Práctica 1: Desarrollo de código para el procesador ARM

Profesores: Javier Resano, María Villarroya, Jesús Alastruey, Enrique Torres, Dario Suarez

Agradecimientos: L. Piñuel, J.I. Gómez, C. Tenllado...

Dep. Arquitectura de Computadores y Automática

Universidad Complutense de Madrid

Objetivos

- Conocer la estructura segmentada del procesador ARM 7
- Desarrollar código en ensamblador con los dos repertorios de instrucciones que soporta este procesador:
  - ARM: adecuado para optimizar el rendimiento
  - Thumb: adecuado para reducir el tamaño del código y el consumo de energía
- Optimizar código a nivel de ensamblador
- Combinar código en ensamblador con código en C
- Depurar código en ensamblador, siguiendo el contenido de los registros internos del procesador y de la memoria

Bibliografía

- Básica:
  - David Seal; **ARM Architecture Reference Manual**; 2ª Ed, Addison-Wesley, 2001 (versión antigua disponible en la web de la asignatura)
  - Primera parte del **cuaderno de prácticas del entorno Embest IDE** (disponible en la web de la asignatura)
- Complementaria:
  - A. Sloss, D. Symes, C. Wright; **ARM system Developer's Guide**; Morgan Kaufman, 2004
  - Steve Furber; **ARM System-on-chip Architecture**; 2ª Ed, Addison-Wesley, 2000
  - William Hohl; **ARM Assembly Language. Fundamentals and Tecniques**; CRC Press, 2009
- Enlaces
  - www.arm.com/documentation
  - www.arm.com/gnu

Entorno de trabajo

- Utilizaremos como entorno de trabajo Eclipse con una cadena de herramientas (*toolchain*) para compilación cruzada:
  - Herramientas de GCC para ARM para compilar y depurar
  - El plugin Zylin Embedded debug para simular el procesador (más adelante dejaremos el simulador y trabajaremos con una placa real)

Material disponible

- Cuadernos de prácticas de una asignatura de la Universidad Complutense que usa el mismo entorno de trabajo y las mismas placas
- Breve resumen del repertorio de instrucciones ARM
- Breve resumen del repertorio de instrucciones Thumb
- Manual de la arquitectura ARM
- Códigos fuente iniciales
- Linker script
- Directrices sobre cómo redactar la memoria

Estructura de la práctica

- Parte A (1 semana):
  - Apartado semi-guiado
  - Objetivos:
    - Familiarizarse con el entorno y los repertorios de instrucciones
    - Comparar los resultados que se obtienen al implementar de tres formas distintas (C, ensamblador ARM y ensamblador Thumb) una función sencilla
  - No hace falta entregarla

Estructura de la práctica

- Parte B (2 semanas):
  - Objetivo: Desarrollar tres versiones equivalentes de una misma función:
    - código C
    - ensamblador ARM
    - ensamblador Thumb
  - Comparar las tres opciones (función original, ARM y Thumb) teniendo en cuenta:
    - Tamaño del código
    - Nº de instrucciones ejecutadas
    - Tiempo de simulación
  - Debéis optimizar lo máximo posible los códigos en ensamblador, pero respetando la misma estructura que el código C y la modularidad.

Evaluación de la práctica

- Fechas estimadas:
- Entrega de esta parte: 22 de Octubre
- Entrega de la memoria: 29 de Octubre
- Las fechas definitivas se publicarán en la página web de la asignatura (moodle)

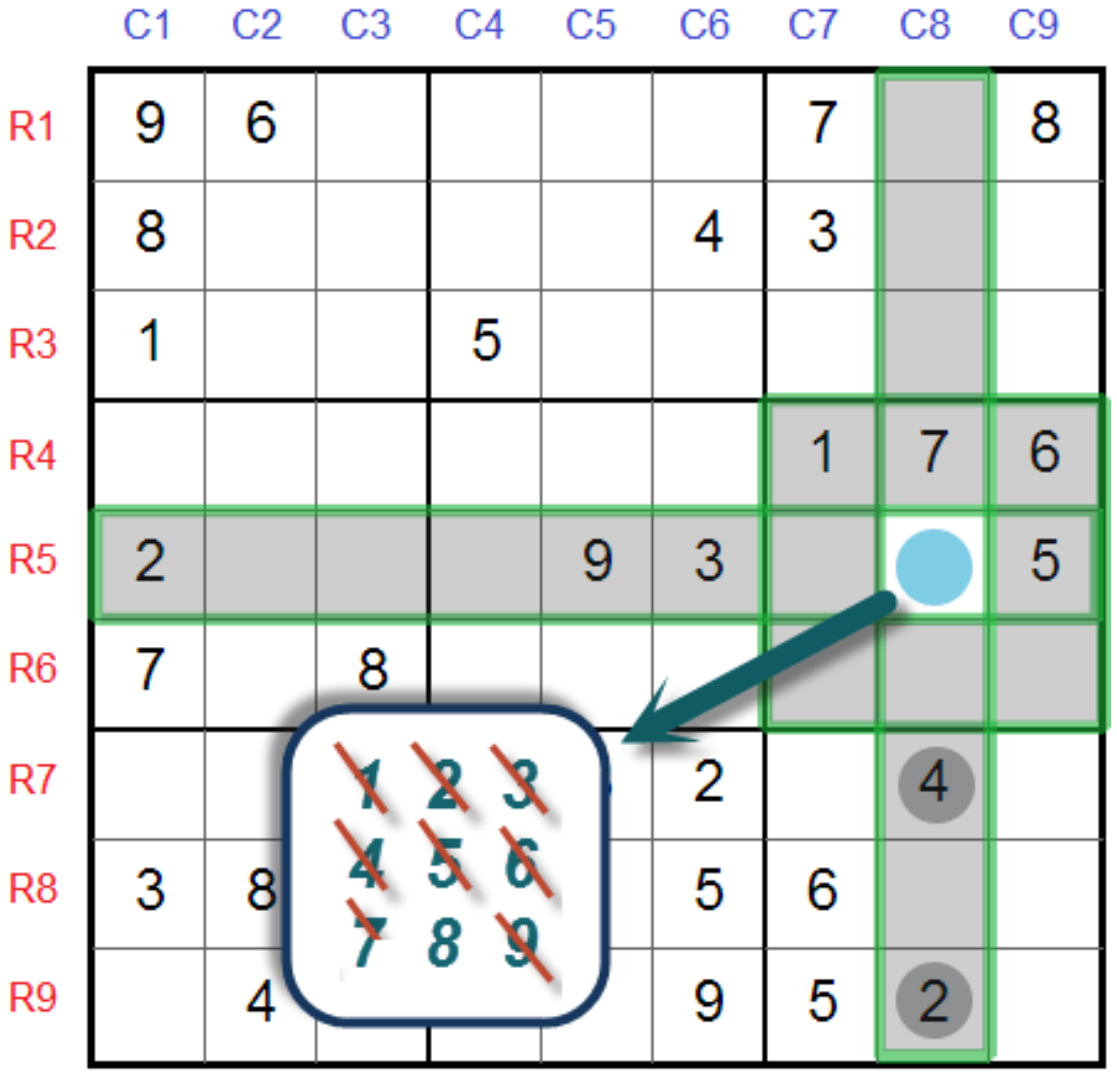
Realización de la memoria

- Resumen ejecutivo (**una cara máximo**)
- Descripción de las optimizaciones realizadas al código ensamblador
- Resultados de la comparación entre las distintas versiones de la función
- Descripción de los problemas encontrados en la realización de la práctica y sus soluciones
- Código fuente del apartado B comentado. Además, las **funciones Thumb y ARM deben incluir una cabecera** en la que se explique cómo funciona, qué parámetros recibe, dónde los recibe y qué uso se da a cada registro (ej. en el registro 4 se guarda el puntero a la primera matriz)

¿Sabéis hacer sudokus?

	C1	C2	C3	C4	C5	C6	C7	C8	C9
R1	9	6					7		8
R2	8					4	3		
R3	1			5					
R4							1	7	6
R5	2				9	3			5
R6	7		8						
R7			7		3	2		4	
R8	3	8	2	1		5	6		
R9		4	1			9	5	2	

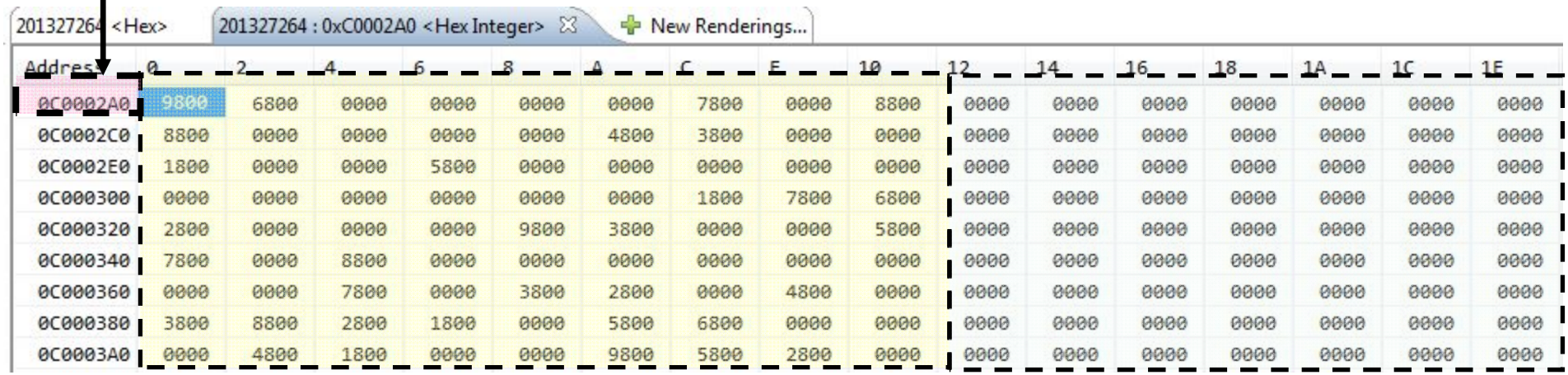
El cálculo de los candidatos es la clave



¿Cómo veremos el sudoku en el simulador?

- La cuadrícula del sudoku está almacenado en memoria
- hay que adaptar el tamaño de la ventana y ponerlo en formato hex_integer para que se visualice bien
- Cada posición son dos bytes que incluyen el número y la información adicional de los candidatos

Posición de comienzo del tablero



Tablero

Padding