Manual GTKWave Analyzer
V3.3.10 (w) 1999-2010 BSI

Elaborado por:
David Ricardo Martínez Hernández
Juan Sebastián Roncancio Arévalo

Electrónica Digital II

Universidad Nacional de Colombia
Facultad de Ingeniería
Bogotá
2012

Índice

Índice de figuras

1. Introducción

GTkwave es una herramienta de análisis utilizada para realizar la comprobación en **Verilog** o **VHDL** de modelos de simulación. Con la excepción de visualización interactiva **VCD**, no se destina a ejecutar de forma interactiva con la simulación, sino que se basa en el uso de ficheros de volcado.
Algunos de los formatos que soporta GTKWave son:

- **VCD** "Cambiar valor de descarga", este es un formato de archivo estándar de la industria que genera la mayoría de los simuladores de Verilog y se especifica en el estándar **IEEE-1364**. es un archivo ASCII que contiene información de encabezado, la definición de variables y los cambios de valor para todas las variables especificadas en las convocatorias de las tareas.
- **LXT** "Entrelazado de seguimiento extensible", este es un formato optimizado utilizando punteros intercalados y cambios de valor. El procesamiento de archivos es más rápido que **VCD**. Fue creado específicamente para su uso con GTKWave, sin embargo algunos otros simuladores lo usan.
- **LXT2** "Entrelazado Versión seguimiento extensible 2", esta es una variante basada en bloques de **LXT** que permite mayores velocidades de compresión y el acceso que se pueden lograr con **LXT**. Permite acceso aleatorio a nivel de bloque y también permite opcionalmente la carga parcial de bloques para el funcionamiento más rápido.
- **VZT** "Trace Verilog con cremallera", este es el resultado de **LXT2**, sin embargo es diferente para la compresión ya que permite tamaños de archivo mucho más pequeños que la mayoría de los formatos de fichero de volcado.

1.1. ¿Qué es GTKWave?

GTKWave como una colección de binarios se compone de dos herramientas las cuales se entrelazan: la aplicación Visor de **GTKWave** y **rtlbrowse**. Además, una colección de aplicaciones de ayuda se utilizan para facilitar tareas tales como la conversión de archivos y la búsqueda de datos de simulación. Están pensadas para funcionar juntos en un sistema coherente, aunque su diseño modular permite que cada uno funcione independientemente de los otros si es necesario.

GTKWave es un analizador de forma de onda y es la principal herramienta utilizada para la visualización. Proporciona un método para la visualización de resultados de la simulación de los datos analógicos y digitales, permite varias operaciones de búsqueda y manipulaciones temporales, puede guardar los resultados parciales (es decir, "las señales de interés") extraído de un volcado completo de la simulación, y, finalmente, puede generar **PostScript** y **FrameMaker** de salida de papel.

rtlbrowse se utiliza para ver y navegar a través de código fuente **RTL** que ha sido analizada y procesada en un archivo.

Las aplicaciones de ayuda realizan diversas tareas especializadas, tales como la conversión de archivos, análisis de **RTL**, y otras operaciones de manipulación de datos.

2. Menú de funciones GTKWave

En la ventana principal del GTKWave se encontraran pestañas con un nombre determinado, si se expande cada una de las diferentes pestañas se pueden encontrar los siguientes enlaces:

- **File**: contiene varios ítems relacionados con el acceso a archivos, impresión, guardar archivos y salida.
- **Edit**: contiene ítems relacionados con la manipulación de las diferentes formas en las que se pueden presentar las señales en la pantalla o la disposición de dichas señales.
- **Search**: contiene ítems relacionados con la búsqueda de señales y valores.
- **Time**: es un conjunto de herramientas que permite tener control sobre el flujo del tiempo y para seleccionar que periodo te muestras son las que se requieren visualizar.
- **Markers**: es un conjunto de herramientas usadas para el control del desplazamiento en la ventana que contienen las señales.

- **View**: es un submenú que tiene control sobre varios atributos gráficos y presentación del status de los ítems.
- **Help**: este menú contiene ítems para la información de la versión del programa.[1]

3. Interfaz de usuario GTKWave

El GTKWave presenta una interfaz la primera vez que se ejecuta como la figura 1:

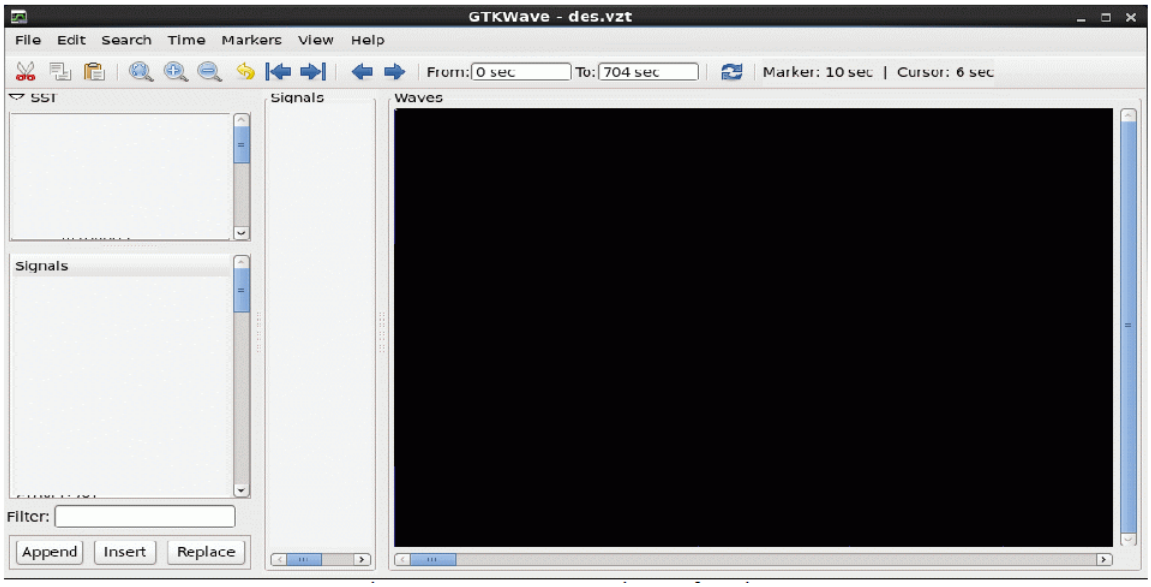


Figura 1: Primera vista gráfica de GTKWave

A continuación se cargan las señales que se necesitan (Recuadros de la izquierda), se seleccionan y se arrastran hasta el panel vacío. Dado que todas las señales no están activadas es necesario hacerlo, para ello es necesario editar el archivo $system.v$, situado en *SIE/SoftCore/lm32/logic/sack*.

Listing 1: Codigo inicial

```
7  module system
8  #(
9  //      parameter    bootram_file      = "../firmware/cain_loader/image.ram",
10 //      parameter    bootram_file      = "../firmware/boot0-serial/image.ram",
11        parameter    clk_freq          = 50000000,
12        parameter    uart_baud_rate    = 57600
```

El codigo 1 se debe descomentar en la linea 9 o 10, depende en que carpeta se valla a trabajar para realizar la compilación correcta, debe quedar como el cogido 4

Listing 2: Código para trabajar en boot0-serial

```
7  module system
8  #(
9  //      parameter    bootram_file      = "../firmware/cain_loader/image.ram",
10       parameter    bootram_file      = "../firmware/boot0-serial/image.ram",
11       parameter    clk_freq          = 50000000,
12       parameter    uart_baud_rate    = 57600
```

Si se ha hecho este procedimiento bien al compilar en al carpeta *sack/*, con el comando *make view* saldrán todas las señales que tiene el procesador para eso se debe fijar en la parte izquierda de la pantalla, en estos dos recuadros se encuentran dichas señales, escogen las que necesitan.
Al realizar el paso mencionado anteriormente y al seleccionar las señales el simulador se observara como en la figura 2.

[1]Para encontrar una completa explicación acerca de la barra de menú, se pueden dirigir a [1], página $33 - 46$

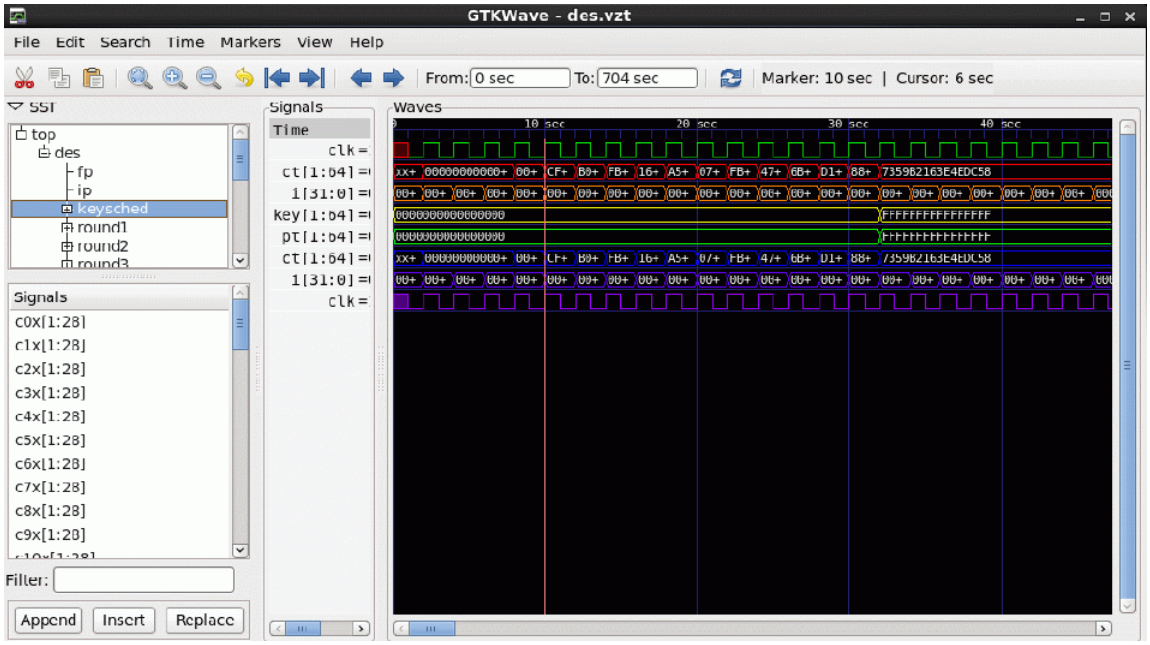
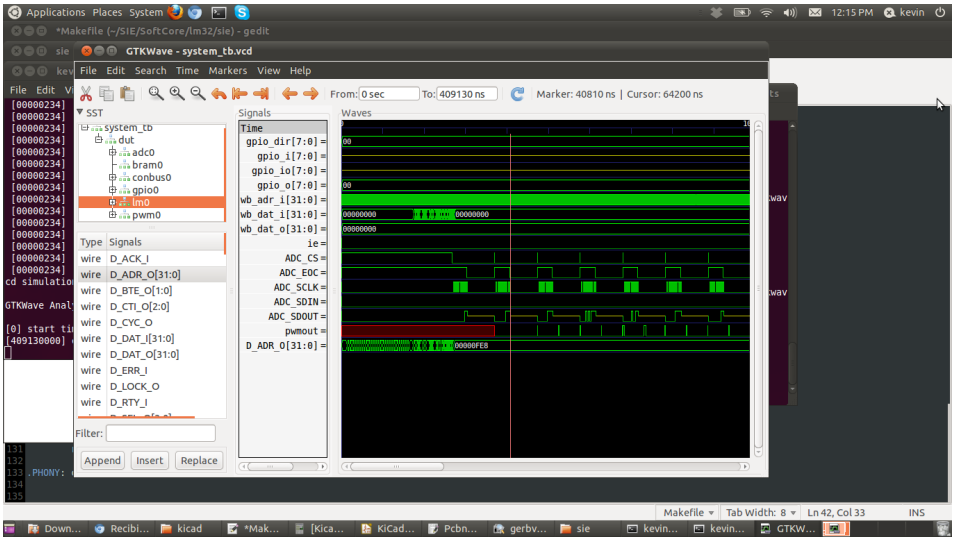


Figura 2: Señales cargadas al GTKWave

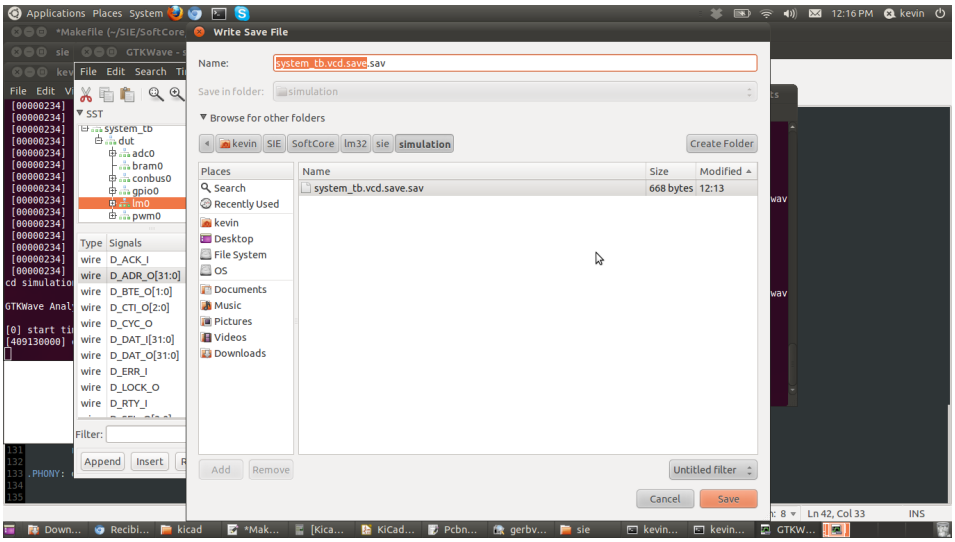
3.1. Cargar señales seleccionadas de la simulación anterior

Como se han podido dar cuenta al simular con GTKWave siempre se tienen que cargar las señales que se van a visualizar, si se esta trabajando en un proyecto o haciendo simulaciones sobre un programa en especifico es muy complicado y realmente molesto tener que cargar cada vez que se quiera simular todas las señales que se quieren visualizar. Por tal motivo a continuación se presenta una forma más fácil de cargar estas señales automáticamente siempre y cuando sean las mismas para cada simulación.

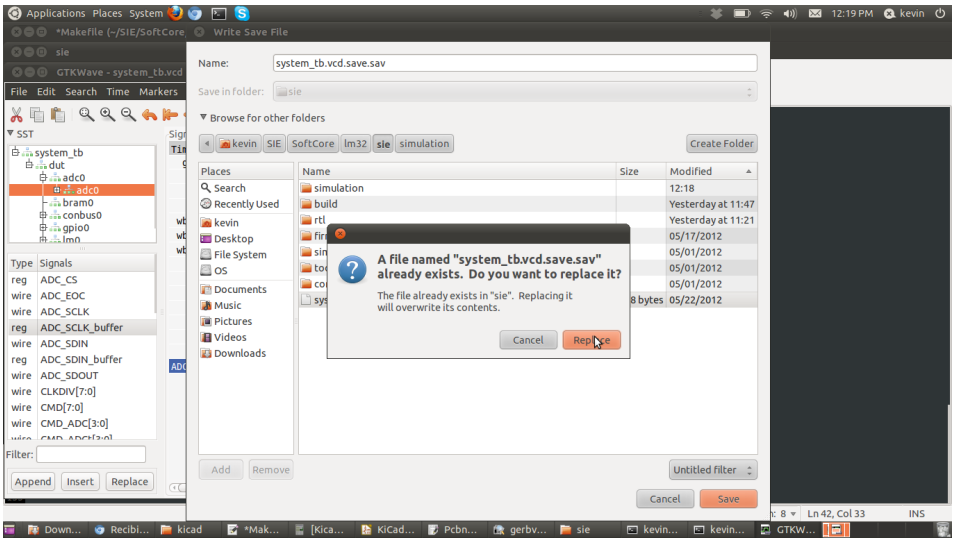
1. Se cargan todas las señales que se quieren ver (Figura 3(a)).
2. File – Write Save File As (Figura 3(b)).
3. Se dirige a la carpeta *SIE/SoftCore/lm32/sie* busca el archivo $system\_tb.vcd.save.sav$, da click en salvar y en la alerta que aparece le dan replace (Figura 3(c)).
4. Ahora nuevamente a la consola y se ejecuta el comando *make view* y cuando se habrá GTKWave aparecerá la simulación con todas las señales que se han estado usando.



(a) Cargar las señales



(b) Buscar la carpeta *sack/*



(c) Sobre escribir el archivo de la carpeta

Figura 3: Pasos a seguir para cargar las señales

4. Comó funciona GTKWave

1. GTKWave espera un archivo. Vcd que se genera a partir de una ejecución de simulación como entrada. Aquí, vamos a suponer que la entrada se fibonacci.vcd que se genera mediante la compilación y simulación "fibonacci.v".

2. Invocar GTKWave: *bash-2.05a $ GTKWave filename.vcd* un seguimiento de la muestra es la siguiente. La ayuda en línea está disponible para todas las funciones de menú en GTKWave.
   Con el fin de acceder a la ayuda en línea, seleccione Help-Wave Ayuda en el menú y luego seleccionar cualquier opción de menú para ver la descripción de la opción de menú.

3. Seleccione la opción Search → Search Tree de la señal. Esto abrirá una nueva ventana. Se ofrece como un medio fácil de añadir las huellas de la pantalla en forma de árbol de un texto basado en la moda. El objetivo principal es añadir las señales a la pantalla para la visualización.
   Árbol de la señal de búsqueda proporciona un medio fácil de añadir las huellas de la pantalla. Varias funciones se proporcionan en el árbol de búsqueda de señales que permiten la búsqueda de una jerarquía de árbol. Ahora selecciona las señales desde el árbol de búsqueda de la señal y haga clic en el botón Añadir. Las señales correspondientes se añaden a la pantalla. Del mismo modo, se puede insertar o cambiar las señales de la pantalla usando la inserción o reemplazar la ficha.

4. Ahora estamos listos para ver y analizar las formas de onda de nuestro diseño. La ventana de la pantalla se divide en partes, a saber las señales de la ventana y ventana de ondas. Podemos observar los valores de las señales en cualquier instante particular del tiempo haciendo clic con el ratón en ese punto en la ventana de la señal.
   Lo que sigue a continuación es la lista y descripción de los elementos de menú de uso común

   a) Para imprimir el resultado en un archivo use la opción File → Print to File, que le preguntará por el nombre de PostScript encapsulado para generar las actuales pantallas de la ventana principal de su contenido.

   b) Para establecer el formato de visualización de datos utiliza la opción Edit → Data format. Se mostrará un conjunto de opciones como decimal, binario, hexadecimal, etc opciones pasará por todos los rastros de relieve y garantizar que el vector con esta clasificación se muestran con valores hexadecimales. Del mismo modo podemos elegir otras opciones.

   c) Podemos buscar formas de onda usando la opción Search → pattern Search. Patrón de búsqueda sólo funciona cuando al menos uno de seguimiento se puso de relieve (haciendo clic en la señal). La solicitud se parece que se enumeran todos los restos seleccionados y permite diferentes criterios que se especifican para cada trazo. La búsqueda puede seguir adelante hacia atrás a partir de la marca primaria no identificada.

4.1. ¿Como encontrar el dato que se requiere en la simulación?

Dado que GTKWave muestra las señales cuadradas no se puede interpretar bien el dato, además aparece en formato Hexadecimal.
Para encontrar los datos es necesario seguir los siguientes pasos:

1. Al tener las señales cargadas en GTkwave (Figuras 2 o 3(a)) en el segundo cuadro *Signals-Time*

2. Se seleccionan las señales que se desean cambiar de formato, como se encuentra en Hexadecimal se quiere cambiar decimal, entonces se ingresa a Edit→Data Format→Decimal, o también se oprime ALT+D.

3. Luego se ingresa a Search→Pattern Search 1, aparecera la figura 4

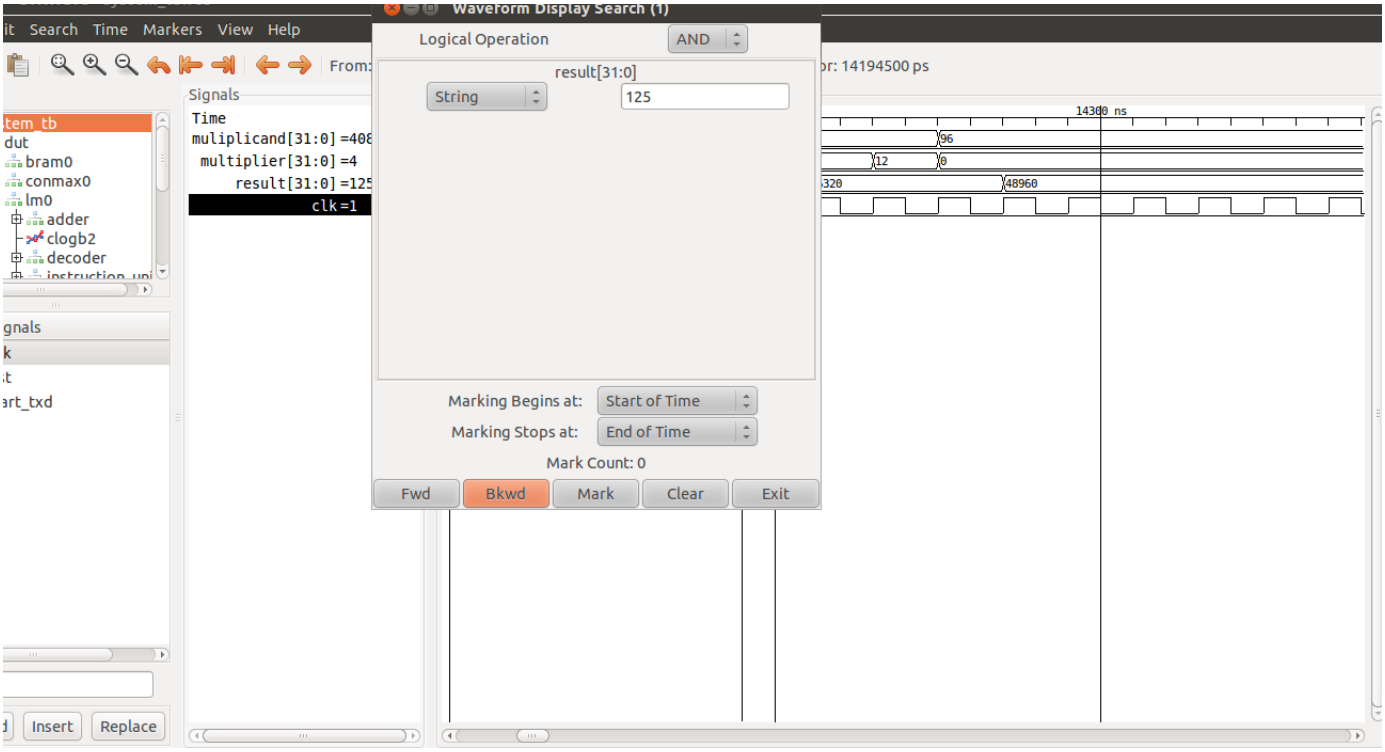


Figura 4: Opción de búsqueda GTKWave

y se dejara como esta en dicha figura.

4. Luego se escribe el dato que se desea buscar y se le da *Bkwd*, así encontrara el dato que desea buscar.

4.2. ¿Como se puede guardar la simulación en un archivo de imagen o PDF?

Primero se debe obtener las señales como las Figuras 2 o 3(a), se ingrese en File→Print to file, y lo guarda en el formato que lo requiera, u oprimiendo CTRL+P.

5. Ejemplos

5.1. Ejemplo 1

Este ejemplo tiene el siguiente código en C

Listing 3: Ejemplo 1 - Multiplicación

```
#include <stdio.h>
int main(){
        volatile unsigned int multiplicador, multiplicando,producto;
        multiplicador=25;
        multiplicando=5;
        producto=0;*
        producto = multiplicador * multiplicando;
        return 0;
}
```

Al realizar el procedimiento anterior se obtuvo como resultado

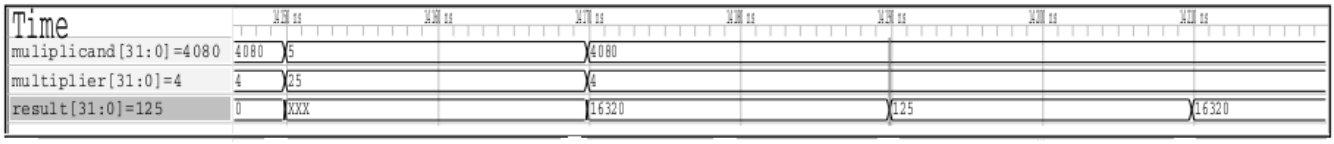


Figura 5: Ejemplo 1

5.2. Ejemplo 2

Este ejemplo tiene el siguiente código en C

Listing 4: Ejemplo 2 - Sumas sucesivas

```
# include"stdio.h"
int main(){
        volatile unsigned int multiplicador,multiplicando,producto,contador;
        multiplicador=7;
        multiplicando=25;
        producto=0;
        contador=0;
        for (contador=0; contador<multiplicador; contador++){
                producto=producto+multiplicando;
        };
        return 0;
}
```

Al realizar el procedimiento anterior se obtuvo como resultado

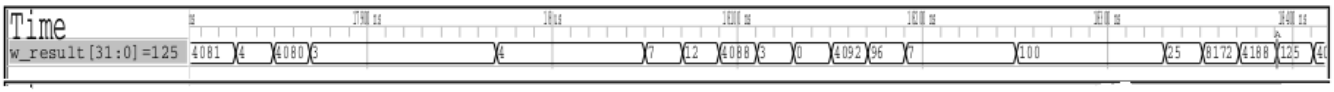


Figura 6: Ejemplo 2

Referencias

[1] http://gtkwave.sourceforge.net/gtkwave.pdf