



Pro-CORE

En el Laboratorio de Comunicación y Redes (LACORE) de la Universidad Central de Venezuela, se está llevando a cabo una propuesta para el diseño de un procesador al cual desean llamar “**Pro-CORE**”, (**Procesador de LACORE**), sin embargo, la implementación de ésta arquitectura tiene un presupuesto muy ajustado, es por ello, que en el diseño se ha establecido la utilización de simulaciones, a fin de hacer todas las pruebas, iteraciones, verificaciones y validaciones posibles, antes de construir el procesador físicamente.

El programa de simulación de la arquitectura debe ser diseñado, desarrollado, probado e implementado en lenguaje **C**.

Entre los documentos aportados por LACORE, las especificaciones de Pro-CORE son:

Descripción de la Arquitectura Pro-CORE

Documento: LACORE-362

Características de Pro-CORE:

- 8 Registros de Propósito General. (Véase LACORE-423).
- 8 Instrucciones. (Véase LACORE-427).

Documento: LACORE-423

Identificadores de los Registros:

| Nombre | ID Decimal | ID Binario |
|--------|------------|------------|
| S0 | 0 | 000 |
| S1 | 1 | 001 |
| S2 | 2 | 010 |
| S3 | 3 | 011 |
| S4 | 4 | 100 |
| S5 | 5 | 101 |
| S6 | 6 | 110 |
| S7 | 7 | 111 |

Documento: LACORE-427

Repertorio de Instrucciones

| Instrucción | Descripción | Ejemplo |
|---------------|------------------------------------|----------------|
| rrmov SD, SF | $SD \leftarrow SF$ | rrmov S4, S8 |
| rimov SD, inm | $SD \leftarrow \text{inm}$ | rimov S5, 0x32 |
| mrmov dir, SF | $M(\text{dir}) \leftarrow SF$ | mrmov 0x31, S2 |
| rmmov SD, dir | $SD \leftarrow M(\text{dir})$ | rmmov S6, 0x12 |
| sum SD, SF | $SD \leftarrow SD + SF$ | sum S4, S0 |
| res SD, SF | $SD \leftarrow SD - SF$ | res S3, S8 |
| xor SD, SF | $SD \leftarrow SD \text{ XOR } SF$ | xor S1, S1 |
| fin | Fin del Prog. | fin |

- SD: Registro Destino.
- SF: Registro Fuente.
- dir: Dirección (8 bits).
- Inm: Inmediato (8 bits).
- Se requiere que dir e inm sean expresados en Hexadecimal (8 bits).

Documento: LACORE-342

Codificación binaria de las Instrucciones

| Instrucción | Codificación |
|-------------|--------------|
| rrmov | 000 |
| rimov | 001 |
| mrmov | 010 |
| rmmov | 011 |
| sum | 100 |
| res | 101 |
| xor | 110 |
| fin | 111 |

Documento: LACORE-142

Formatos de Instrucción

| Instrucción | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Bits (9-16) |
|-------------|-----|---|---|-----|---|---|-----|---|---|-------------|
| rrmov | 000 | | | SD | | | SF | | | |
| rimov | 001 | | | SD | | | 000 | | | Inmediato |
| mrmov | 010 | | | 000 | | | SF | | | Dirección |
| rmmov | 011 | | | SD | | | 000 | | | Dirección |
| sum | 100 | | | SD | | | SF | | | |
| res | 101 | | | SD | | | SF | | | |
| xor | 110 | | | SD | | | SF | | | |
| fin | 111 | | | | | | | | | |

- Nótese que hay instrucciones de 3 bits, 9 bits y de 17 bits.

Simulación de la Arquitectura Pro-CORE

El objetivo de cada grupo de proyecto es crear una aplicación que sea capaz de:

- Mostrar el estado del procesador.
- Desensamblar un programa, dada su codificación binaria.
- Ejecutar un programa completo, dada su codificación binaria.

Mostrar el estado del procesador

Esta opción muestra el estado completo del procesador, es decir, el contenido de los registros (S0 al S7), contenido de los registros de control y registros auxiliares que el equipo de desarrolladores haya decidido agregar.

Mostrar el estado de la memoria principal

Esta opción pide al usuario dos posiciones y muestra por pantalla el contenido de todas las posiciones de memoria en el rango dado. Las entradas deben ser dos (2) posiciones de memoria validas.

Desensamblar un programa, dada su codificación binaria

En esta opción, recibe como entrada un programa en binario y genera como salida su traducción en lenguaje ensamblador Pro-CORE. Por ejemplo, dado el programa anterior **100001000110000010111**, el programa debe mostrar, la traducción como.

| | |
|------------|---------------|
| sum S1, S0 | # 100 001 000 |
| xor S0, S2 | # 110 000 010 |
| fin | # 111 |

Mostrar los comentarios no es obligatorio, se colocaron para ejemplificar cual es el fragmento del programa binario que representa cada instrucción.

Ejecutar un programa, dada su codificación binaria

En esta opción se pedirá como entrada una cadena de unos y ceros, que representa todo el programa en binario. Por ejemplo, si el programa de entrada es **100001000110000010111**, el programa deberá, en primer lugar, sumar el contenido de S1 con S0 y dejar el resultado en S1, luego, hacer la operación XOR entre S0 y S2 y dejar el resultado en S0, por último finalizar el programa, como se muestra en el ejemplo anterior.

Requerimientos y Consideraciones de Implementación

- Cada implementación de **Pro-CORE** debe ser única, las copias serán severamente sancionadas y bajo la ley de universidades podría ocasionar la expulsión de la universidad.
- Cada equipo está en la libertad de crear los menues y las funcionalidades que considere necesarias. Sin embargo, el cumplimiento de las funcionalidades descritas es obligatorio.
- El programa debe compilar sin errores, ya que, de no ser así el proyecto se considerará “no entregado”.
- El Proyecto debe ser realizado individual o en parejas **DE LA MISMA SECCIÓN DE LABORATORIO**, no serán recibidos proyectos de personas que no cumplan este requisito.
- Como se dijo anteriormente, el proyecto debe ser implementado en el lenguaje C.
- Un aspecto a tomar en cuenta en la evaluación es la intradocumentación, por lo tanto es aconsejable que el código esté bien documentado.
- Cada grupo debe entregar un informe del proyecto.
- El programa debe mostrar un menú, el cual se debe tener como mínimo.
 - Mostrar el estado del procesador.
 - Mostrar el estado de memoria.
 - Desensamblar un programa, dada su codificación binaria.
 - Ejecutar un programa completo, dada su codificación binaria.

- Créditos (en el que se debe mostrar los nombres de los desarrolladores).
 - Salir (Opción para salir del programa).
- El programa deberá tener implementado el sistema de menú, validando que si el usuario se equivoca y marca una opción que no está en el menú principal, esto no cause la salida del programa, además, que cuando entre a ver los créditos, por ejemplo, tenga una opción clara de volver al menú principal. El programa sólo debe finalizar con la opción “Salir”.
- El programa deberá ser capaz de mostrar el estado del procesador, la memoria y los créditos.
- El programa deberá ser capaz de Permitir simular perfectamente el procesador **Pro-CORE**.

Fecha de Entrega: lunes 24 de Abril de 2017, sin prórroga (vía email).

Recuerde: Es **MUY IMPORTANTE** que antes de hacer cualquier consulta haya leído detalladamente el enunciado.

¡Éxito! – GDOECI