

Simulador y depurador para la plataforma SISAF

Julio del 2005

Autor: Christian Pérez Llamas <christip@ac.upc.edu>

Coordinador: Juan J. Navarro Guerrero <juanjo@ac.upc.edu>

1. Introducción

SISAF es la plataforma SISA que se utiliza en la asignatura EC1 del plan de estudios de la FIB, se trata de un procesador RISC de 16 bits con 8 registros de propósito general y 8 específicos para coma flotante, con acceso a memoria lineal a nivel de bytes sin protección ni modo sistema. Está conectado mediante un sencillo bus en entrada/salida con un conjunto de dispositivos que permiten al alumno practicar con la pantalla, el teclado, la impresora, el reloj y el disco duro.

Se ha desarrollado un simulador de dicha plataforma integrado junto con una interficie de depuración. De esta forma es posible depurar las aplicaciones desarrolladas al tiempo que se ven los resultados de la simulación por pantalla. Permite varias formas de avance, por instrucciones, con puntos de interrupción, de forma continuada, etc. Además tiene varias vistas del sistema que permiten inspeccionar el desensamblado de la memoria, su contenido en diversos formatos, los registros del procesador y de los dispositivos de E/S, el código fuente, el trazado de ejecución con sus tiempos, inspección de variables y puntos de memoria, la pantalla, la impresora, etc.

Un entorno ideal para la práctica y puesta a punto de los conceptos adquiridos en las clases teóricas, como soporte de las prácticas de laboratorio.

2. Requisitos del sistema

- Sistema compatible con UNIX

- Para la compilación:
 - Herramientas de compilación de GNU: make, gcc.
 - libncurses
 - libpthread
 - sisa-binutils

3. Compilación e instalación

Para la compilación del código fuente, además de disponer de las librerías previamente mencionadas, se requiere una determinada estructura de directorios para que encaje con sisa-binutils.

En concreto es aconsejable crear un directorio donde se extraigan sendos paquetes, sisa-binutils se deberá extraer en un subdirectorio con el nombre binutils, y el sisa-dbg en otro subdirectorio al mismo nivel que el anterior con el nombre que se desee. Quedaría algo así:

```
.../compilacion
.../compilacion/binutils
.../compilacion/dbg
```

A continuación se debe compilar e instalar binutils, para ello ejecutaremos las siguientes instrucciones:

```
$ cd .../compilacion/binutils
$ ./configure --target=sisa --prefix=[lugar donde se instalará, si no se especifica nada en /usr]
$ make
$ make install
```

Según dónde hayamos especificado que lo queremos instalar, necesitaremos permisos de root o no. Otro detalle a tener en cuenta es incluir en la variable de entorno del PATH dicha ruta.

Con estos comandos ya tendremos instalado el paquete de los binutils que incluye entre otras cosas el ensamblador, el enlazador y varias herramientas para trabajar con binarios elf y demás.

Seguiremos con la compilación del depurador SIN borrar nada de lo anterior, pues sera necesario para su correcta compilación. Éstos son los comandos:

```
$ cd .../compilacion/dbg
$ ./configure --prefix=[lugar donde se instalará, si no se especifica nada en /usr]
$ make
$ make install
```

Con esto ya podríamos crear programas en ensamblador y depurarlos y/o simularlos.

4. Opciones de la línea de comandos

Uso: **sisa-dbg** [options] [binary]

Opciones disponibles:

-f,--file [binary]

Carga en la memoria el programa que se desea depurar/simular

Se puede omitir el indicador -f o --file y especificar el nombre del programa directamente.

-d,--debug [filename]

Guardar información de depuración de la propia aplicación. Importante para avisar de fallos del depurador.

-v,--version

Muestra información sobre la versión y el autor.

-h,--help

Muestra un resumen de uso y las opciones disponibles.

5. Interfaz de usuario

The screenshot displays the Sisa Debugger 0.9 interface. At the top, the title bar reads "Sisa Debugger 0.9" and the status bar on the right says "[Stopped]". The main window is divided into three panes. The top pane, titled "[1] Disassembler", shows assembly instructions: 0156 5580 movhi r2, -128 <0x80>, 0158 7538 out 56, r2 <0x0038>, 015a 5288 movi r1, -120 <0x88>, 015c 5301 movhi r1, 1 <0x01>, 015e* d440 ldb r2, 0(r1) (highlighted), 0160 64fe bz r2, 0x015e, and 0162 ffff halt. The middle pane, titled "[2] Data", shows a memory dump from 0180 to 01d0, with the value at 0188 highlighted as 0000. The bottom pane, titled "[4] Registers", shows register values: R0:0000, R1:0188, R2:0000, R3:0000, R4:0000, R5:0012, R6:0000, R7:0000, PC:015e, S0:0002, S1:015e, S2:000f, S3:0000, S4:0000, S5:003a, S6:0000, S7:0002, and floating-point registers F0 through F7 all set to 0.00000e+00. At the bottom, a status bar contains function key shortcuts: [S-F1 Help], [F2 Emulator], [F3 Debugger], [F4 Run], [F5 Step], [F11 Panel], and [F12 Main].

5.1. Conceptos básicos

Podemos encontrar los siguientes elementos en la pantalla:

1. Barra de título
2. Area de trabajo
3. Barra de menus y entrada
4. Barra de estado

La barra del titulo muestra además del nombre de la aplicación, la versión y en la parte derecha el estado de ejecución del simulador entre corchetes.


El área de trabajo puede albergar una de los tres modos de trabajo posibles:

- Ayuda, muestra un resumen de las funcionalidades y las teclas de función disponibles.
- Emulación, muestra la pantalla, la impresora y los registros de E/S del simulador.
- Depuración, permite seguir paso a paso e inspeccionar diversos aspectos del estado del procesador y la memoria.

La barra de menus y entrada se utiliza indistintamente para mostrar los menus o para las entradas de datos del usuario. Existen dos tipos de menú, el menú global de la aplicación, y los menús contextuales. El primero esta disponible siempre durante el transcurso de la aplicación y contiene opciones globales, mientras que los menús contextuales dependen del contenido del area de trabajo.

La barra de estado muestra información sobre las opciones del menú y las acciones del usuario.

5.2. Expresiones matemáticas



```
bucle + r5 * 2
Enter an expression containing the new value
```

En determinadas situaciones el usuario deberá introducir una expresión matemática en la linea de comandos con el fin de dar o bien una dirección de memoria o bien un dato.

Se admiten los siguientes operandos:

- Números enteros en diferentes formatos:
 - Precedidos de 0x para números hexadecimales. Ejemplo: 0x3fab

- Precedido de un 0 para números octales. Ejemplo: 0472
- Números decimales.
- Registros, computará el valor que en ese momento haya en el registro.
Ejemplo: r2 * 4
- Símbolos. Ejemplo: main

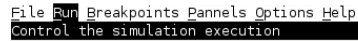
Se admiten las siguientes operaciones, en orden de precedencia:

- - (signo negativo)
- * (multiplicación), / (división), % (módulo)
- + (suma), - (resta)
- <<(desplazamiento aritmético izquierda), >>(deracha)
- & (y lógico), | (o lógico), ^ (xor lógico)

5.3. Teclas globales

- Shift-F1 : Visualizar la ayuda.
- F2 : Alternar el modo de trabajo.
- F4 : Continuar / Detener ejecución del simulador.
- F5 : Ejecutar una instrucción.
- F6 : Ejecutar una instrucción sin entrar en subrutinas.
- F8 : Ejecutar hasta la dirección del cursor de posición (solo en vistas de desensamblador y código fuente)
- F9 : Recargar el programa. No restaura el contenido de la memoria.
- Shift-F9 : Recargar el programa. Restaura el contenido de la memoria.
- F10 : Añadir punto de ruptura. En la línea de entrada introducir la expresión matemática de la dirección.
- F11 : Entrar en el menú contextual asociado a la vista activa.
- F12 : Entrar en el menú global de la aplicación.
- Control-X : Terminar la aplicación.

5.4. Menú global



File Run Breakpoints Pannels Options Help
Control the simulation execution

5.4.1. File

Engloba operaciones relacionadas con el programa de la simulación.

- Load : Cargar un programa en memoria, se introduce la ruta al fichero en la línea de comandos.
- Reload : Vuelve a cargar el último programa en memoria. Reinicializa el simulador.
- Exit : Termina la aplicación.

5.4.2. Run

Engloba operaciones relacionadas con la ejecución del simulador.

- Step : Ejecutar una instrucción.
- Continue : Continuar / Detener ejecución del procesador.
- Stop : Detener ejecución del procesador.
- Restart : Reiniciar el estado del procesador y la ejecución del simulador.

5.4.3. Breakpoints

Engloba operaciones relacionadas con los puntos de interrupción de la ejecución.

- Add : Añadir un punto de interrupción. En la línea de entrada introducir la expresión matemática de la dirección.
- Remove : Eliminar un punto de interrupción. En la línea de entrada introducir la expresión matemática de la dirección.
- ClearAll : Eliminar todos los puntos de ruptura.

5.4.4. Pannels

Permite cambiar la vista asociada a un panel o cerrarlo. En primer lugar aparece un menú con el número de panel sobre el que se desea operar. A continuación aparecerá otro menú cuya primera opción (Close) permite cerrar el panel, el resto de opciones se refiere al nombre de la vista que se puede asociar a dicho panel. Se pueden cerrar todos los paneles a excepción del primero. No a todos los paneles se les puede asociar todas las vistas.

5.4.5. Options

Permite cambiar las opciones globales del programa. Los cambios no son persistentes entre diferentes ejecuciones de la aplicación.

- RunFollow : Alternar el seguimiento global de la ejecución del simulador, seguimiento del pc, de accesos a memoria, etc.
- Trace : Alternar el trazado de ejecución del procesador.

5.4.6. Help

Muestra la ayuda de la aplicación.

5.5. Modos de trabajo

5.5.1. Ayuda

```
Sisa Debugger 0.9 [Stopped]
Ctrl-X Salir de la aplicación

*** ESPACIO DE TRABAJO DEL DEPURADOR ***

TAB Alterna entre los paneles abiertos
[1,2,3,4] Enfocar el panel 1, 2, 3 o 4

- DESENSAMBLADOR:

Teclas de desplazamiento, AvPag, RePag, Inicio, Fin
O Alternar seguimiento del PC
A Posicionar el cursor en la dirección del PC
G Mover el cursor a una determinada dirección
B Alternar punto de ruptura en la dirección del cursor
+ Activar punto de ruptura
- Desactivar punto de ruptura

- CÓDIGO FUENTE:

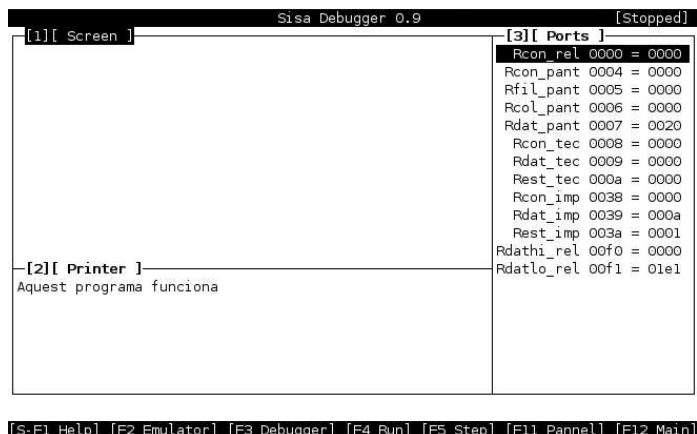
Teclas de desplazamiento, AvPag, RePag, Inicio, Fin

[S-F1 Help] [F2 Emulator] [F3 Debugger] [F4 Run] [F5 Step] [F11 Pannel] [F12 Main]
```

Se puede desplazar mediante las teclas de cursor y desplazamiento de página hacia arriba y abajo con el fin de visualizar todo el contenido de la ayuda.

Para entrar en este modo de trabajo se debe presionar la combinación de teclas Shift-F1.

5.5.2. Emulaci3n

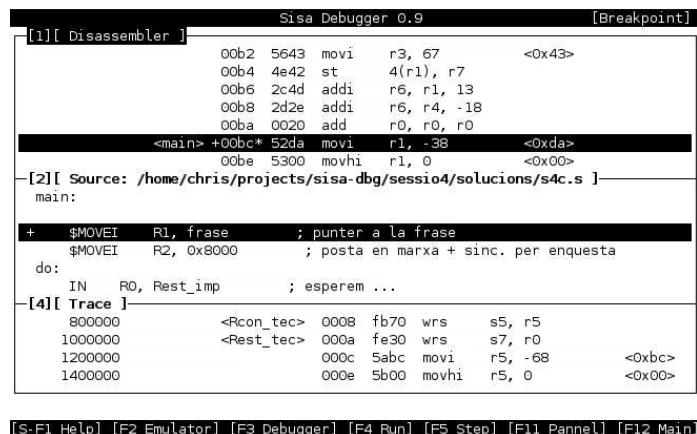


En este modo de trabajo se muestran las interficies externas del simulador, como son la pantalla y la impresora, y los registros de entrada/salida de los dispositivos.

Mientras se encuentra en este modo no hay acceso al men3 contextual y las pulsaciones de teclas num3ricas y alfanum3ricas junto con algunos signos de puntuaci3n y las teclas de retorno y borrar son eviadas al simulador para que las procesen los dispositivos de entrada/salida.

La tecla F2 alterna entre los modos de trabajo de emulaci3n y depuraci3n. Si nos encontr3bamos en la ayuda, saldr3 de 3sta antes de alternar.

5.5.3. Depuraci3n



En este modo el area de trabajo se divide en 4 paneles dispuestos de una manera espec3fica. Cada panel tiene un t3tulo y esta separado de los adyacentes por barras de separaci3n. Adem3s pueden aparecer o desaparecer a voluntad del

usuario. Cada panel puede albergar una vista entre todas las posibles. Una vista se encarga de visualizar y/o interactuar con el usuario.

Hay varias vistas disponibles, desensamblador, registros, memoria, código fuente, inspección de variables y trazado de ejecución. A continuación se describirán más detalladamente.

5.6. Vistas del depurador

5.6.1. Desensamblador (Disasm)

Sisa Debugger 0.9					[Breakpoint]
[1] [Disassembler]					
	00b2	5643	movi	r3, 67	<0x43>
	00b4	4e42	st	4(r1), r7	
	00b6	2c4d	addi	r6, r1, 13	
	00b8	2d2e	addi	r6, r4, -18	
	00ba	0020	add	r0, r0, r0	
<main>	+00bc*	52da	movi	r1, -38	<0xda>
	00be	5300	movhi	r1, 0	<0x00>
	00c0	5400	movi	r2, 0	<0x00>
	00c2	5580	movhi	r2, -128	<0x80>
<do>	00c4	703a	in	r0, 58	<0x003a>
	00c6	60fe	bz	r0, 0x00c4	
	00c8	d640	ldb	r3, 0(r1)	
	00ca	6604	bz	r3, 0x00d4	
	00cc	7739	out	57, r3	<0x0039>
	00ce	7538	out	56, r2	<0x0038>
	00d0	2241	addi	r1, r1, 1	
	00d2	67f8	bnz	r3, 0x00c4	
<enddo>	00d4	703a	in	r0, 58	<0x003a>
	00d6	60fe	bz	r0, 0x00d4	
[S-F1 Help] [F2 Emulator] [F3 Debugger] [F4 Run] [F5 Step] [F11 Pannel] [F12 Main]					

Muestra el desensamblado de la memoria. Hay dos tipos de cursores, el cursor de posición se muestra en modo resaltado y permite moverse por diferentes posiciones de la memoria mediante las teclas de desplazamiento y avance y retroceso. El otro cursor, de ejecución, se muestra mediante un asterisco (*) junto a la columna de dirección de memoria, e indica el punto de ejecución en el que se encuentra el procesador (pc).

Se muestran 4 columnas:

- Símbolo asociado a la dirección de memoria
- Dirección de memoria
- Código hexadecimal de la instrucción
- Desensamblado de la ejecución

Teclas de función específicas de la vista:

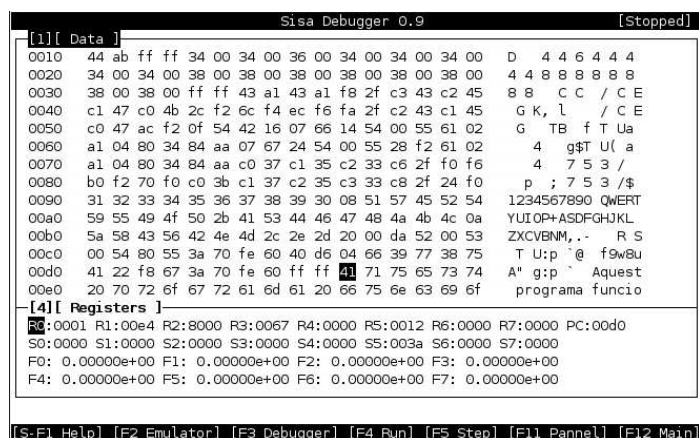
- Teclas de desplazamiento, avance, retroceso e inicio y fin de página para moverse por diferentes posiciones de la memoria.
- O : Alternar el seguimiento del punto de ejecución del procesador (pc).

- A : Posicionar el cursor de posición en la posición del cursor de ejecución.
- G : Mover el cursor a una determinada posición de memoria determinada por una expresión.
- B : Alternar un punto de ruptura en la posición del cursor de posición.
- + : Activar un punto de ruptura en la posición del cursor de posición.
- - : Desactivar un punto de ruptura en la posición del cursor de posición.

Opciones del menú contextual:

- Goto: Situar el cursor de posición en una dirección de memoria. En la línea de entrada se puede introducir una expresión matemática que represente la dirección.
- Breakpoint: Añadir un punto de interrupción. En la línea de entrada se puede introducir una expresión matemática que represente la dirección.
- Following: Alternar la opción de seguimiento del PC. Cuando está activada el cursor de posición seguirá al de ejecución automáticamente.

5.6.2. Memoria (Data)



Muestra un volcado de la memoria principal. Se puede seleccionar mediante un cursor una determinada posición de memoria. La memoria crece de izquierda a derecha y de arriba a abajo.

Teclas de función:

- Teclas de desplazamiento, avance y retroceso de página, inicio y final.

- F : Cambiar el formato de visualización de los datos de forma secuencial, en el siguiente orden: ascii, hexadecimal (byte, word, long), decimal con signo (byte, word), sin signo (byte, word), real (coma fija, científico).
- O : Alternar el seguimiento de los accesos a memoria de la cpu. Si está activo el cursor se situará automáticamente en la última posición de memoria accedida por la cpu.
- A : Mover el cursor a la última posición de memoria accedida por la cpu.
- G : Mover el cursor a una determinada posición de memoria. Se espera una expresión en la línea de entrada.
- M : Modificar el contenido de una posición de memoria. Es sensible al formato de visualización activo.
 - Ascii: se entra una cadena de caracteres ascii.
 - Hexadecimal o decimal: una expresión con números enteros
 - Real: Un número en coma fija o en formato científico.

Opciones del menú contextual:

- Format: Seleccionar el formato de visualización del contenido de la memoria.
- Modify: Modificar el contenido de una posición de memoria. Es sensible al formato de visualización activo. Se comporta igual que la tecla de función M.
- Goto: Mover el cursor a una determinada posición de memoria. Se espera una expresión en la línea de entrada.
- Following: Alternar el seguimiento de los accesos a memoria de la cpu. Si está activo el cursor se situará automáticamente en la última posición de memoria accedida por la cpu.

5.6.3. Registros (Reg)

Muestra los registros del procesador. Cada vez que el estado de los registros del procesador cambia, el valor correspondiente se muestra iluminado en pantalla con el fin de avisar de dicho evento.

Teclas de función:

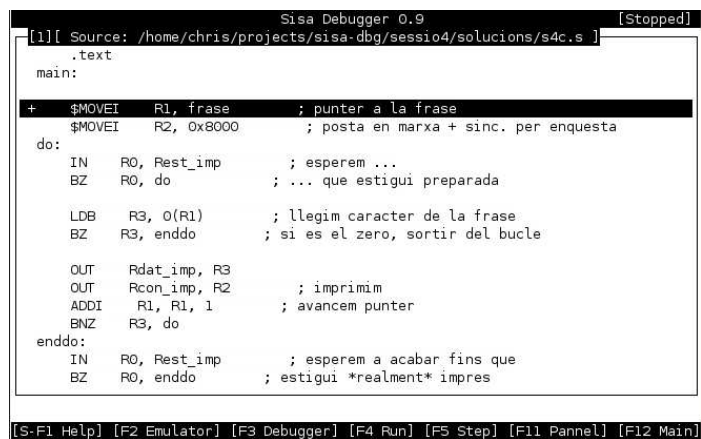
- Teclas de desplazamiento: Arriba y abajo cambia entre los diferentes bancos de registros y izquierda, derecha cambia entre los registros de un determinado banco.

- M : Modificar el valor del registro. Se espera una expresión matemática para el nuevo valor.

Opciones del menú contextual:

- Modify: Modificar el valor del registro seleccionado.

5.6.4. Código fuente (Source)



```

Sisa Debugger 0.9 [Stopped]
[1] Source: /home/chris/projects/sisa-dbg/sessio4/soluciones/s4c.s
.text
main:
+ $MOVEI R1, frase ; punter a la frase
$MOVEI R2, 0x8000 ; posta en marxa + sinc. per enquesta
do:
IN R0, Rest_imp ; esperem ...
BZ R0, do ; ... que estigui preparada

LDB R3, 0(R1) ; llegim caracter de la frase
BZ R3, enddo ; si es el zero, sortir del bucle

OUT Rdat_imp, R3
OUT Rcon_imp, R2 ; imprimim
ADDI R1, R1, 1 ; avancem punter
BNZ R3, do
enddo:
IN R0, Rest_imp ; esperem a acabar fins que
BZ R0, enddo ; estigui *realment* impres
  
```

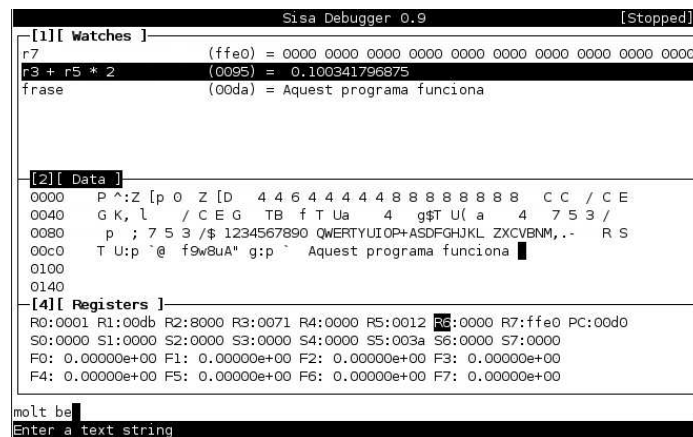
Muestra el código fuente en ensamblador asociado a la posición de memoria en ejecución. Para que esto sea posible se debe haber compilado con información de depuración (-gstabs).

Teclas de función:

- N : Alternar la opción de visualización de números de línea.
- O : Alternar el seguimiento del punto de ejecución del procesador (pc).
- A : Posicionar el cursor de posición en la posición del cursor de ejecución.
- G : Mover el cursor a una determinada posición de memoria determinada por una expresión.
- B : Alternar un punto de ruptura en la posición del cursor de posición.
- + : Activar un punto de ruptura en la posición del cursor de posición.
- - : Desactivar un punto de ruptura en la posición del cursor de posición.

Opciones del menú contextual:

5.6.5. Inspección de variables (Watch)



Permite inspeccionar rangos de memoria a partir de una determinada expresión matemática que represente una dirección de memoria válida. Cada línea representa un punto de inspección. En la primera columna se muestra la expresión que representa la dirección de memoria, en la segunda columna el valor calculado en hexadecimal entre paréntesis y a continuación el contenido de la memoria a partir de dicha dirección en el formato seleccionado.

Teclas de función:

- Teclas de desplazamiento, avance y retroceso de página, inicio y final.
- W : Añadir punto de inspección. Aparecerá un menú para seleccionar el formato de visualización, a continuación deberemos entrar en la línea de comandos, y en el siguiente orden, el número de elementos que se desea mostrar y una expresión matemática que represente la dirección.
- C : Elimina el punto de inspección seleccionado.

5.6.6. Trazado de ejecución (Trace)

Sisa Debugger 0.9						[Stopped]
[1] [Trace]						
1800000	<main>	00bc	52da	movi	r1, -38	<0xda>
2000000		00be	5300	movhi	r1, 0	<0x00>
2200000		00c0	5400	movi	r2, 0	<0x00>
2400000		00c2	5580	movhi	r2, -128	<0x80>
2600000	<do>	00c4	703a	in	r0, 58	<0x003a>
2800000		00c6	60fe	bz	r0, 0x00c4	
3000000		00c8	d640	ldb	r3, 0(r1)	
3200000		00ca	6604	bz	r3, 0x00d4	
3400000		00cc	7739	out	57, r3	<0x0039>
3600000		00ce	7538	out	56, r2	<0x0038>
3800000		00d0	2241	addi	r1, r1, 1	
4000000		00d2	67f8	bnz	r3, 0x00c4	
4200000	<do>	00c4	703a	in	r0, 58	<0x003a>
4400000		00c6	60fe	bz	r0, 0x00c4	
4600000		00c8	d640	ldb	r3, 0(r1)	
4800000		00ca	6604	bz	r3, 0x00d4	
5000000		00cc	7739	out	57, r3	<0x0039>
5200000		00ce	7538	out	56, r2	<0x0038>
5400000		00d0	2241	addi	r1, r1, 1	

[S-F1 Help] [F2 Emulator] [F3 Debugger] [F4 Run] [F5 Step] [F11 Pannel] [F12 Main]

Si la opción global de traza está activa (ver menú global de la aplicación) esta vista mostrará un historial de todas la instrucciones ejecutadas hasta el momento.

Se muestran 5 columnas:

- El instante de tiempo en que se ejecutó, en nanosegundos.
- El símbolo asociado a la posición de memoria de la instrucción que se ha ejecutado.
- La dirección de memoria, en hexadecimal.
- La codificación de la instrucción en hexadecimal.
- La instrucción en código ensamblador.

Teclas de función:

- Teclas de desplazamiento, avance y retroceso de página, inicio y final.
- O : Alternar el seguimiento del punto de ejecución del procesador (pc).

No tiene menú con-textual.

5.7. Vistas del emulador

Ninguna de las siguientes vistas posee un menú con-textual.

5.7.1. Pantalla

Esta será la vista que tenga el foco siempre, en ella se representa el contenido de la pantalla de la plataforma SISAF, tal y como se vería si se tratara de una pantalla real. Todo el espacio que sobre se rellenará para que quede claro que forma parte del contenido y qué no.

En esta misma vista se recogen las pulsaciones de teclado del usuario y son notificadas al simulador para que pueda emular la existencia de un teclado externo. En realidad solo se admiten un subconjunto de todas las teclas que tiene un teclado de PC estándar. Dichas teclas están documentadas en el material de la asignatura EC1.

5.7.2. Impresora

En esta vista se muestran las últimas líneas que se han enviado a la impresora, tal y como aparecería en el papel acorde con las características del hardware emulado.

5.7.3. Puertos de E/S

Aquí podremos ver el estado de los registros de E/S de los controladores de hardware. Se muestra el nombre del puerto, el número de puerto y el valor que tiene en un momento dado.

6. Conclusión y futuras aportaciones

Se ha desarrollado una aplicación que cumple con las expectativas de la asignatura de EC1 en cuanto a la posibilidad de poner en práctica por parte de los alumnos los conocimientos adquiridos sobre la plataforma SISAF. Ya se está utilizando en las sesiones de laboratorio y el resultado está siendo muy positivo.

También ha sido utilizado en un proyecto final de carrera para ponerse al día y practicar con la programación del SISAF con el fin de implementarlo en hardware real.

Aunque no dejan de haber posibles mejoras y frentes de desarrollo.

En primer lugar, una interfaz de usuario gráfica sería más deseable, y haría más sencillo su utilización.

Otro aspecto importante sería adaptarlo a otras plataformas de la familia SISA con el fin de abarcar más aplicaciones y que pudiera ser utilizado en otras asignaturas.