

CIIC 4082 – Arquitectura de Computadoras II

Diseño de Emulador de CPU

Borrador

Introducción

El MSP430FR6989 Launchpad provee múltiples opciones para operaciones de entrada y salida. Entre ellas se encuentra la pantalla LCD y entrada de datos por botones. En esta actividad se diseñará e implementará una aplicación que hará uso estas herramientas.

Descripción General del Sistema

En la primera tarea del curso diseñamos un sistema con un CPU (CPURUM en adelante), memoria y dispositivos de I/O. En esta tarea utilizaremos un sistema con un MSP430FR6989 Launchpad que se comporte de forma similar al CPU que diseñamos en la primera tarea.

El objetivo de esta tarea es que, si cargamos un programa en la memoria del MSP430 como el que cargaríamos en la primera tarea, los resultados que obtengamos sean lo más similares posible en ambos casos.

Arquitectura general

La versión del CPURUM que implementaremos será una versión mejorada con un set de instrucciones más amplio. A continuación, describimos la arquitectura general y detalles sobre la emulación.

Los primeros 16 bytes de la memoria RAM del MSP430 se utilizarán para representar la memoria RAM del CPURUM. Los registros R10, R11, R12 y R13 realizarán, respectivamente las funciones de registros R0, R1, R2 y R3 del CPURUM. Así que en cualquier momento los registros R10-R13 contendrán lo que contendrían R0-R3.

El set de instrucciones se actualizó y el resumen de las instrucciones se presenta en la Tabla 1.

Tabla 1: Set de instrucciones en lenguaje de máquina

Instr	Formato	Operación	Comentarios
Load	00ddaaaa	$R[dd] \leftarrow \text{Mem}[aaaa]$	Cargar registro dd con contenido de memoria aaaa
Store	01ssaaaa	$\text{Mem}[aaaa] \leftarrow R[ss]$	Guardar contenido de registro ss en memoria aaaa
Add	10ddsxxx	$R[dd] \leftarrow R[dd] + R[ss]$	Suma de registros
Neg	110000ss	$R[ss] \leftarrow 2's \text{ comp } R[ss]$	Complemento de dos del contenido del registro

Shr	110001ss	R[ss] <- Shift right R[ss]	Shift hacia la derecha del contenido del registro
In	110011dd	R[dd] <- S2S1	Recibe un valor de dos bits formado por el estado de los bits correspondientes a los botones S2 y S1
Out	1110ssaa	Output[aa] <- R[ss]	Enviar el contenido del registro ss al dispositivo de salida con dirección aa
JR30	1111aaaa	If R3 == 0 -> PC <- aaaa	Brinca a la dirección indicada si el valor de R3 es 0

*Debajo de la sección del código que representa la memoria del CPURUM definirá la referencia

runningInDebugger db 0 //1 if you are going to test it using the debugger. 0 otherwise

Esto se hará para facilitar probar el emulador por parte de los evaluadores. Si se probará utilizando el debugger el programa tiene que caer directamente en el comienzo de la emulación y no habrá un retraso de 3 segundos luego de activarse el icono R para que se realice la operación de lectura. R se encenderá en el momento en que corresponde y se apagará con la ejecución de la próxima instrucción.

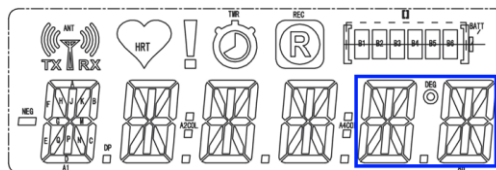
Durante la ejecución con el debugger, si se utiliza la instrucción Out para escribir a la dirección 2, el valor aparecerá en el despliegue, pero no se decrementará su valor.

Instrucción In

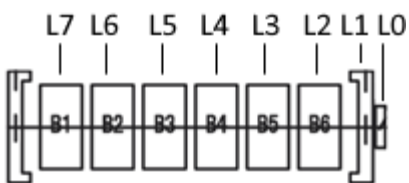
Cuando se ejecute la instrucción In el ícono R en el despliegue se encenderá por aproximadamente tres segundos. Esto le indicará al usuario que próximamente se realizará una lectura del estado de los botones S1 y S2. Luego de transcurrido este tiempo el procesador realizará la lectura y apagará R.

Despliegues

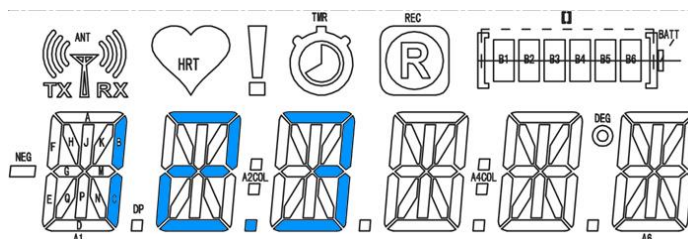
El despliegue hexadecimal del CPURUM se presentará en los dos caracteres de la extrema derecha del despliegue del Launchpad.



Los 8 bits del despliegue LED se presentarán con 8 segmentos de la batería del forman la batería. Se utilizarán los segmentos B1 a B6, el extremo derecho y el polo positivo para representar cada uno de los LEDs del L7 al L0 como se muestra a continuación.



Se añade un despliegue adicional asociado a la dirección 2 (10b) de salida. Este despliegue ocupará los tres caracteres alfanuméricos de la extrema izquierda del Launchpad. Cuando se realice una operación de *Out* a la dirección 2 ocurrirá lo siguiente: el valor que se envíe, se escribirá, en decimal en el despliegue. Así que, si el contenido en el registro de origen es 0x7B, como esto representa 123 decimal en el despliegue aparecerá lo siguiente:



El 123 es el número de décimas de segundos de un temporizador (timer). Una vez se escriba el valor en la pantalla el conteo comenzará a decrementarse. El conteo del tiempo se realizará utilizando un temporizador del MSP430 generando interrupciones para que se actualice la representación en el despliegue. Así que, si el registro de origen tenía un 123, en el despliegue aparecerá un 12.3 que se comenzará a decrementar y aparecerá 00.0 en el despliegue al cabo de 12.3 segundos. Si antes de llegar a 00.0 se escribe nuevamente en la dirección 2 con una instrucción de *Out*, el nuevo conteo de tiempo aparecerá en la pantalla y el proceso comenzará desde ese valor.

Inicio de la ejecución

Al inicio de la aplicación aparecerá en pantalla el nombre de la misma: CPURUM. Luego, al presionar el S2 se mostrarán los nombres de los integrantes del equipo. Si el nombre de un integrante contiene más de seis caracteres se mostrarán sólo los primeros seis. Cada vez que se presione S2 se cambia a presentar el próximo integrante. Si se llega al final de la lista se siguen presentando los nombres de forma circular.

Al presionar S1 aparecerá en el despliegue el mensaje S2TOGO. Esto indica que al presionar S2 comenzará la ejecución del programa para el CPURUM.

Requisitos Técnicos

La lectura del estado de los botones hasta el punto de presionar S2 para comenzar la emulación se realizará utilizando interrupciones. La lectura de los botones para la ejecución de la

instrucción In se realizará utilizando polling. Hasta antes de comenzar la fase de emulación el microcontrolador estará en un estado de mínimo consumo de potencia todo el tiempo que no esté manejando un proceso asociado a presionar un botón.

El programa tiene que ser modular, dividido en subrutinas. Cada subrutina terminará con un ret y en la medida en que sea posible operará sobre valores que recibirá por medio de registros o del stack.

No se realizará brincos de una subrutina a otra por medio de instrucciones que no sean call.

Reporte

Con la posible excepción de la portada estará escrito con letra (font) Calibri 12.

Portada: Título, curso, profesor, integrantes, fecha

Proceso de solución del problema: Cómo se realizó el análisis del problema, se determinaron las tareas, qué tuvieron que buscar e investigar (con referencias). 1.5 a 3 páginas

Distribución de tareas: Tabla mostrando las tareas que realizaron (investigaciones, pruebas, algoritmos, creación de código, etc.) y los integrantes del equipo a que realizaron las mismas.

Ejemplo:

Tarea	Persona a cargo	Comentarios
Diseñar la subrutina que va presenta un string en el despliegue del Launchpad	Pepito Pérez	

Código del programa: Copia del archivo con el código del programa. Letra Courier New o Consolas en un archivo .txt.

Nota: El reporte se entregará en formato compatible con MS-Word (.docx o .doc). El nombre del archivo con el reporte será CPU##SuNombre en donde ## es el número del grupo utilizando dos dígitos. SuNombre es su primer nombre seguido de su segundo apellido. Para su nombre no utilice segundo nombre ni segundo apellido. No utilice acentos, diéresis, tildes ni carácter alguno que no sean letras

Entrega de programa

En un comentarios en la primera línea del archivo con el código del programa indicará qué modificación hay que hacer en el programa para correr el mismo en el debugger o fuera del mismo.

En las líneas siguientes presentarán en comentarios los nombres de las personas que **contribuyeron** con el desarrollo del trabajo y su número de ID.

Documentación de las subrutinas

Objetivo: Propósito de la subrutina. Debe hacer sólo una tarea

Precondiciones: Lo que siempre tiene que ser cierto antes de la ejecución de la subrutina para que la misma funcione de manera correcta

Postcondiciones: Lo que siempre será cierto luego de ejecutar la subrutina

Autor/a: Persona que diseñó la subrutina

Fecha: Cuándo se creó esta versión de la subrutina

Ejemplo

;Objetivo: Presentar un dígito en una posición específica del despliegue

;Precondiciones: El dígito se encuentra en R5 y la posición en R6. La primera posición de la izquierda se representa con 1.

;Postcondiciones: El dígito contenido en R5 se verá en la posición R6 en el despliegue.

;Autor: Pepito Pérez

;Fecha: 10/abr/2022

Entregará el programa de la siguiente forma. En un archivo de texto (.txt) tendrá el código completo del programa. Este archivo se utilizará para copiar el código a un programa nuevo en el IAR y probar el funcionamiento. Si el programa copiado de esta forma no funciona no recibe puntos en la tarea. El nombre de este archivo será CPU##SuNombre. En adición entregará el folder que contiene todos los folders y archivos de su workspace. Comprimirá el folder y el archivo tiempo texto en un solo archivo zip (NO utilice otro formato como .rar, .7z etc.) con el nombre CPU##SuNombre.

Entrega de vídeo

Vídeo de la operación de la aplicación: Muestra que el trabajo cumple con todos los requisitos. El vídeo mostrará el funcionamiento de todas las instrucciones, manejo de dispositivos de entrada y salida y que el funcionamiento cumple con los requisitos. Sólo mostrará la operación, no el código. Al inicio del vídeo tienen que aparecer (la persona, no el nombre) cada integrante del equipo que contribuyó con la tarea y mencionar el nombre

Para crear el vídeo considere el uso de una aplicación como ScreenCast-O-Matic que es gratuita, corre en la nube y le permite grabar simultáneamente la pantalla de la computadora, la cámara y el audio.

Tome en consideración que es posible que el dispositivo que utilice para tomar el vídeo tenga más resolución de la que hace falta. Así que verifique cuál es la resolución más baja con la cual puede tomar un vídeo que sea vea bien. Asegúrese de hablar en voz alta y clara. No se preocupe porque haya un poco de ruido de fondo o porque cometa algunas equivocaciones al hablar (se requiere que el vídeo esté

en una sola toma, sin edición). El tiempo máximo que puede durar el vídeo es 8 minutos. No hay problema en que lo haga en menos tiempo. El archivo tiene que estar en formato .mp4 y el nombre tiene que ser CPU##SuNombre. El archivo con el vídeo lo subirá a la nube y proveerá el enlace (que no requiera password) al mismo en la página de portada del trabajo. No enviará el archivo del vídeo con el trabajo. Si el enlace en el reporte no funciona no tendrá los puntos correspondientes al vídeo.

Hoja de Autoevaluación y Evaluación de los Pares

Entregará el archivo de Excel completado. El nombre del archivo será CPU##SuNombre.

Evaluación

- [4]Estructura general del reporte
- [5]Proceso de solución
- [5]Distribución de tareas
- [10]Video
- [5]Presentación y navegación en interfaz
- [5]Entradas manejadas con interrupciones
- [3]MCU en Low Power y sólo despierta para INTs
- [7]Dividido correctamente en subrutinas
- [7]Documentación
- [30] Manejo temporizador con instrucción Out.
- [39]Operación general sin incluir manejo de temporizador
- [120]Total

Entrega del trabajo

Comprimirá todos los archivos en uno solo de tipo zip con el nombre CPU##SuNombre.

Sugerencias

La tarea está diseñada para que se pueda dividir en módulos (subrutina) de forma que facilite el trabajo en equipo. Tome su tiempo para analizar los módulos en que se puede dividir y nuevamente los módulos en los que se pueden dividir los primeros módulos. Considere pasar los valores a las subrutinas por medio de registros. De esta forma las hará independientes de nombres globales y facilitará el trabajo independiente de los integrantes del equipo.

Presuma que no quedará conforme con las primeras 10 versiones del vídeo y que el último día para entregar la tarea la conexión a internet estará lentísima.