

# Tabla de Contenido

<b>Introducción</b>	<b>1</b>
<b>1. Marco teórico y estado del arte</b>	<b>4</b>
1.1. Marco teórico . . . . .	4
1.1.1. Definiciones básicas . . . . .	4
1.1.2. El procesador . . . . .	5
1.1.3. La ISA RISC-V . . . . .	5
1.1.4. El lenguaje <i>Assembler</i> . . . . .	6
1.1.5. La consistencia de memoria . . . . .	7
1.1.6. El módulo <i>Atomic</i> . . . . .	7
1.1.7. Métodos de sincronización . . . . .	9
1.1.8. Secciones paralelizables . . . . .	10
1.1.9. Complejidad temporal . . . . .	10
1.1.10. <i>SpeedUp</i> . . . . .	11
1.1.11. <i>CPU Time</i> y <i>Wall Time</i> . . . . .	11
1.1.12. Insertion sort . . . . .	11
1.1.13. Freedom studio . . . . .	12
1.1.14. <i>gem5</i> . . . . .	12
1.1.15. <i>OpenMP</i> . . . . .	14
1.2. Estado del arte . . . . .	15
1.2.1. Programa básico en assembler . . . . .	15
1.2.2. <i>Insertion sort</i> adaptativo . . . . .	16
1.2.3. <i>Insertion sort</i> de dos elementos . . . . .	17
<b>2. Metodología</b>	<b>18</b>
2.1. Análisis y consideraciones previas . . . . .	18
2.1.1. Análisis preliminar de <i>Insertion sort</i> . . . . .	18
2.1.2. Formalización del problema . . . . .	18
2.1.3. Resultados esperados . . . . .	19
2.2. Metodología de trabajo . . . . .	19
2.2.1. Parámetros de trabajo . . . . .	19
2.2.2. Estrategia de medición . . . . .	20
2.2.3. Procedimiento propuesto . . . . .	21
2.2.4. Ejemplo de aplicación . . . . .	23
2.2.5. Implementación del programa . . . . .	24

<b>3. Resultados y discusión</b>	<b>26</b>
3.1. Factores de la configuración	26
3.1.1. <i>Overheads</i> de <i>OpenMP</i>	26
3.1.2. Dispersión de los datos	27
3.1.3. Nivel de optimización en compilación del programa	28
3.2. Resultados de simulaciones	29
3.2.1. Según tamaño del arreglo	29
3.2.2. Según número de procesadores	30
3.2.3. Efecto de lenguaje <i>assembly</i>	31
3.2.4. Rendimiento CPU DerivO3	32
3.2.5. Rendimiento en peor caso	32
3.2.6. Comparación multifactorial	33
3.3. Discusión de resultados	35
3.3.1. Rendimiento solo con paralelismo	35
3.3.2. Sobre <i>assembly</i> y optimización	35
<b>Conclusión</b>	<b>36</b>
4.1. En base a resultados de la experiencia	36
4.2. Trabajo futuro	37
<b>Bibliografía</b>	<b>38</b>
<b>Appendices</b>	<b>40</b>
<b>A. Conceptos</b>	<b>41</b>
A.1. Acrónimos utilizados	41
A.2. Paralelismo y Concurrencia	42
A.3. Niveles de memoria	42
A.4. <i>Assembler</i> en línea y extendido	42
A.5. Optimización en GCC	43
<b>B. Programa utilizado</b>	<b>44</b>
B.1. Programa en C	44
B.2. Conversión a <i>assembly</i>	49
B.2.1. Primer <i>loop</i>	49
B.2.2. Segundo <i>loop</i>	50
B.2.3. Barreras	51
<b>C. Instrucciones RISC-V</b>	<b>53</b>
C.1. Formato base de instrucciones	53
C.2. Módulos estándares comunes de RISC-V	54
C.2.1. Módulo I	54
C.2.2. Módulo M	55
C.2.3. Módulo A	56
C.2.4. Módulos F y D	56
C.2.5. Módulo C	57