

Examen Introducción a los Sistemas Embebidos

Malcolm Davis, Miembro Estudiantil, IEEE
mdavis.cr@ieee.org

Resumen— Este documento contiene las respuestas teóricas así como, si es necesario, la justificación de las decisiones tomadas para resolver los problemas de desarrollo del examen final del curso Introducción a los Sistemas Embebidos del Tecnológico de Costa Rica. En el segundo semestre del 2018.

Index Terms— TEC, Sistemas Embebidos, Yocto, Codiseño, Drivers, Toolchain.

1. I PARTE. RESPUESTA BREVE

1.1. Mencione 3 características de los sistemas embebidos

Tres de las principales características de un sistema embebido son:

- La **funcionalidad**, donde incorpora elementos de Software y Hardware para realizar una función específica.
- La **complejidad**, que gracias a su especialización, son sistemas normalmente complejos.
- El **costo**, que no sólo incluye el costo de manufactura sino también toma en cuenta el costo energético.

1.2. Mencione 2 etapas del codiseño de sistemas embebidos

El codiseño de sistemas embebidos tiene varias etapas, dos de ellas son la **Especificación de Hardware**, **Especificación de Software** y la **Integración y depuración**.

1.3. Lucía trata de portear el paquete opencv para raspberry pi zero, utilizando Yocto. Al realizar el comando `bitbake rpi-basic-image`, tiene el siguiente error: "Nothing provides opencv" ¿Cómo podría solucionar el error Lucía?

Tomando en cuenta el error, todo parece indicar que Lucía no está incluyendo la receta para la compilación de opencv, esta receta se puede encontrar en algún layer de opencv dependiendo de la versión requerida en la página <https://layers.openembedded.org>.

1.4. ¿A qué comando, en espacio de usuario, corresponde el método `module_init(void)`?

El método en espacio de usuario que inicializa el driver es `insmod`, se debe de usar con permisos de administrador, por lo que su ejecución en un derivado de debian se vería de la siguiente forma: "sudo insmod module".

1.5. Mencione 2 métodos de compilación en sistemas embebidos

Si se refiere con métodos de compilación a herramientas que se utilizan para compilar, entonces está el **Make** y **auto tools**. Pero si se refiere a formas de compilar, se puede compilar **local** en el sistema embebido si este tiene las herramientas para la compilación, o se puede hacer una **compilación cruzada** desde otro dispositivo.

1.6. En el comando `$(CC) -o this isWrong.cpp -I../include -L../lib -lsomething`, ¿Cuál es el nombre del archivo completo de la biblioteca dinámica enlazada?

Cuando se utilizan bibliotecas dinámicas, se nombran como **libnombre.so** y se incluyen como **-lnombre**, por lo tanto el nombre del archivo es **libsomething.so**.

1.7. Escriba un comando (bash) que establezca el macro del compilador a gcc

El comando `export` establece variables de entorno, y el CC es el compilador; por eso `export CC=gcc` configura gcc como el compilador predeterminado para los archivos de C.

1.8. Mencione 3 niveles de abstracción en el diseño de sistemas embebidos

Tres niveles de abstracción de arriba a abajo serían, el **Sistema**, los **componentes del procesador** y el **circuito o transistor components**.

1.9. Mencione 3 tipos de modelos en el diseño de sistemas embebidos

Los 3 modelos vistos en clase fueron el **Modelo de Especificación**, el **Modelo de nivel de Transacción** y el **Modelo con precisión de ciclo**.

1.10. ¿Qué es un toolchain?

Un toolchain es un set de herramientas que compilan el código fuente para crear ejecutables que pueden correr en un dispositivo específico. Incluye compiladores, enlazadores, y bibliotecas entre otras cosas para realizar esta tarea.

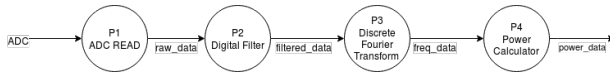


Figura 1. Modelo ADC

1.11. Al tratar de compilar un código fuente, Juanito tiene el siguiente error `undefined reference to 'sqrt'`. Juanito dice que `za incluyó math.h`, pero el error no se corrige. ¿Cómo solucionaría el problema Juanito?

El problema que tiene Juanito es un problema en la etapa de enlazado, por y esto es porque la biblioteca `math` es dinámica, al agregarla debe de agregar una bandera de compilación que se le envía al linker especificando que se debe de enlazar esa biblioteca con el código fuente. Agregar `-lmath` a la compilación debería de solucionar el problema.

1.12. ¿Cuál es la diferencia principal entre una biblioteca estática y una dinámica?

La principal diferencia entre ambas, es como se usan en el código de la aplicación a compilar. Las bibliotecas estáticas se agregan al ejecutable compilado, mientras que las dinámicas residen en el sistema y lo que se agrega al ejecutable es la dirección de la biblioteca para que se ejecute su código. Cada una tiene sus ventajas y desventajas, ya sea por portabilidad, espacio del ejecutable o otras.

2. PROBLEMA 1

Utilizando la información vista en clase, los pasos de la síntesis deberían de ser, **Perfilado** y **estimación**, **Selección de componentes y conexiones**, **Asignación de procesos y canales**, **Calendarización de procesos**, **Inserción de interfaces** y **Refinamiento del modelo**. Como es un proceso manual algunos de estos se pasan por alto o se ven con muy poca profundidad. A continuación se explica lo que se realizó en cada paso.

Perfilado En los módulos se realizan lecturas y escrituras de memoria(para el primer proceso), y en los buses para los demás, de estos hay al menos 3 por proceso. Además, manejan contadores, sumas restas y multiplicaciones, de estas hay al menos 4 por módulo, siendo los procesos 3 y 4 los que más hacen uso de estos recursos con aproximadamente 14 y 5 operaciones aritméticas por ciclo respectivamente. Dentro de los procesos, también se hacen llamados a funciones, por lo cual se van a necesitar herramientas para el control del flujo de ejecución, estos se hacen con menor frecuencia que las operaciones aritméticas, pero al menos los procesos uno, tres y cuatro hacen uso de 1, 1 y 3 llamadas a funciones por ciclo respectivamente.

Allocation Según se especifica, y tomando en cuenta la cantidad de operaciones aritméticas, se utilizará una ALU por proceso para agilizar la ejecución del sistema. La placa de desarrollo posee un dispositivo ADC que va a ser utilizado, y ocupa ser mapeado a memoria, así como las instrucciones. Además se necesitan buses para poder comunicar los datos entre procesos. En cuanto al sistema operativo y otros componentes que facilitarían la implementación del sistema, se puede obviar para poder utilizar la menor cantidad de componentes y así obtener un sistema mínimo.

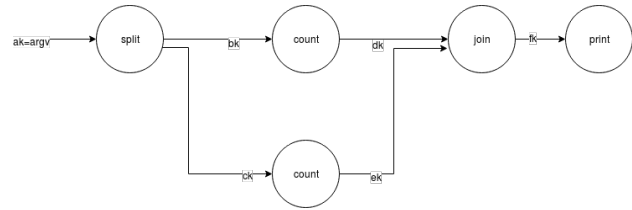


Figura 2. Modelo RZE

Binding Como se habló anteriormente, cada proceso va a tener su propia unidad lógica aritmética(aunque talvez luego de hacer pruebas, sea mejor utilizar sólo 3 unidades y que los procesos 1 y 4 compartan ALU, ya que las ALUs van a estar ociosas en el tiempo que se dura entre muestras. Cada proceso tiene 1 bus de salida y entrada que comparte con otro. Y como se mencionó anteriormente, los procesos 1 y 3 tienen acceso a memoria. Para los pasos restantes, se utilizan herramientas ya que son más complejos, y por eso no se ejemplificará en la solución de este problema.

3. PROBLEMA 2

Para este problema se implementó un modelo Kahn del algoritmo de encodificación Zero-Run. Los Procesos que se muestran en la figura 2 se detallan seguidamente.

- **Split:** Este módulo de hardware tiene como entrada el arreglo provisto por el usuario en una cola que, se transfiere elemento por elemento a los siguientes dos módulos.
- **Count:** Este módulo de hardware procesa la entrada, y aumenta un contador cada vez que recibe un cero, si el elemento recibido no es cero, entonces envía el valor del contador y luego lo resetea. Cuando el valor ingresado es cero, este modulo no tiene ninguna salida.
- **Delay:** Este módulo de hardware retrasa la salida hasta que tenga un valor que no sea cero.
- **Join:** Este módulo de hardware espera a tener un valor tanto del módulo Count como el Delay para dar un resultado, que es, en orden el valor de la cuenta y luego el valor del delay.
- **Print:** Este módulo sólo muestra el resultado en pantalla, cada vez que recibe un dato lo imprime en pantalla.

REFERENCIAS

- [1] R. Hamming and J. Kaiser. *You and Your Research*. Morristown, New York: Bell Communications Research, 1986.