

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores
(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores
(Licentiate Degree Program in Computer Engineering)

Curso: CE-4301 Arquitectura de Computadores I
(Course: CE-4301 Computer Architecture I)



Especificación Proyecto I
(Project I specification)

Profesor:
(Professor)

Ing.Jeferson González Gómez, M.Sc

Fecha de entrega: 31 de Marzo de 2016
(Due Date: March 31st, 2016)

Proyecto 1. Ensamblador y simulador de arquitectura ARMv4

1. Objetivo

Mediante el desarrollo de este proyecto, el estudiante aplicará los conceptos de arquitectura de computadores en el diseño de una herramienta de software para la simulación y ensamble de programas para la arquitectura ARMv4, considerando además los aspectos de hardware relacionados con la misma.

Atributos relacionados: Herramientas de Ingeniería (HI), Diseño (DI).

2. Descripción general

En el transcurso de las dos décadas pasadas, la arquitectura ARM ha crecido significativamente en popularidad debido principalmente a su eficiencia. El mercado de los sistemas embebidos, en general, y en específico los dispositivos móviles actuales, se encuentra dominado por procesadores ARM y computadores basados en Sistemas en Chip (SoC) o microcontroladores (MCU), cuyo núcleo, es también un procesador ARM. Es por esta razón que, en el marco de la arquitectura de computadores moderna, es fundamental conocer las características de una arquitectura tan rica, aplicable directamente a tecnologías y tendencias de punta como lo es el Internet de las cosas (IoT) y los sistemas embebidos en general.

Para este proyecto se deberán aplicar los conceptos de arquitectura de computadores, vista como una combinación de elementos de software y hardware, en el diseño e implementación de un simulador funcional de lenguaje ensamblador para arquitectura ARMv4. En el proyecto se desarrollará un acercamiento práctico a los principios de diseño del set de instrucciones, desde el punto de vista de arquitectura (instrucciones, formatos, modos de direccionamiento, codificación, etc.) y organización.

3. Especificación

Para este proyecto, se deberá diseñar e implementar un simulador de arquitectura ARMv4, que permita la escritura de código en ensamblador y su ensamblado correspondiente (traducción a lenguaje máquina), así como la simulación de la ejecución del código en cuestión, que deberá ser visualizada dentro de la aplicación a diseñar.

La aplicación a diseñar debe tener tres modos de operación: El primero deberá permitir la escritura de programas basados en ensamblador ARMv4. En este modo, la interfaz gráfica deberá permitir, además, guardar y cargar archivos de programas, en extensión .armv4. En el segundo modo de operación, se deberá permitir el ensamblado (traducción a lenguaje máquina) del código en ensamblador. El tercer modo deberá permitir la simulación del programa escrito y ensamblado en los modos 1 y 2.

A continuación se describe a mayor detalle la especificación del proyecto:

Detalles de arquitectura

Instrucciones

Para el desarrollo del proyecto, el simulador deberá permitir la escritura de programas utilizando ensamblador para ARMv4. Por lo tanto, el simulador/ensamblador deberá soportar las siguientes instrucciones:

- **Procesamiento de datos:**

- **AND**, **EOR**, **SUB**, **RSB**, **ADD**, **ADC**, **SBC**, **RSC**, **CMP**, **CMN**, **ORR**, **MOV**, **LSL**, **ASR**, **RRX**, **ROR**, **BIC**, **MVN**, **MUL**, **MLA**.

- **Memoria:**

- **STR**, **LDR**, **STRB**, **LDRB**.

- **Salto:**

- **B**, **BL**

Tipos de datos

Los tipos de datos con que tendrá interacción el sistema serán **enteros** de 8 y 32 bits. El uso de constantes (inmediatos) deberá apegarse a descripción de arquitectura ARMv4.

- Opcionalmente, se puede trabajar con tipos de datos más intermedios (halfwords, por ejemplo), pero su uso y soporte en arquitectura queda a criterio de cada grupo.

Registros

El simulador/ensamblador deberá soportar como operandos y resultados el **set completo de registros** de la arquitectura ARMv4 (R0-R15)

Modos de direccionamiento

El simulador/ensamblador deberá soportar los siguientes modos de direccionamiento, según especificación ARMv4:

- **Registro:** Registro solo
- **Base:** Offset inmediato, Offset de registro.
- **Relativo a PC**

Memoria

- La memoria de programa tendrá un tamaño máximo de 1KB y deberá empezar en la dirección 0x000.
- El espacio de memoria de datos inicia en la dirección 0x400 y deberá tener tamaño direccionable por la arquitectura de 1KB.
- Para simplificar, podrá ordenarse la memoria en bytes (8 bits) o en palabras (32 bits), según conveniencia de cada grupo.
- Con el fin de verificar el funcionamiento del simulador, el sistema deberá permitir almacenar (manualmente) valores en memoria de datos (puramente virtual). Por esta razón, la memoria de datos podrá modificarse ANTES de realizar la ejecución de los programas.

Ensamblado

El modo de ensamblado deberá permitir la correcta traducción a lenguaje máquina del código en ensamblador. Este es el modo principal de la aplicación, por lo que debe aplicar las reglas vistas en cursos anterior (lenguajes y compiladores) de análisis semántico y sintáctico para la detección de errores en la escritura del código en ensamblador. Cuando el código esté correcto, la aplicación deberá realizar un ensamblado del programa. Para esto, se deberá generar un archivo out.txt, que contenga la traducción a lenguaje máquina de las instrucciones escritas en ensamblador ARMv4. El archivo deberá tener un formato como sigue:

```
-----
Encabezado (opcional)
Dirección (hex) [tab] Instrucción 1 (hex)
Dirección (hex) [tab] Instrucción 2 (hex)
...      ...
-----
```

Simulación

En modo de simulación deberá permitir la correcta ejecución del programa. Se deberán implementar reglas básicas de detección de errores de ejecución (direcciones fuera de rango, desbordamiento... etc) En la simulación debe desplegarse visualmente el contenido de los 16 registros, así como de la memoria de código y datos. Además, se deberá estimar un tiempo de ejecución basado en una frecuencia de reloj de procesador de 100KHz. La interfaz de simulación deberá ser intuitiva y permitir todas las operaciones descritas anteriormente.

Otros

- Cada grupo estará conformado por 2 personas.
- El lenguaje de programación y toda plataforma de software queda a criterio de cada grupo.

- En todo momento se evaluará el diseño en ingeniería, por lo que las decisiones se deben justificar objetivamente y se debe hacer una comparación de las diferentes opciones que surjan.
- Cualquier cambio en esta especificación podrá realizarse previo aviso siempre y cuando exista un consenso entre el profesor y TODOS los grupos.

4. Entregables

Como entregables en este proyecto se evaluará lo siguiente:

- Presentación funcional completa (50 %). Se evaluará según rúbrica correspondiente.
- Paper (máximo 4 páginas) 25 %
 - Abstract - 2 %
 - Introducción - 5 %
 - Sistema desarrollado - 5 %
 - Resultados - 5 %
 - Conclusiones - 5 %
 - Referencias - 3 %
- Documentación de diseño (25 %)
 - Documento de diseño de software: Deberá contar con toda la documentación del desarrollo de software en el sistema: diagramas de clase, UML, descripción de métodos, bibliotecas, API, etc. Incluir requisitos de software del sistema (extraídos de especificación y profesor) y lista de chequeo de cumplimiento de los mismos. (10 %)
 - Metodología de diseño de sistema: Deberá detallar la metodología de diseño utilizada en el proyecto que involucre el análisis del problema, investigación respectiva, propuestas de diseño, comparación y evaluación de propuestas. (10 %)
 - Herramientas de ingeniería: Deberá detallar el uso de las principales herramientas involucradas en el proyecto, así como todo modelo, ecuación, script, y herramienta en general que el grupo haya creado o modificado para solucionar el problema planteado. (5 %)