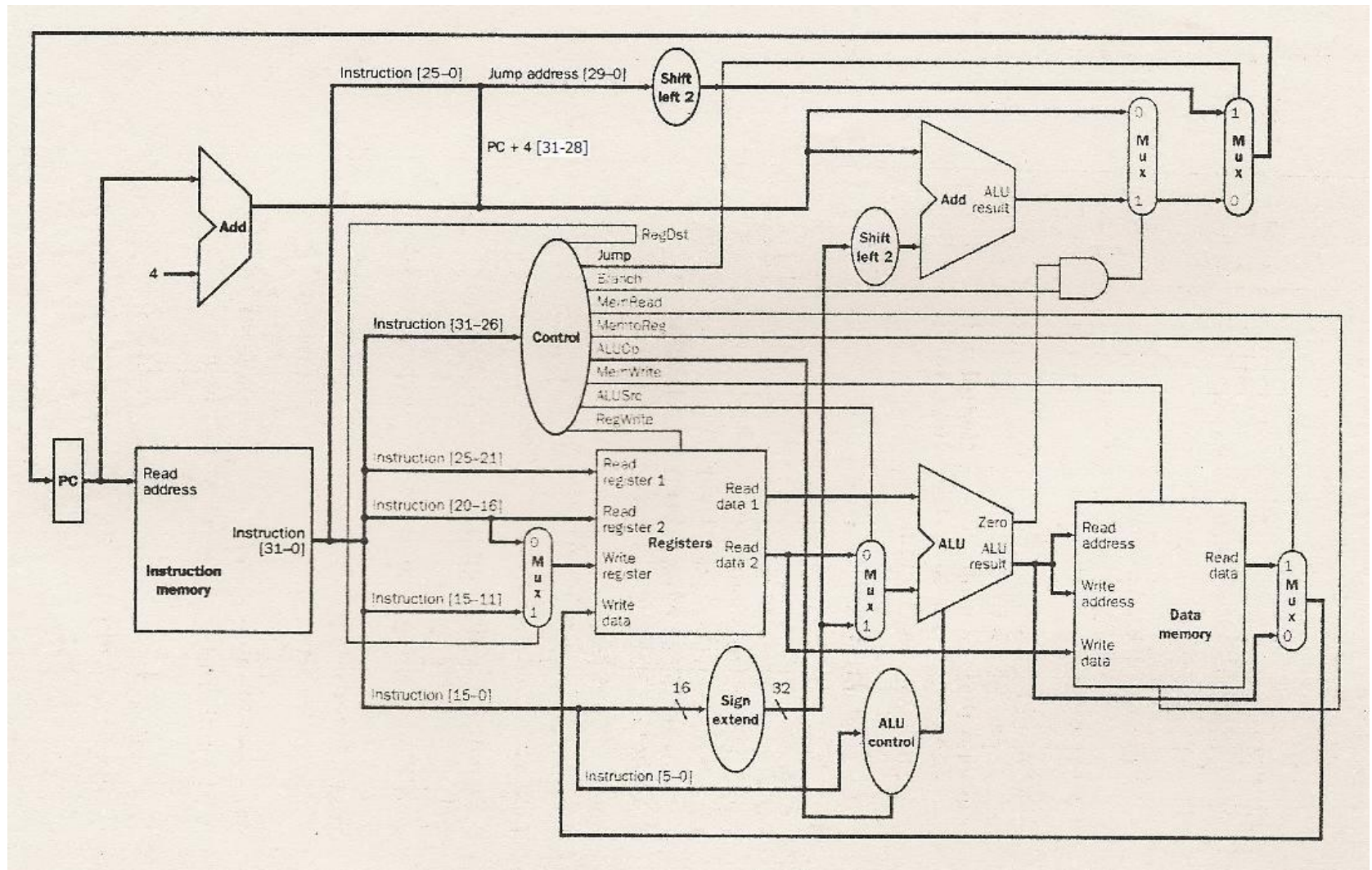


☒ Indirecto con
desplazamiento
relativo a PC



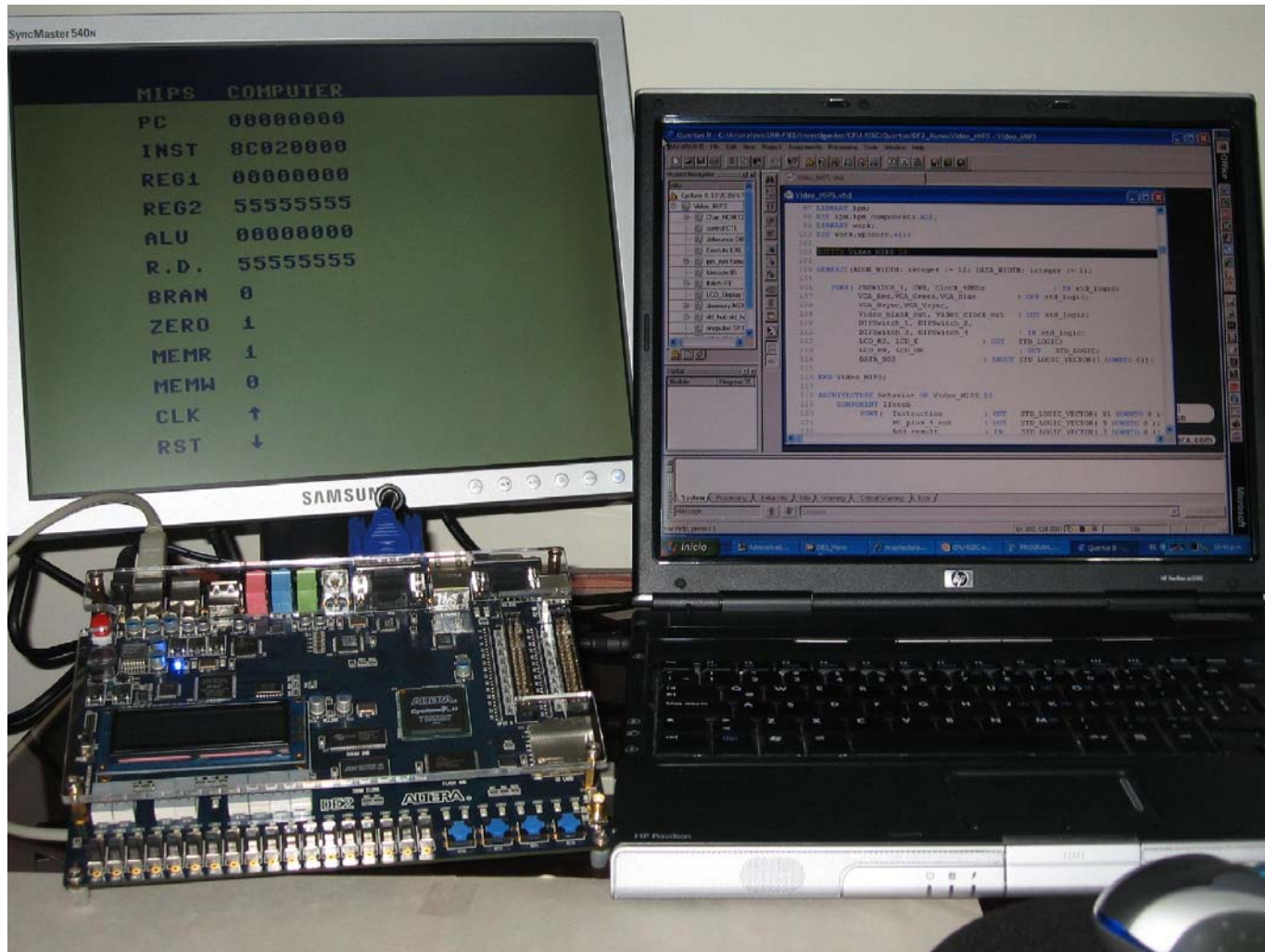
CPU RISC Monociclo en un FPGA (cont.)

Diagrama simplificado de implementación monociclo de CPU RISC



CPU RISC Monociclo en un FPGA (cont.)

Implementación de CPU RISC monociclo en tarjeta DE2 de Altera

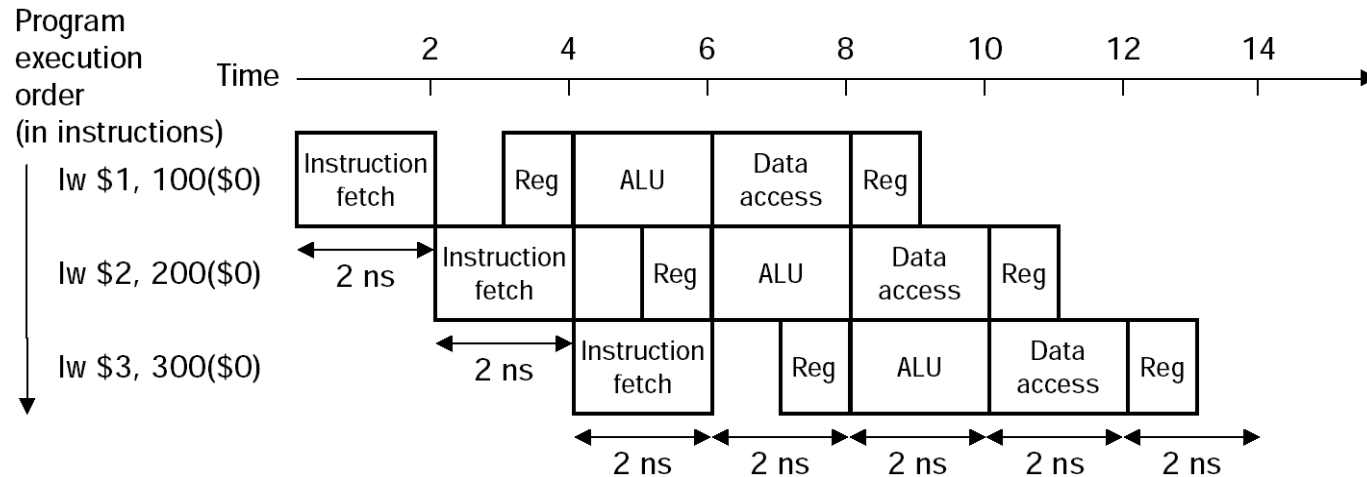
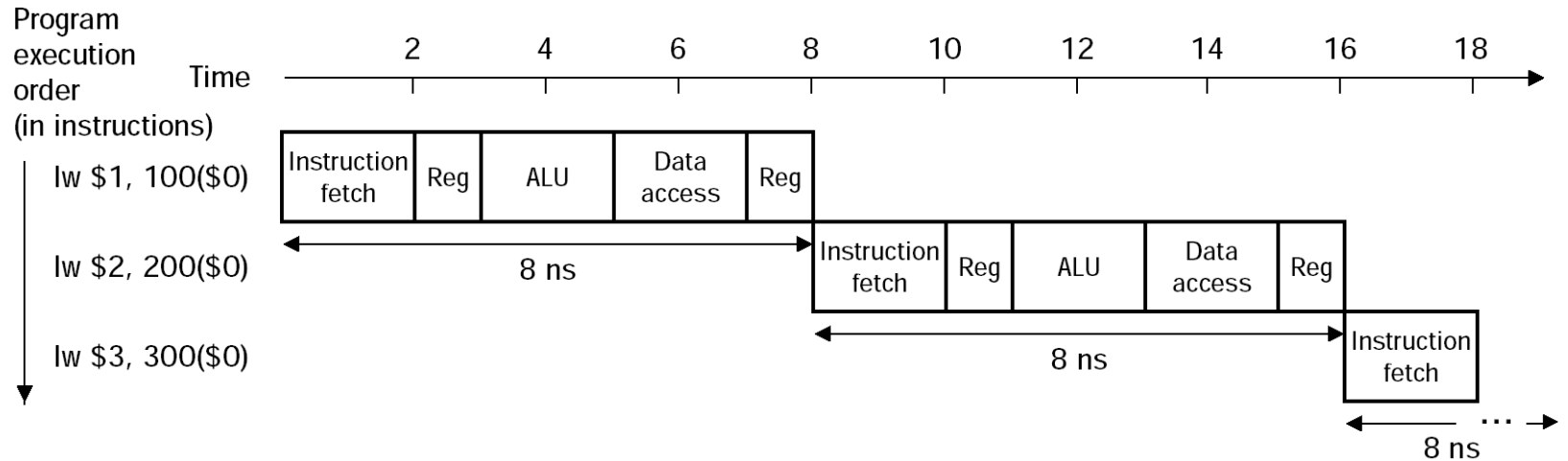


Agenda

- Introducción
- CPUs tipos CISC y RISC
- Selección de modelo de CPU RISC de 32 bits
- Herramienta EDA de Diseño Digital
- CPU RISC Monociclo en un FPGA
- **CPU RISC Pipeline en un FPGA**
- Demostración
- Resumen

CPU RISC Pipeline en un FPGA (cont.)

Pipeline: comparación



CPU RISC Pipeline en un FPGA

Conceptos para implementación pipeline

- Identificar las fases por las que puede pasar una instrucción.
- Distinguir las unidades funcionales involucradas en cada fase de la instrucción.
- Separar las unidades funcionales mediante registros.

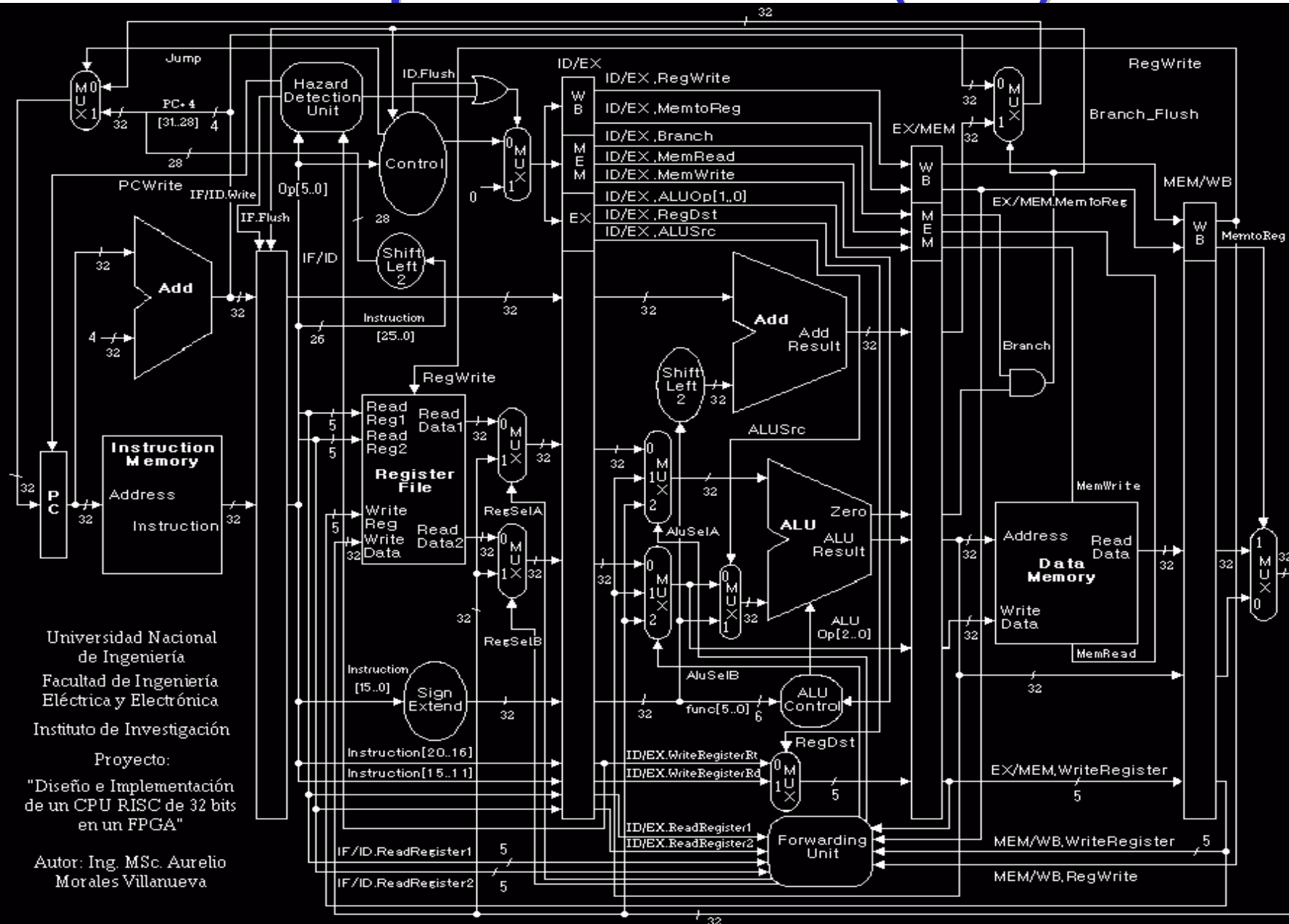
Problemas que se presentan

- Dependencias de datos **RAW** (Read after Write)
Ej: add r2, r3, r4 ; $r2 \leftarrow r3 + r4$
 sub r6, r2, r7 ; $r6 \leftarrow r2 + r7$
- Ejecución de instrucciones de ramificación

Solución a los problemas

- Bloques **HDU** (Hazard Detection Unit) y **FU** (Forwarding Unit)

CPU RISC Pipeline en un FPGA (cont.)



Universidad Nacional
de Ingeniería
Facultad de Ingeniería
Eléctrica y Electrónica
Instituto de Investigación

Proyecto:
"Diseño e Implementación
de un CPU RISC de 32 bits
en un FPGA"

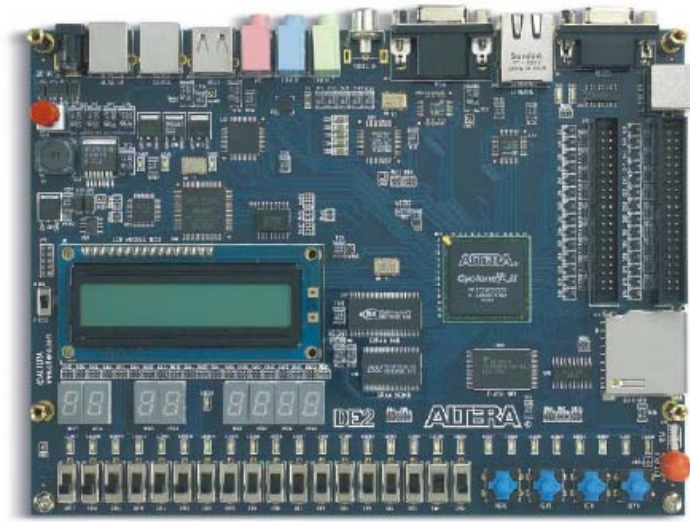
Autor: Ing. MSc. Aurelio
Morales Villanueva

Agenda

- Introducción
- CPUs tipos CISC y RISC
- Selección de modelo de CPU RISC de 32 bits
- Herramienta EDA de Diseño Digital
- CPU RISC Monociclo en un FPGA
- CPU RISC Pipeline en un FPGA
- **Demostración**
- Resumen

Demostración

- Implementación pipeline del CPU RISC de 32 bits. Incluye bloques HDU y FU.
- Implementación en tarjeta de desarrollo DE2 de Altera.
- Dispositivo FPGA Cyclone II EP2C35F672C6 de Altera.
- Interfaz a video VGA 1024x768 y visualizador LCD.
- Herramienta EDA Quartus II.
- Especificación de diseño usando lenguaje VHDL.
- Implementa un “emulador” de un CPU RISC.
- Para fines educativos y de investigación en el campo de Arquitectura de Computadoras.



Agenda

- Introducción
- CPUs tipos CISC y RISC
- Selección de modelo de CPU RISC de 32 bits
- Herramienta EDA de Diseño Digital
- CPU RISC Monociclo en un FPGA
- CPU RISC Pipeline en un FPGA
- Demostración
- **Resumen**

Resumen

- Revisión de CPUs CISC y RISC
- Implementación de un CPU RISC en un FPGA
- Demostración

Referencias

- <http://en.wikipedia.org/wiki/RISC>
- <http://en.wikipedia.org/wiki/FPGA>
- http://es.wikipedia.org/wiki/Arquitectura_MIPS
- <http://www.altera.com>