GEM51: Emulador de microcontrolador 8051

Autor: Juan Guido Salaya
Facultad de Ingenieria, UBA

Resumen:

GEM51 es una poderosa herramienta que permite emular **en tiempo real** el funcionamiento de un microcontrolador 8051/52 y su interacción con los periféricos.

Puede emular los siguientes periféricos: leds, displays 7 segmentos, pulsadores y displays LCD gráficos.

Mediante la utilización de las bibliotecas OpenGL (Open Graphics Library), los periféricos son visualizados en 3D.

La interfaz con el usuario es agradable e intuitiva, permitiendo arrastrar y colocar los elementos con el mouse, así como modificar sus características abriendo un menú con el botón derecho.

Mediante la consola, es posible ejecutar comandos y modificar o mostrar variables o posiciones de memoria. Por ejemplo, es posible cambiar la frecuencia de reloj del microcontrolador o ver la información de los componentes en el proyecto.

Permite implementar PWM (modulación de ancho de pulso) aplicado a leds y displays 7 segmentos (modelados como leds), multiplexado de displays 7 segmentos, conexión de teclado matricial. Además, es capaz de simular pulsadores ideales o con rebote.

Los requerimientos son mínimos, pudiendo correr en cualquier computadora actual. Requiere OpenGL y Windows XP. Se recomienda un procesador de 1GHz. Utiliza menos de 16MB de RAM.

La aplicación fue probada en una computadora con un procesador Intel® Core™ Solo Processor T1300 (2M Cache, 1.66 GHz, 667 MHz FSB), pudiendo emular el microcontrolador con una gran cantidad de periféricos (mas de 100) a velocidades muy superiores a las típicas.

GEM51 esta siendo utilizado actualmente en la Facultad de Ingeniería de la Universidad de Buenos Aires en la materia “Laboratorio de Microcomputadoras” para mostrar el funcionamiento del microcontrolador y emular programas sin la necesidad de tener que armar ningún tipo de hardware. De esta forma, los alumnos pueden desde el comienzo programar el microcontrolador y elementos periféricos sin necesidad de involucrarse en la conexión física de los mismos.

Introducción:

Las circunstancias con las que nos encontramos hoy en el campo de los microcontroladores tienen sus raíces en el desarrollo de la tecnología de los circuitos integrados. Este desarrollo ha hecho posible contener cientos de miles de transistores en un solo chip. Ese era uno de los requisitos previos para la producción de los microprocesadores, y las primeras computadoras eran hechas agregando periféricos externos como la memoria, timers, etc. lo que aumentaba el volumen de los circuitos integrados. Estos circuitos integrados contenían procesador y periféricos. Así es cómo se desarrolló el primer chip que contenía una microcomputadora, o lo que después se llegaría a conocer como un microcontrolador.

Un microcontrolador es, esencialmente, una computadora llevada a un tamaño reducido, con menores prestaciones, pero a su vez con un muy bajo costo.

Los microcontroladores tienen diversas aplicaciones, como por ejemplo en juguetes, electrodomésticos, equipos de audio y video, automóviles, aviones, etc.

Un sistema embebido o empotrado es un sistema de computación diseñado para realizar una o algunas pocas funciones dedicadas frecuentemente en un sistema de computación en tiempo real. Los sistemas embebidos se utilizan para usos muy diferentes a los usos generales a los que se suelen someter a las computadoras personales. En un sistema embebido la mayoría de los componentes se encuentran incluidos en la placa base. Su componente principal es el microcontrolador.

El Intel 8051 es un microcontrolador (µC) desarrollado por Intel en 1980 para uso en sistemas embebidos. Es un microcontrolador muy popular.

Los núcleos 8051 se usan en más de 100 microcontroladores de más de 20 fabricantes independientes como Atmel, Dallas Semiconductor, Philips, Winbond, entre otros.

La denominación oficial de Intel para familia de µCs 8051 es MCS 51.

Actualmente existen diversas herramientas de desarrollo de software para sistemas embebidos con microcontroladores MCS 51. Algunos ejemplos de ellos son: compiladores, ensambladores, debuggers, simuladores. También existen entornos de desarrollo integrados.

En el campo de los simuladores, existe una gran variedad. Por ejemplo, algunos pueden mostrar el estado de las variables, memoria y puertos. Otros pueden simular la interacción con displays 7 segmentos, leds, displays lcd. Sin embargo, la mayoría **no trabaja en tiempo real**. Existen unos pocos emuladores que pueden trabajar en tiempo real y presentan la desventaja de tener una interfaz complicada y por lo general son muy costosos.

Por este motivo se decidió desarrollar un emulador de 8051 que implemente las mejores prestaciones de cada uno.

Objetivos:

- Desarrollar un programa que emule el funcionamiento de un microcontrolador 8051 de manera precisa, utilizando un modelo muy similar al hardware real.
- Lograr una emulación en tiempo real, programando de manera eficiente en un lenguaje orientado a objetos que administre adecuadamente los recursos del sistema. Se escogió el lenguaje C++.
- Visualizar los periféricos en tres dimensiones utilizando las bibliotecas OpenGL.
- Procesar los datos entre cada actualización de pantalla para los periféricos, permitiendo una correcta comunicación. Utilizar modulación de ancho de pulso para modificar la intensidad de los elementos emisores de luz. Esto hace posible el multiplexado de displays 7 segmentos.
- Modelar pulsadores de tal forma que permitan su interconexión formando un teclado matricial. Además permitir la simulación del efecto rebote de los mismos.
- Introducir una consola para establecer una interfaz por línea de comando, permitiendo ejecutar comandos avanzados y ver o modificar variables, parámetros o datos en memoria.
- Proveer una interfaz con el usuario simple e intuitiva, permitiendo arrastrar y colocar los periféricos y modificando sus propiedades y parámetros mediante el uso de un menú, al cual se accede presionando el botón derecho del mouse sobre el elemento a modificar.

Materiales e implementación

La aplicación fue desarrollada en el entorno de desarrollo Microsoft Visual Studio 6.0 en una computadora Lenovo T60 con un procesador Intel® Core™ Solo Processor T1300 (2M Cache, 1.66 GHz, 667 MHz FSB) con 1GB de RAM.

Se utilizó el entorno de desarrollo Keil uVision 2 para realizar los programas de prueba para luego ser simulados y comprobar la validez del modelo.

Pruebas preliminares:

Con el fin de evaluar la viabilidad del proyecto y el cumplimiento de los objetivos propuestos, fue preciso realizar previamente algunas pruebas.

En primer lugar, se evaluó la capacidad del emulador de realizar la simulación en tiempo real. Para evaluar la posibilidad de hacerlo y habiéndose implementado previamente algunas instrucciones, la aplicación debe emular a una velocidad mayor de la deseada. Es decir, si el microcontrolador debe ejecutar una cierta cantidad de ciclos de máquina en un determinado tiempo, el emulador debería ser capaz de ejecutar la misma cantidad de ciclos de máquina en un tiempo menor, para que así la velocidad sea mayor.

Para estos microcontroladores se utiliza típicamente un cristal de 12MHz, dando como resultado un ciclo de máquina de 1us. Para esta prueba se programó una función que ejecute el equivalente a 1 segundo, es decir 10^6 ciclos de máquina, calcule el tiempo transcurrido y luego, la máxima frecuencia de reloj. El resultado fue superior a 100MHz, lo que indicó que aun era posible la emulación en tiempo real.

Otro de los objetivos era procesar los datos entre las actualizaciones de pantalla para los dispositivos visuales. El más simple de ellos es el led, por lo cual fue el primer periférico implementado.
La potencia media en un intervalo de tiempo de N ciclos de máquina se puede calcular como: $\frac{N_1}{N} \cdot P_0$, siendo P_0 la potencia máxima. Para lograr esto, se debía muestrear el valor de salida del pin del microcontrolador al cual estaba conectado el led luego de cada instrucción. Sin embargo, al incrementar la cantidad de leds, la emulación se tornaba más lenta, dejando de funcionar en tiempo real. Por lo cual esto no era viable.
Para solucionar este problema, se optó por una aproximación. La opción era muestrear el valor del pin conectado al led cada T ciclos de máquina, lo cual reduciría notablemente la complejidad computacional agregada. Luego de algunos ensayos, se comprobó que la velocidad se pudo incrementar notablemente y la aproximación era muy buena y la diferencia era imperceptible.

Modelos utilizados:

Para poder emular correctamente, fue preciso modelar el microcontrolador y los periféricos con los que debe interactuar.

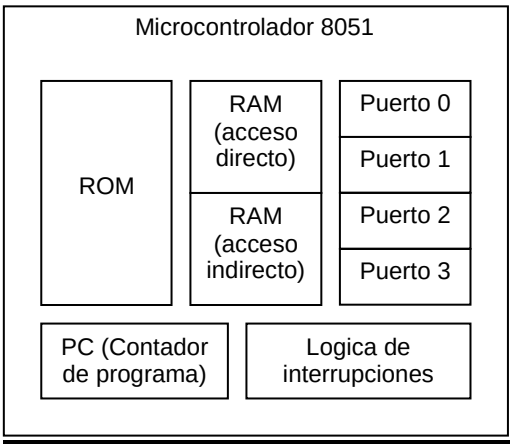
Microcontrolador:

Variables de estado:

El microcontrolador es un circuito sincrónico, es decir que su estado esta definido por un conjunto de variables. Para hacer un modelo adecuado, se implementó una clase con las mismas variables que el microcontrolador real.

El modelo del microcontrolador tendrá los siguientes elementos:

- Memoria ROM
- Memoria RAM de acceso directo
- Memoria RAM de acceso indirecto
- Puertos
- PC (contador de programa)
- Lógica de interrupción



Lógica de funcionamiento

El modelo implementa la misma lógica de funcionamiento que el hardware real. Se basa en el ciclo de instrucción (también llamado ciclo de fetch-and-execute o ciclo de fetch-decode-execute en inglés).

Además del ciclo de instrucción, el microcontrolador resuelve simultáneamente por hardware la lógica de temporizadores e interrupciones. Al emular el microcontrolador, esta lógica debe resolverse por software.
Por lo tanto el ciclo de instrucción quedaría de la siguiente forma:

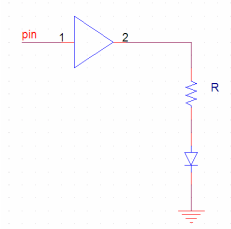
- Resolver lógica de temporizadores.
- Resolver lógica de interrupciones.
- Buscar la instrucción en la memoria de programa (ROM)
- Incrementar el contador de programa
- Decodificar la instrucción.
- Ejecutar la instrucción.

Led:

Es un dispositivo semiconductor que emite luz cuando se polariza de forma directa y circula por él una corriente eléctrica.

Circuito equivalente:

El circuito equivalente supone un driver ideal, es decir que no hay corriente desde el micro hacia el led o viceversa. La resistencia define la corriente que atravesara el led y su intensidad lumínica.



Variables de estado:

Las variables de estado que definen a un led son las siguientes:

- Posición
- Tamaño
- Color
- Resistencia
- Caída de tensión (V_0)
- Corriente para la cual se alcanza la potencia máxima (I_0)
- Ciclo de trabajo (cantidad de muestras encendido/cantidad de muestras totales)
- Pin al cual esta conectado

Lógica de funcionamiento

La simulación del led consiste simplemente en calcular la intensidad lumínica del mismo, en base a las siguientes variables:

- V_0: Caída de tensión en el led
- I_0: Corriente para la cual se alcanza la potencia máxima
- R: Resistencia del circuito equivalente
- N_1: Cantidad de muestras en las que el pin estuvo en nivel alto (5V)
- N: Cantidad de muestras totales

Recorriendo la malla del led, la corriente I se puede calcular como:

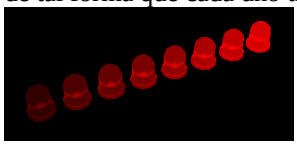
$$I = \frac{5 - V_0}{R}$$

Si asumimos la potencia máxima como 1, entonces, la potencia media se puede calcular como:

$$P = \frac{N_1}{N} \cdot \frac{I}{I_0} = \frac{N_1}{N} \cdot \frac{\frac{5 - V_0}{R}}{I_0} = \frac{N_1}{N} \cdot \frac{5 - V_0}{R \cdot I_0}$$

Visualización:

En la siguiente imagen se pueden observar 8 leds con modulación de ancho de pulso, de tal forma que cada uno tiene una intensidad distinta.

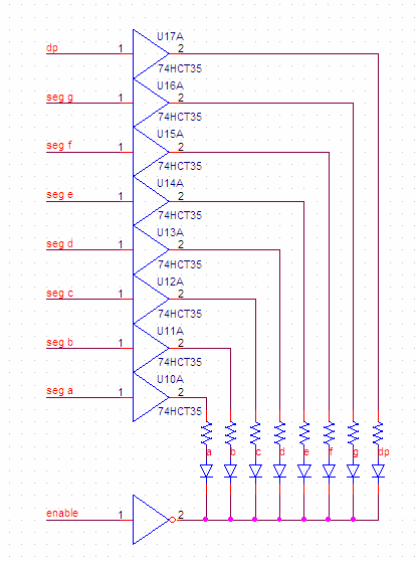


Display 7 segmentos:

Son utilizados generalmente para representar números en equipos electrónicos. Está compuesto de siete segmentos que se pueden encender o apagar individualmente. Cada segmento tiene la forma de una pequeña línea. Internamente están constituidos por una serie de leds.

Circuito equivalente:

Los displays 7 segmentos están modelados como 8 leds, uno por cada segmento y otro para el punto decimal. Además, tiene una línea de habilitación (enable), la cual habilita o deshabilita todos los leds.



Variables de estado:

Además de contener las variables de estado de los leds que conforman cada segmento, contiene las siguientes:

- Posición
- Tamaño

- Color de fondo
- Pin de habilitación (enable)

Lógica de funcionamiento

Al estar modelado como 8 leds, la lógica de funcionamiento es igual a la de 8 leds separados. Se agrega el pin de habilitación para posibilitar el multiplexado de varios displays.

Visualización:

En la siguiente imagen se puede apreciar 4 displays 7 segmentos multiplexados, es decir que se activan de a uno por vez, pero con una velocidad tal que se ven como si estuvieran encendidos todos simultáneamente:

Pulsador:

Un botón o pulsador es un dispositivo utilizado para activar alguna función. Un botón de un dispositivo electrónico funciona como un interruptor eléctrico, es decir en su interior tiene 2 contactos, los cuales se interconectan al ser pulsado.

Variables de estado:

Las variables de estado que definen a un pulsador son las siguientes:

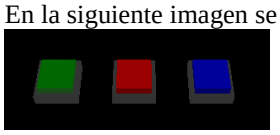
- Posición
- Tamaño
- Color del pulsador
- Color de fondo
- Tecla asociada
- Tiempo de rebote
- Pines a los cuales esta conectado

Lógica de funcionamiento

La lógica de funcionamiento de un pulsador sin rebote es simplemente interconectar los pines a los cuales esta conectado. En el caso de tener rebote, al presionarlo o soltarlo, quedará aleatoriamente conectado o desconectado durante un determinado tiempo y luego se estabilizará.
Como las salidas de los puertos del 8051 pueden recibir una mayor cantidad de corriente en estado bajo que la que pueden entregar en estado alto, al conectar una salida en estado alto con una en estado bajo, ambas quedarán en estado bajo.

Visualización

En la siguiente imagen se puede observar 3 pulsadores:



Display gráfico

Se modeló según el display gráfico WG12864 LCD (cristal líquido) de 128 x 64 pixeles, el cual posee un controlador Samsung KS0108 y KS0107. Se recomienda leer las hojas de datos para conocer mejor su funcionamiento.

Variables de estado:

Al igual que al modelar el microcontrolador, el display elegido es un circuito secuencial sincrónico y se modeló con las mismas variables de estado que el dispositivo real. El display se divide en dos secciones idénticas, las cuales son controladas independientemente. Sus variables de estado son:

- Posición
- Tamaño
- Color del backlight (modelado como un led)
- Pines a los que está conectado
- Variables de estado de cada una de las secciones:
    - Memoria RAM de visualización (donde se guarda lo que se debe mostrar)
    - Posición x,y en el display
    - Entrada de datos o comando
    - Línea de comienzo
    - Bit de encendido/apagado

Lógica de funcionamiento

De acuerdo con las hojas de datos del display y su controlador, el display recibe instrucciones cuando se encuentra un flanco positivo en la línea E. Por lo tanto, deben monitorearse los flancos del pin al cual está conectada en cada ciclo de máquina. En caso de encontrar un flanco, para la mitad que se encuentra seleccionada, se procederá a leer el dato del puerto al cual esta conectado el bus de datos del display y se interpretará como comando o se escribirá en la memoria (según el estado del pin D/I).
La lógica para el backlight es la misma que para los leds y respeta las especificaciones de la hoja de datos del display.

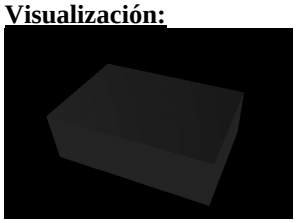
Visualización

En la siguiente imagen, se puede observar un display gráfico mostrando una imagen.

Gabinetes

Es posible colocar gabinetes similares a los gabinetes plásticos utilizados comúnmente para proyectos con estos microcontroladores. Su función es solo estética.

Visualización:

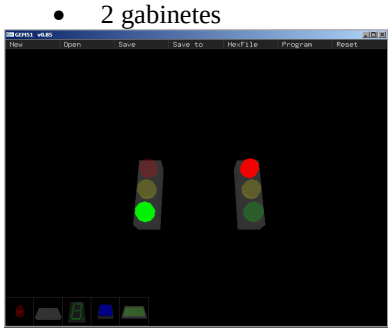


Resultados de las simulaciones:

Semáforos:

Una simulación sencilla es la de dos semáforos sincronizados.
Para esto fueron utilizados:

- 6 leds
- 2 gabinetes



Los intervalos de tiempo fueron los esperados y la visualización es correcta.

Modulación de ancho de pulso

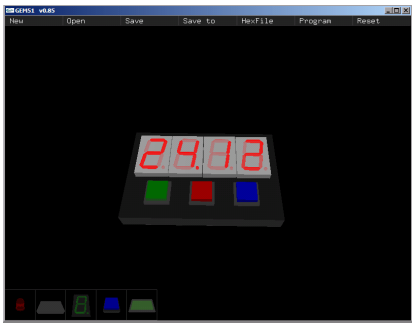
En esta simulación se utilizó una modulación de ancho de pulso diferente para cada uno de los 8 leds.

El resultado es el esperado, cada led tiene una intensidad proporcional a su ancho de pulso.

Cronómetro

Esta simulación consiste en un cronómetro de 4 dígitos, 2 para los segundos y 2 para las centésimas. Además contiene 3 pulsadores. Uno para comenzar la cuenta, otro para detenerla y el último para volver a cero.
Para esta simulación se utilizaron:

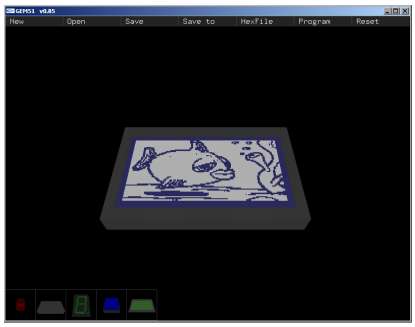
- 4 displays 7 segmentos (multiplexados)
- 3 pulsadores
- 1 gabinete



La simulación fue satisfactoria, el multiplexado de los displays funciona correctamente y se respetan los intervalos de tiempo.

Display gráfico:

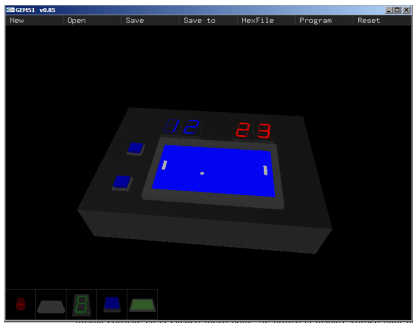
Esta simulación consiste en mostrar una imagen en un display gráfico.



El display gráfico responde normalmente a los comandos y datos enviados por el microcontrolador.

Pong!

Por último se realizó una prueba combinando todos los elementos que son capaces de simular por GEM51. La misma consiste en el clásico juego pong. Se utilizó el display gráfico para mostrar las paletas y la pelota, 4 displays 7 segmentos para visualizar los puntajes y 2 pulsadores para mover la paleta izquierda. La paleta derecha está controlada por el microcontrolador.



Aplicación en la enseñanza:

GEM51 esta siendo utilizado actualmente en la Facultad de Ingeniería de la Universidad de Buenos Aires en la materia “Laboratorio de Microcomputadoras” para mostrar el funcionamiento del microcontrolador y emular programas sin la necesidad de tener que armar ningún tipo de hardware. De esta forma, los alumnos pueden concentrarse más en la programación del microcontrolador.

Conclusiones

Se ha logrado cumplir con todos los objetivos propuestos.
Las simulaciones fueron satisfactorias tanto en los programas más simples como en los más complejos.
Los leds responden según lo esperado a la modulación de ancho de pulso y la visualización de los displays 7 segmentos multiplexados es satisfactoria.
El display gráfico responde adecuadamente a los comandos y datos suministrados por el microcontrolador.
El uso en las clases es de gran ayuda para la cátedra de “Laboratorio de microcomputadoras” en la Facultad de Ingeniería de la Universidad de Buenos Aires, ya que disponiendo de un proyector, se puede mostrar el funcionamiento de los programas a toda la clase.

Bibliografía

- *"The 8051 Microcontroller"*, Second edition, Scott McKenzie, Prentice-Hall, 1995.
- *"Digital design, Principles and Practices"*, Second edition, Wakerly J., Prentice Hall.