

# Contenido

**Prefacio** xv

*Por Maurice V. Wilkes*

**Prólogo** xvii

**El elemento Gran Cuadro** xxix

**El simulador SPIM para los MIPS R2000/R3000** xxxiii

*Por James R. Larus, Universidad de Wisconsin*

## **1. Abstracciones y tecnología de computadores** 1

- 1.1. Introducción 1
- 1.2. Debajo del programa 3
- 1.3. Debajo de la carcasa 7
- 1.4. Circuitos integrados: el combustible innovador 17
- 1.5. Falacias y pifias 22
- 1.6. Observaciones finales 24
- 1.7. Perspectiva histórica y lecturas adicionales 26
- 1.8. Ejercicios 36

## **2. El papel del rendimiento** 41

- 2.1. Introducción 42
- 2.2. Medidas del rendimiento 46
- 2.3. Relación entre las métricas 47
- 2.4. Métricas populares del rendimiento 53
- 2.5. Elección de programas para evaluar el rendimiento 58
- 2.6. Comparación y resumen de rendimientos 60
- 2.7. Falacias y pifias 62
- 2.8. Observaciones finales 68
- 2.9. Perspectiva histórica y lecturas adicionales 69
- 2.10. Ejercicios 73

**3. Instrucciones: lenguaje de la máquina 81**

- 3.1. Introducción 82
- 3.2. Operaciones del hardware del computador 83
- 3.3. Operandos del hardware del computador 85
- 3.4. Representación de las instrucciones en el computador 89
- 3.5. Instrucciones para tomar decisiones 96
- 3.6. Soportar procedimientos en el hardware del computador 102
- 3.7. Otros estilos de direccionamiento de MIPS 108
- 3.8. Alternativas a la aproximación de MIPS 116
- 3.9. Un ejemplo para juntar todo 118
- 3.10. Un ejemplo más largo 121
- 3.11. Arrays frente a punteros 127
- 3.12. Falacias y pifias 130
- 3.13. Observaciones finales 131
- 3.14. Perspectiva histórica y lecturas adicionales 133
- 3.15. Ejercicios 137

**4. Aritmética para computadores 147**

- 4.1. Introducción 148
- 4.2. Números negativos 148
- 4.3. Suma y resta 154
- 4.4. Operaciones lógicas 159
- 4.5. Construcción de una unidad aritmético lógica 162
- 4.6. Multiplicación 175
- 4.7. División 188
- 4.8. Punto flotante 199
- 4.9. Falacias y pifias 217
- 4.10. Observaciones finales 218
- 4.11. Perspectiva histórica y lecturas adicionales 219
- 4.12. Ejercicios 229

**5. El procesador: Camino de datos y control 237**

- 5.1. Introducción 238
- 5.2. Construcción de un camino de datos 243
- 5.3. Un sencillo esquema de implementación 249
- 5.4. Una implementación de múltiples ciclos de reloj 276
- 5.5. Microprogramación: Simplificar el diseño del control 296
- 5.6. Excepciones 307
- 5.7. Falacias y pifias 312
- 5.8. Observaciones finales 313
- 5.9. Perspectiva histórica y lecturas adicionales 315
- 5.10. Ejercicios 318

**6. Mejora del rendimiento con la segmentación 321**

- 6.1. Introducción 322
- 6.2. Un camino de datos segmentado 325
- 6.3. Control segmentado 337
- 6.4. Riesgos por dependencias de datos 345
- 6.5. Control de los riesgos por dependencias de datos: detenciones 354
- 6.6. Reducción de los riesgos por dependencias de datos: anticipación 365
- 6.7. Riesgos de saltos 376
- 6.8. Excepciones 382
- 6.9. Rendimiento de los sistemas segmentados 386
- 6.10. Falacias y pifias 387
- 6.11. Observaciones finales 389
- 6.12. Perspectiva histórica y lecturas adicionales 392
- 6.13. Ejercicios 396

**7. Grande y rápido: explotar la jerarquía de memoria 401**

- 7.1. Introducción 402
- 7.2. Cachés 406
- 7.3. Memoria virtual 427
- 7.4. Un marco común para las jerarquías de memoria 446
- 7.5. Falacias y pifias 458
- 7.6. Observaciones finales 462
- 7.7. Perspectiva histórica y lecturas adicionales 465
- 7.8. Ejercicios 470

**8. Interfaz entre procesadores y periféricos 473**

- 8.1. Introducción 474
- 8.2. Medidas de rendimiento de E/S: algunos ejemplos de discos y sistemas de archivos 477
- 8.3. Tipos y características de los dispositivos de E/S 479
- 8.4. Buses: conexión de dispositivos de E/S al procesador y a la memoria 487
- 8.5. Interfaces de los dispositivos de E/S con memoria, procesador y sistema operativo 501
- 8.6. Falacias y pifias 512
- 8.7. Observaciones finales 514
- 8.8. Perspectiva histórica y lecturas adicionales 517
- 8.9. Ejercicios 520

**9. Procesadores paralelos 527**

- 9.1. Introducción 528
- 9.2. Computadores SIMD: flujo único de instrucciones, flujos múltiples de datos 528

- 9.3. Computadores MIMD: múltiples flujos de instrucciones, múltiples flujos de datos 533
- 9.4. Programación de los MIMD 534
- 9.5. MIMD conectados a un solo bus 537
- 9.6. MIMD conectados en red 549
- 9.7. Tendencias futuras para los procesadores paralelos 559
- 9.8. Falacias y pifias 565
- 9.9. Observaciones finales: Evolución frente a revolución en arquitectura de computadores 568
- 9.10. Perspectiva histórica y lecturas adicionales 570
- 9.11. Ejercicios 573

## **APÉNDICES**

### **A. Ensambladores, montadores de enlace y el simulador SPIM 577**

*Por James R. Larus, Universidad de Wisconsin*

- A.1. Introducción 577
- A.2. Ensambladores 584
- A.3. Montadores de enlace 590
- A.4. Proceso de carga 591
- A.5. Utilización de memoria 591
- A.6. Convenio de llamadas a procedimientos 593
- A.7. Excepciones e interrupciones 601
- A.8. Entrada y salida 604
- A.9. SPIM 606
- A.10. Lenguaje ensamblador MIPS R2000 616
- A.11. Observaciones finales 638
- A.12. Ejercicios 638

### **B. Las bases del diseño lógico 641**

- B.1. Introducción 641
- B.2. Puertas, tablas de verdad y ecuaciones lógicas 642
- B.3. Lógica combinacional 645
- B.4. Relojes 655
- B.5. Elementos de memoria 657
- B.6. Máquinas de estados finitos 670
- B.7. Metodologías de temporización 674
- B.8. Ejercicios 679

### **C. Implementación hardware del control 681**

- C.1. Introducción 681
- C.2. Implementación del control de máquinas de estados finitos 682

- C.3. Implementación de la función del estado siguiente con un secuenciador 694
- C.4. Traducción de un microprograma a hardware 700
- C.5. Observaciones finales 704
- C.6. Ejercicios 705

#### **D. Introducción a C para los programadores de Pascal 707**

- D.1. Introducción 707
- D.2. Declaraciones de variables 707
- D.3. Instrucciones de asignación 708
- D.4. Expresiones relacionales e instrucciones condicionales 709
- D.5. Bucles 710
- D.6. Ejemplos para poner todo junto 711
- D.7. Ejercicios 712

#### **E. Otra aproximación a la arquitectura de repertorio de instrucciones: VAX 713**

- E.1. Introducción 713
- E.2. Operandos del VAX y modos de direccionamiento 714
- E.3. Codificación de las instrucciones VAX 717
- E.4. Operaciones VAX 719
- E.5. Un ejemplo para poner todo junto: swap (intercambiar) 721
- E.6. Un ejemplo más largo: sort (ordenación) 725
- E.7. Falacias y pifias 729
- E.8. Observaciones finales 732
- E.9. Perspectiva histórica y lecturas adicionales 733
- E.10. Ejercicios 734

#### **Índice 737**