



Simulador de Arquitecturas Q

Susana Rosito

Tatiana Molinari

Índice

1. Arranque del Simulador QSim	1
2. Agregar archivos	2
3. Ensamblar	2
4. Cargar en memoria	2
5. Ciclo de instrucción	3
6. Visualización de puertos	3
7. Repositorios remotos del código de la aplicación	4

1. Arranque del Simulador QSim

Para poder arrancar el simulador contamos con 2 archivos ejecutables. Dependiendo del sistema operativo en el que estemos utilizamos uno u otro.

A Continuación se describirá como se ejecuta el simulador QSim tanto en Windows como en Linux.

- Ejecución QSim en Linux.

En Linux tenemos que fijarnos cuantos bits tiene el sistema operativo (32bits o 64bits). Si es de 32bits, el .jar que tenemos que utilizar para el ejecutable tiene que estar generado en 32bits y a si mismo sucede con el sistema en 64bit. El ejecutable para Linux tiene extensión .sh. El comando para encender el simulador es la siguiente:

```
$ sh qsim.sh
```

Este comando se tiene que realizar parándonos en la carpeta descomprimida del ejecutable. A continuación se abre la pantalla principal de Qsim, la cual se muestra mas adelante.

- Ejecución de QSim en Windows.

En Windows utilizaremos uno de los ejecutables, para ser mas precisos el archivo ejecutable con extensión .bat. A dicho archivo le hacemos doble clic para poder encender el simulador que nos lleva a la pantalla principal QSim.

2. Agregar archivos

Como se observa en la figura 1, la ventana de cargado, la misma cuenta con la opción de agregar archivos .qsim y se encuentran deshabilitadas las otras opciones ya que, sin uno o mas programas de la arquitectura Qi carece de sentido realizar la acción de ensamblar o de cargar en memoria.

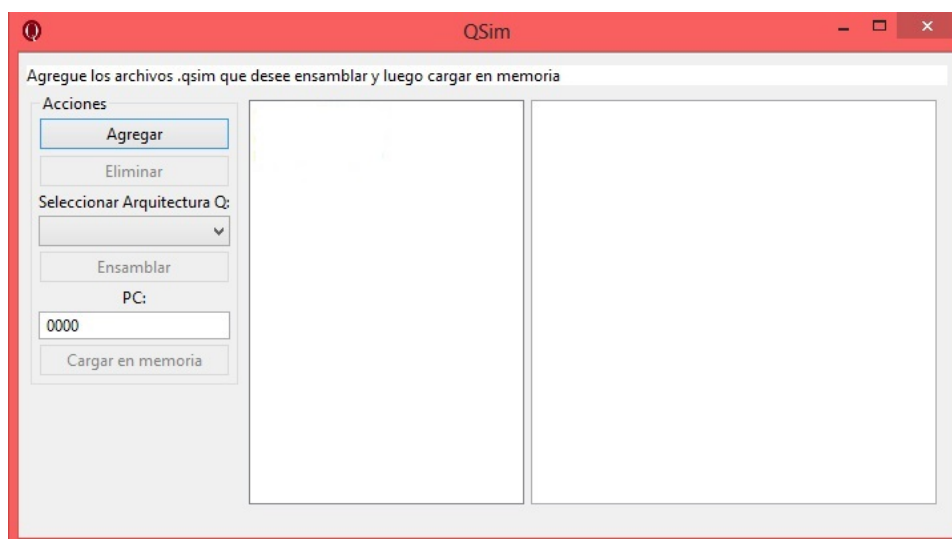


Figura 1: Cargado de archivos

3. Ensamblar

Como se observa en la figura 2 de la ventana Ensamblar una vez que los programas Qi en los archivos -qsim se encuentran agregados se habilitan las opciones de seleccionar la arquitectura Qi que se desee (Q1.. Q6) y la

opción de Ensamblar el programa para que se realice el chequeo de sintaxis y se genere el código máquina que luego será cargado en memoria.

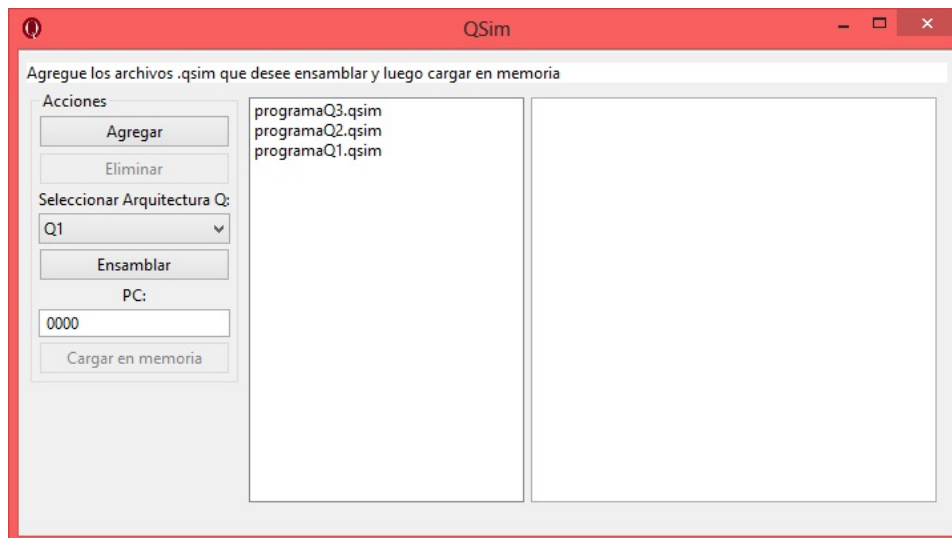


Figura 2: Ensamblar

4. Cargar en memoria

Como se observa en la figura 3 de *cargado en memoria* los programas Qi en los archivos .qsim se encuentran ensamblados en la arquitectura Qi elegida.

Para ese entonces todas las opciones están habilitadas ya que puede desearse agregar otro archivo .qsim o quitar alguno (no todos) y volver a ensamblar, o bien, elegir el Contador de programa a partir del cual se quiere cargar en memoria el programa Qi (por defecto '0000') y hacer clic en el botón 'Cargar en memoria'.

5. Ciclo de instrucción

Como se observa en la figura 4 de la ventana de ejecución una vez que el programa es exitosamente cargado en memoria se abre esta nueva ventana que contiene los registro especiales Contador de programa, Registro de instrucción, Puntero de pila, los flags, los registro de uso general (R0...R7), la memoria, una consola de información al usuario y los botones habilitados para realizar el fetch, ver los puertos y editar los valores de los registros especiales, los flags y pc.

Para poder realizar el ciclo de ejecución de una instrucción, se debe seguir el orden de las etapas de dicho ciclo al apretar los botones que están

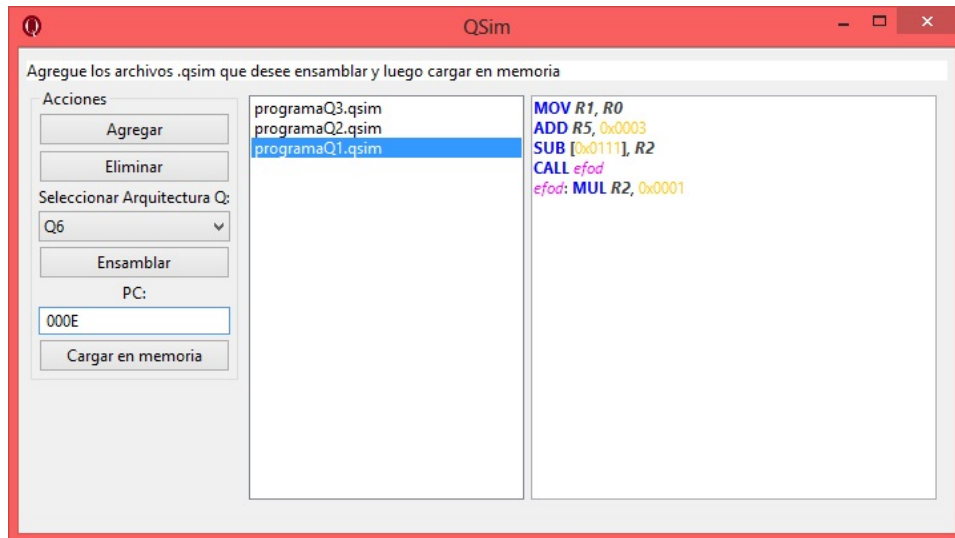


Figura 3: Cargado en memoria

debajo en el panel llamado 'Ciclo de ejecución'. De todas maneras como se ve en las figuras 5, 6 y 7 sólo luego de realizarse el *Fetch*¹ a pedido del usuario puede utilizarse el *decode*², luego el *execute*³ y el ciclo vuelve a repetirse.

6. Visualización de puertos

Si se desea ver el valor que tienen los puertos, basta con hacer clic en el botón "Ver puertos", y se abrirá una nueva ventana tal y como se ve en el *screenshot* de la ventana de puertos (figura 8) donde se podrá observar el número de puerto y su valor actual y también al igual que en la ventana principal seleccionar su edición.

7. Repositorios remotos del código de la aplicación

Código del modelo: <https://github.com/molinarirosito/QSim>

Código de la interfaz: https://github.com/molinarirosito/QSim_UI

¹búsqueda de la instrucción

²decodificación de la instrucción

³Ejecución de la instrucción

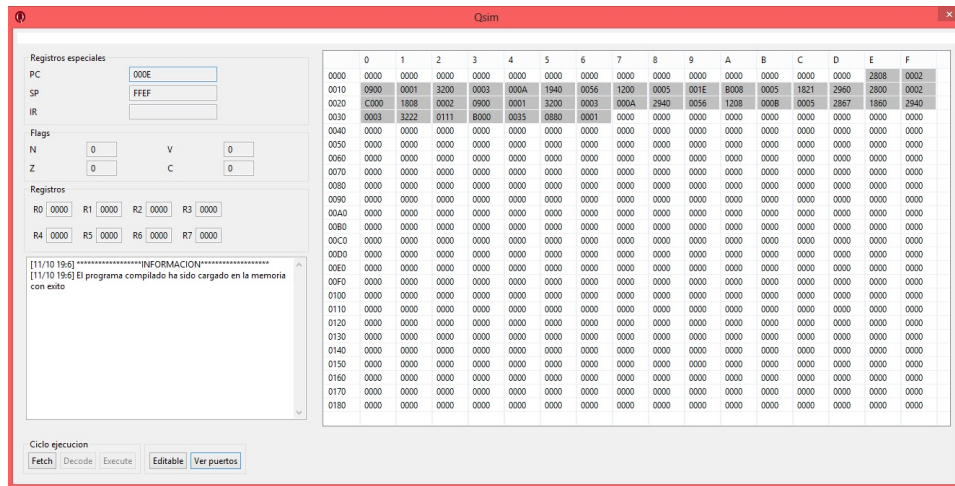


Figura 4: Ventana de ejecución

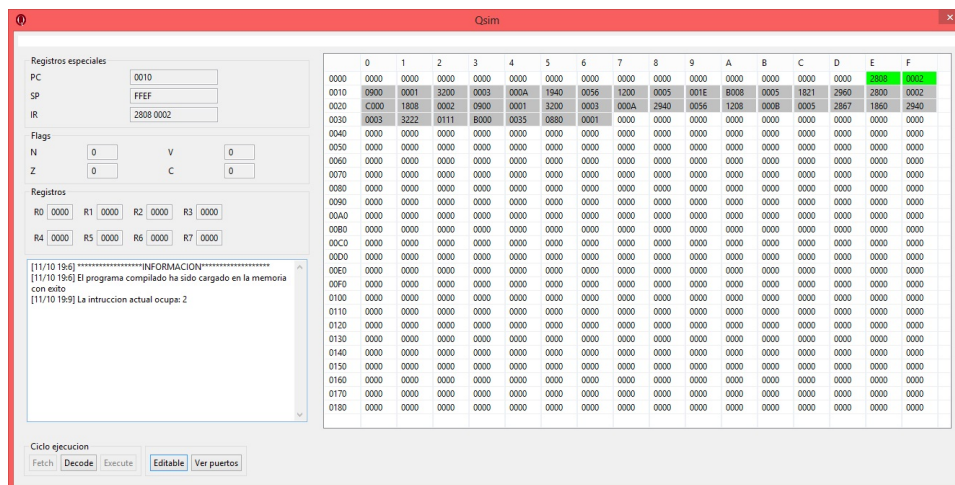


Figura 5: Ventana de ejecución luego del *fetch*

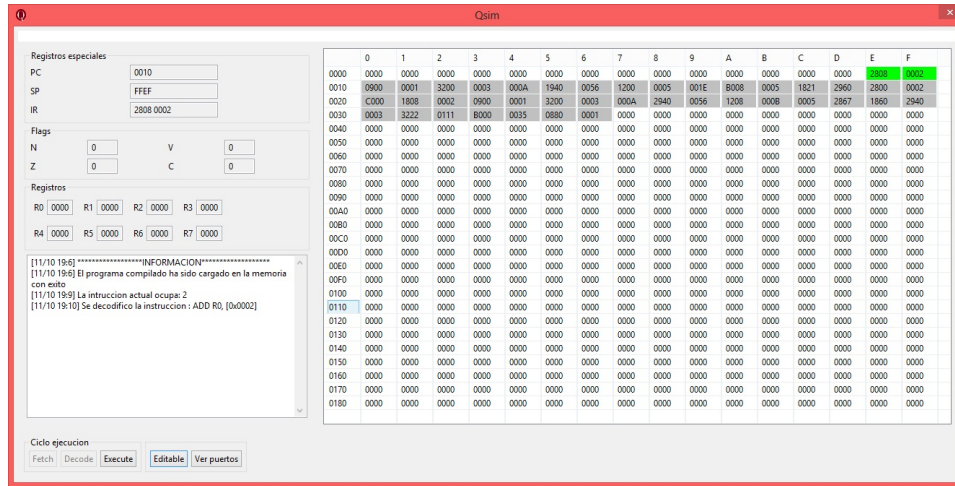


Figura 6: Ventana de ejecución luego del *decode*

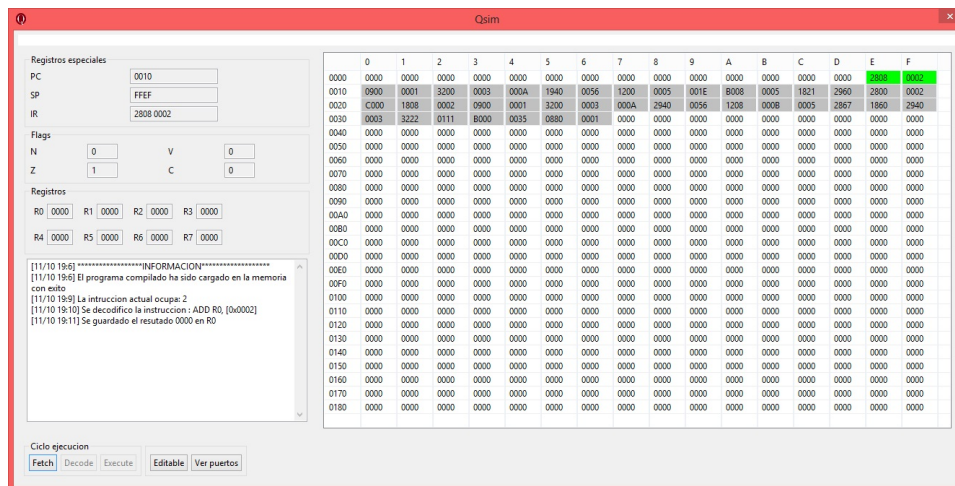


Figura 7: Ventana de ejecución luego del *execute*



Figura 8: Ventana de puertos