

# La Aritmética del Computador en el Diseño Curricular

Julio Sahuquillo<sup>1</sup>, José Flich<sup>1</sup>, Salvador Petit<sup>2</sup>, Juan-Carlos Cano<sup>1</sup>

Departamento de Informática de Sistemas y Computadores

<sup>1</sup>Escuela Técnica Superior de Informática Aplicada

<sup>2</sup>Facultad de Informática

Universidad Politécnica de Valencia

e-mail: {jsahuqui, jflich, spetit, jucano}@disca.upv.es

## Resumen

El procesador, la entrada/salida y la memoria son las tres unidades funcionales clásicas establecidas en la denominada máquina de von Neumann. Dentro del procesador, los operadores digitales encargados de realizar las operaciones aritméticas (suma, resta, etc) y lógicas (and, or, etc) quedan englobadas en la denominada Unidad Aritmético-Lógica (ALU).

El estudio de este conjunto de circuitos se cubre a lo largo de distintos temas en las asignaturas de Estructura y Arquitectura del Computador. El enfoque típico utilizado se centra en la implementación de los operadores a nivel de puertas, para posteriormente pasar a describir el funcionamiento del mismo. Este enfoque es de gran ayuda para el aprendizaje y asentamiento de las bases teóricas. Sin embargo, es necesario utilizar nuevos enfoques tales como los simuladores de circuitos y el lenguaje ensamblador, para que el alumno tenga una visión global y aplicada de la ALU y conozca la interfaz entre el hardware y el software. Organizar todas estas herramientas a lo largo de dos asignaturas de diferentes cursos es complicado.

En este trabajo, se describe la organización de las enseñanzas relativas a la aritmética del computador en las asignaturas de Estructuras de Computadores en los estudios de Informática de la Universidad Politécnica de Valencia, y se presentan distintas alternativas ortogonales para abordar el estudio de la ALU. Para ello, se describe brevemente para cada curso, qué bloques temáticos abarca y qué tipo de prácticas se pueden realizar. También se discuten las

ventajas y limitaciones de cada tipo de prácticas en función de la herramienta utilizada.

## 1. Introducción

Las actividades propuestas en los modelos de diseño curricular para alcanzar los objetivos propuestos por una determinada asignatura son muchos y de muy diversa índole. El presente trabajo se enmarca en el plan de estudios de la titulación de Ingeniero Técnico en Informática de Sistemas (ITIS) e Ingeniero en Informática (II). En concreto en las asignaturas relativas al estudio de la estructura del Computador; es decir, Estructura y Tecnología de Computadores I (ETCI) y Estructura y Tecnología de Computadores II (ETCII) en ITIS, y las análogas Fundamentos de Computadores (FCO) y Estructuras del Computador (EC) en II.

El estudio de la unidad aritmético-lógica, tiene por sí sola, un gran interés tanto a nivel de investigación como docente. De hecho, muchas de las tesis doctorales suelen dirigirse a resolver problemas relacionados con la temática. Esta temática, debido a su complejidad se suele abordar en varias asignaturas. Por ejemplo, en el contexto de ITIS, previamente citado, se aborda durante el primer y segundo curso, mientras que en el contexto de II, además de estudiar en primero y segundo cursos de carrera los mismos contenidos que en ITIS, se profundiza posteriormente en la asignatura de Arquitectura del Computador que se imparte durante el cuarto curso de carrera.

Entre los distintos elementos curriculares que se utilizan para conseguir el aprendizaje (clases teóricas, de problemas, laboratorio,

trabajos, etc) en este artículo nos centramos en los trabajos de laboratorio. Para poder analizar que actividades puede realizar el alumno necesitamos conocer los conocimientos previos que éste ha adquirido tanto en clases magistrales como de problemas. Por este motivo, realizaremos un breve repaso a los distintos bloques temáticos que el alumno ha estudiado.

Además, se analizarán las ventajas y desventajas de cada una de las propuestas en función de la herramienta utilizada en el laboratorio para poder realizar la práctica.

El resto del artículo se organiza como sigue. En la sección 2 se presentan las asignaturas de estructura del computador y su relación con la ALU. En la sección 3 se describen algunas de las prácticas realizadas. Finalmente, en la sección 4 se presentan algunas de las conclusiones más relevantes.

## 2. Las asignaturas de Estructuras y su relación con la ALU

En las asignaturas de ETCI de la titulación de ITIS así como la análoga (FCO) de la titulación de II se estudian los bloques temáticos mostrados en la Tabla 1.

Bloque	Contenido
1	Representación de la Información
2	Lógica Digital
3	Lenguajes de bajo nivel
4	Rutas de datos sencillas

Tabla 1. Bloques temáticos de las asignaturas de primer curso: ETCI y FCO

Por otra parte, los bloques temáticos que se estudian en las correspondientes asignaturas de segundo curso (ETCII en la titulación de ITIS y Estructuras del Computador en II) se muestran en la Tabla 2.

Bloque	Contenido
1	Subsistema de Memoria
2	Subsistema de Entrada/Salida
3	Unidad Aritmético-Lógica
4	Segmentación Básica

Tabla 2. Bloques temáticos de las asignaturas de segundo curso: ETCII y Estructuras de Computadores

En relación con la aritmética del computador, los alumnos de primer curso estudian las bases sobre la representación de la información (binario natural, enteros con signo, reales y caracteres) en el bloque 1 y el funcionamiento de un sumador con propagación de acarreo básico en el bloque 2, por lo que el posible abanico de prácticas que pueden realizar en relación con la ALU es relativamente limitado. En relación con las herramientas para realización de prácticas, los alumnos se familiarizan con el entrenador lógico para la implementación de circuitos digitales básicos. Por tanto, se podría pensar en la preparación de prácticas con un conjunto reducido de circuitos integrados y conexiones a realizar.

Además, los alumnos se inician en el estudio del lenguaje ensamblador del MIPS R2000 que continua a lo largo de la asignatura ETCII de segundo curso.

Por otra parte, los alumnos de segundo curso ya conocen un simulador digital, el cual han empezado a utilizar durante el segundo cuatrimestre de primer curso y lo siguen utilizando en diferentes asignaturas (diseño lógico y ETCII). En concreto se trata de una versión educacional del simulador Xilinx [1].

En relación a los contenidos teóricos de la ALU en ETCII se estructuran en 4 temas:

- Suma y resta de enteros.
- Multiplicación y división de enteros.
- Aritmética en coma flotante.
- Segmentación de circuitos.

En estos temas se aborda el estudio de circuitos realmente complejos. Así, si se desea estudiar el funcionamiento de éstos, es aconsejable la utilización de simuladores de circuitos digitales. Por tanto, las asignaturas de segundo curso son ideales para realizar la propuesta de prácticas utilizando estas herramientas de simulación.

## 3. Tipos y distribución de prácticas de ALU

En base a los contenidos expuestos en el punto anterior, en este trabajo se presentan tres alternativas para realizar prácticas relacionadas con la ALU según la herramienta utilizada:

- Entrenador lógico.
- Simulador digital.
- Lenguaje ensamblador.

### 3.1. Consideraciones previas sobre la preparación de prácticas

Las prácticas que se pueden realizar sobre aritmética del computador pueden ser muy distintas en función de la herramienta que se utilice. Sin embargo, los logros alcanzables con las distintas herramientas son complementarios.

Un entrenador lógico es una herramienta inviable si hay que utilizar más de unos cuarenta cables para realizar las conexiones. Por tanto, los entrenadores y montajes reales sólo se pueden utilizar para montajes de circuitos relativamente pequeños. Por el contrario, tienen la ventaja de trabajar con chips reales, lo que supone un estímulo para el alumno.

El simulador digital tiene como principal ventaja su gran flexibilidad, que les permite simular circuitos de gran tamaño. Por tanto es útil para realizar circuitos como: sumadores con seis o más operandos, multiplicadores, CPAs de números grandes (64 o más bits), etc. Por el contrario, con frecuencia, su uso no incentiva demasiado al alumno. Esto, en parte es debido al entorno que ofrece la herramienta de simulación, por lo que es conveniente elegir una herramienta potente y con un entorno “casi” profesional.

El contacto del alumno con el ensamblador es muy recomendable, ya que ayuda al alumno a comprender aspectos relacionados con el hardware; por ejemplo cómo un procesador con una ALU de 32 bits permite a los lenguajes de programación trabajar con datos de 64 bits, o más grandes. Sin embargo, requiere que el alumno conozca con soltura el lenguaje ensamblador. Muchas veces, esta faceta supone un serio obstáculo, aunque se haya estudiado previamente en clases de teoría.

Por último, se elija la herramienta que se elija, siempre es conveniente que los alumnos conozcan la tarea a realizar en el laboratorio con vistas a que traigan la práctica preparada al laboratorio a fin de aprovechar al máximo el tiempo en el mismo.

A continuación se describe brevemente una práctica ejemplo, para cada una de las

herramientas analizadas, presentando ejercicios complementarios y de ampliación, a fin de tener una visión de las posibles prácticas realizables con la herramienta. Asimismo, se analizan las ventajas y limitaciones que conlleva cada herramienta.

### 3.2. Prácticas con el entrenador lógico

Los alumnos utilizan el entrenador lógico durante el primer cuatrimestre de primer curso, por lo que ya es posible preparar una práctica con esta herramienta.

**Práctica ejemplo.** El objetivo de la práctica que se describe como ejemplo es que los alumnos construyan una unidad aritmético-lógica sencilla mediante el uso de algunos circuitos integrados. En líneas generales, las tareas que realiza el alumno son:

- Construcción de un sumador utilizando el integrado 7483 (sumador binario de 4 bits).
- Construcción de un sumador/restador utilizando el integrado 7483 y puertas XOR.
- Construcción de la unidad aritmético-lógica que implementa dos operaciones lógicas y dos aritméticas, utilizando el sumador, puertas lógicas, y el integrado 74183 (multiplexor dual) como circuito de selección de la operación a realizar.

A continuación se describen algunos montajes que el alumno debe realizar durante la práctica.

#### Sumador binario de 4 bits

La Figura 1 muestra el esquema de patillaje del integrado 7483.

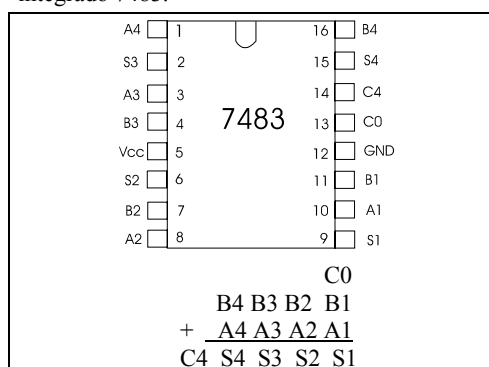


Figura 1. Esquema de patillaje y operación del 7483.

El alumno debe utilizar el chip para obtener los bits de suma y los indicadores de resultado (V, C, y Z) estudiados en clase de teoría. Antes de implementarlo, el alumno debe rellenar previamente el esquema de conexionado que se ofrece en la Figura 2.

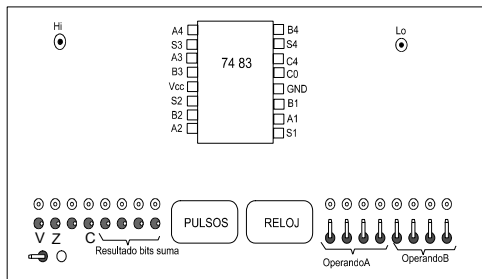


Figura 2. Plantilla para realizar las conexiones del sumador de 4 bits.

Una vez obtenido el esquema, y montado el circuito, el alumno debe verificar su funcionamiento, realizando algunas sumas en el circuito.

**Sumador/restador en Ca2**

A continuación los alumnos deben construir un circuito sumador/restador utilizando el 7483 y puertas XOR del entrenador, utilizando el esquema mostrado en la Figura 3. Una vez montado los alumnos deben verificar su correcto funcionamiento.

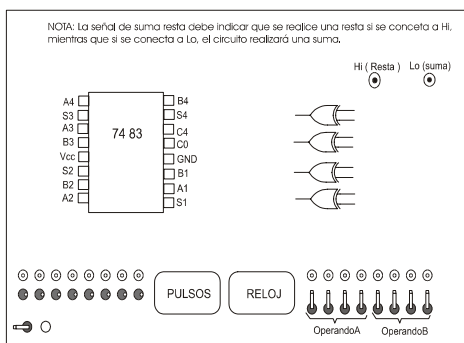


Figura 3. Plantilla para realizar las conexiones de un sumador/restador de 4 bits.

**Construcción de la ALU**

Finalmente, los alumnos deben implementar una unidad aritmético-lógica sencilla. A fin de reducir el número de conexiones, y facilitar la interpretación se ha elegido un ALU de dos operandos de dos bits, con solo una operación aritmética (la suma) y 3 lógicas. Para ello el alumno utiliza, el sumador, puertas lógicas y dos multiplexores, uno para cada bit de salida. En concreto, se utiliza el multiplexor dual 74153 (integra dos multiplexores de 4 a 1 que comparten entradas de selección).

El esquema de conexionado de la ALU se muestra en la Figura 4. Una vez realizado el esquema de conexiones, los alumnos deben implementarlo en el entrenador. Para verificar su funcionamiento deben comprobar las salidas del circuito para distintas operaciones variando los valores de los operandos.

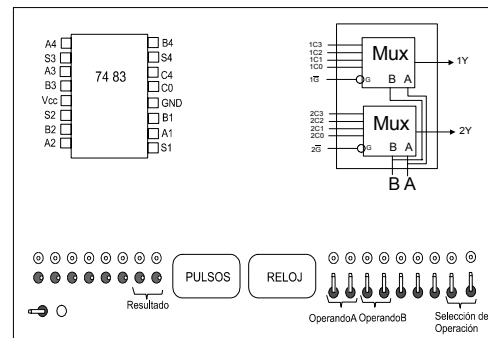


Figura 4. Plantilla para realizar las conexiones de la ALU.

**3.3. Prácticas con un simulador digital**

Plantear prácticas con este tipo de herramientas en primer curso es difícil debido a que desconocen el uso de la herramienta. Por otra parte dada la sencillez de los circuitos implementados, este tipo de herramientas no aportarían ningún beneficio. Por lo tanto, las prácticas sobre simuladores digitales deben realizarse en la asignatura de estructuras del computador de segundo curso.

Las prácticas que se proponen como ejemplo son:

- Implementar sumadores/restadores con propagación de acarreo, a partir de circuitos más simples (semisumadores o sumadores completos).
- Construir sumadores de gran tamaño a partir de sumadores más pequeños.
- Implementar un árbol de Wallace para sumar 6 operandos de cuatro bits.

Durante los dos últimos cursos académicos se ha utilizado el entorno de simulación digital Xilinx [1]. Esta herramienta tiene una versión profesional y otra de estudiante. La versión de estudiantil es más reducida, pero ofrece un buen entorno y amplias posibilidades de trabajo.

#### **Implementación de un semisumador o sumador completo a partir de semisumadores**

En la primera parte los alumnos construyen un semisumador a partir de puertas lógicas, y a partir de los semisumadores construyen un sumador completo.

#### **Implementación de un CPA de 8 bits mediante sumadores completos**

A continuación, el alumno utiliza los sumadores completos construidos previamente para construir un sumador con propagación de acarreo para dos números de 8 bits.

Una vez construido el circuito, el alumno necesita realizar un circuito de prueba adicional para verificar su funcionamiento.

Básicamente consta de dos contadores, que generan los valores de los números a sumar. Los operandos se suman al mismo tiempo en el CPA y en un circuito ya implementado previamente. Si los resultados de ambas partes coinciden, entonces el CPA funciona correctamente, sino habrá que repasar las conexiones. El circuito de comprobación se debe implementar para comprobar el funcionamiento de todos los circuitos que se implementen con la herramienta.

#### **Circuito sumador/restador de 16 bits**

Utilizando el CPAs de 8 bits se implementa un sumador/restador de 16 bits. El circuito es

relativamente sencillo utilizando puertas XOR adicionales y la señal de suma/resta.

#### **Composición de sumadores/restadores**

A partir del circuito sumador/restador de 8 bits el alumno debe implementar un sumador/restador de 32 bits. El circuito es relativamente sencillo de construir a partir del anterior, ya que se trabaja a nivel de bloques.

#### **Árboles de Wallace**

En clases de teoría se estudian los árboles de Wallace como técnica para implementar sumadores rápidos con cualquier número de operandos.

En la práctica se trata de implementar un nuevo módulo denominado CSA (*carry save adder*) o sumador sin propagación de acarreo.

Posteriormente, el alumno, debe interconectar distintos niveles de CSA, con el nivel inferior donde colocará el CPA implementado previamente.

#### **3.4. Prácticas con lenguaje ensamblador**

Como ejemplo de práctica en lenguaje ensamblador el alumno debe implementar la suma de dos números de 64 bits (o múltiplos de 32) utilizando la ALU de un procesador real de 32 bits.

Además, el alumno debe obtener el valor de los indicadores de resultado (*flags*) de las operaciones aritméticas. En concreto debe obtener los valores de Z (Zero), N (Negative), C (Carry) y O (Overflow).

Para realizar la suma, deben sumarse independientemente la parte baja (bits del 31 al 0) y la parte alta (bits del 63 al 32). La Figura 5 representa gráficamente el problema utilizando para ello CPAs de 32 bits. La diferencia principal radica en que ahora se desea utilizar la ALU del procesador y no un simulador digital. Como se aprecia en la figura, se necesita saber si se genera acarreo al sumar los 32 bits de menor peso ( $C_{31}$ ), para sumárselos a los 32 bits de mayor peso.

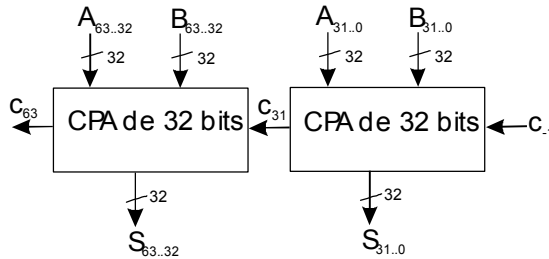


Figura 5. Diagrama de bloques de un CPA de 64 bits a partir de CPAs de 32 bits.

El código del programa es el expuesto en la Figura 6. Aunque este estudio se realiza mediante el simulador del MIPS R2000 [3], el ejercicio se puede particularizar para cualquier lenguaje ensamblador conocido por el alumno.

```
.data 0x10000000
A:    .word A31..0, A63..32
B:    .word B31..0, B63..32
Resul: .space 8
      .text
__start:
      la $4,A
      lw $3, 0($4) # parte baja A
      lw $4, 4($4) # parte alta A
      la $6,B
      lw $5, 0($6) # parte baja B
      lw $6, 4($6) # parte alta B
      add $7,$3,$5
      sltu $8,$7,$3 # $8 = C31
      add $8,$8,$4
      add $8,$8,$6
      la $10, Resul
      sw $7, 0($10) # p. baja Res
      sw $8, 4($10) # p. alta Res
```

Figura 6. Código del programa ejemplo

Otro ejemplo de práctica ubicable en el segundo curso, es el desarrollo de pequeñas rutinas que realicen operaciones aritméticas no soportadas en el procesador MIPS R2000 [2]. Como ejemplo, se pueden citar los siguientes:

- Detección de desbordamiento en la suma/resta de enteros sin signo.
- Implementación de aritmética de saturación de 32 bits y de 64 bits.
- Comparación de operandos de 64 bits.

```
suma8cs:
      add $v0, $a1, $a0
      slti $t0, $v0, 127
      bne $t0, $zero, nosat
      # saturacion por arriba
      move $v0, $t0
      b finsuma8cs
nosat:  slti $t0, $v0, -128
      beq $t0, $zero, finsuma8cs
      # saturacion por abajo
      move $v0, $t0

finsuma8cs:
      jr $31
```

Figura 7. Código de aritmética de saturación de 8 bits con signo.

```
comparar:
      #A=$a1|$a2, B=$a3|$a2
      slt $t0, $a1, $a3
      beq $t0, $zero, com1
      # $a1<$a3, -> A<B
      move $v1, $a3
      move $v0, $a2
      li $s1, 0
      b comfin
com1:  beq $a1,$a3, com2
      # $a1>$a3, -> A>B
      move $v1, $a1
      move $v0, $a0
      li $s1, 2
      b comfin
com2:  # $a1==$a3, -> A ¿? B
      slt $t0, $a0, $a2
      beq $t0, $zero, com3
      # $a0 < $a2, -> A<B
      move $v1, $a3
      move $v0, $a2
      li $s1, 0
      b comfin
com3:  beq $a0,$a2, com4
      # $a0>$a2, -> A>B
      move $v1, $a1
      move $v0, $a0
      li $s1, 2
      b comfin
com4:  # $a3|$a2==$a1|$a0 -> A==B
      move $v1, $a1
      move $v0, $a0
      li $s1, 1

comfin:
      jr $31
```

Figura 8. Código para comparar dos operandos de 64 bits.

La Figura 7 muestra la implementación de la suma con saturación de 8 bits con signo para el MIPS R2000. Por su parte, la Figura 8 muestra la implementación de la comparación de dos operandos de 64 bits.

#### 4. Conclusiones

En este artículo se han presentado los contenidos teóricos así como las herramientas de laboratorio con las que trabajan los alumnos en cada una de las asignaturas que abordan el estudio de la estructura del computador en las titulaciones de ingeniero técnico en informática de sistemas e ingeniero en informática.

Se ha particularizado sobre la unidad aritmético lógica, cuyo estudio se distribuye en las asignaturas de ETCI y ETCII de la Escuela Técnica Superior de Informática Aplicada (y las análogas de la Facultad) distribuidas en primer y segundo curso respectivamente.

Se ha analizado tanto los contenidos teóricos como las habilidades que adquiere el alumno en el laboratorio en las unidades temáticas relacionadas con la ALU.

En función de los conocimientos teóricos y las características de las distintas herramientas, se han presentado algunas prácticas complementarias con distinto grado de dificultad, cada una de ellas preparada para realizarse en una herramienta distinta.

Como fruto de esta experiencia podemos sintetizar las siguientes conclusiones:

- El trabajo con las distintas herramientas es enriquecedor para el alumno ya que al trabajar desde distintas perspectivas tiene una mejor visión de conjunto.
- Debido a que el tiempo es limitado, las prácticas deben prepararse cuidadosamente, prestando especial atención a que el alumno se pueda centrar en el concepto y no en la herramienta. En otras palabras, aquellas prácticas que requieran herramientas de mayor complejidad es conveniente que se realicen durante el segundo curso académico, donde el alumno ya se encuentra más familiarizado con las herramientas.

Por último, se desea matizar que aunque las propuestas que se ha discutido están estrechamente relacionadas en el marco docente en el que se ubican las asignaturas; éstas se pueden aplicar a cualquier plan de estudios, en función de la asignatura y curso en que se estudien los conceptos pertinentes.

#### 5. Referencias

- [1] Xilinx Foundation Series 2.1i Student Edition software, <URL: <http://www.xilinx.com/>>
- [2] G. Kane and J. Heinrich, MIPS RISC Architecture, Prentice Hall, 1992.
- [3] G. Larus, SPIM: a MIPS R2000/R3000 Simulator. Disponible en <http://www.cs.wisc.edu/~larus/spim.html>