

DISEÑO DE SISTEMAS DIGITALES

PROYECTO DE LABORATORIO

TEMA DE LA ACTIVIDAD: BEHAVIORAL SIGNAL PROCESSING WITH MACHINE LEARNING BASED ON FPGA

OBJETIVOS DE APRENDIZAJE:

- Elaborar un sistema multi-núcleo utilizando dos microprocesadores Nios II, para la verificación de dos procesos simultáneos utilizando el acceso a memoria para compartición de variables y el acceso a puertos periféricos.
- Predecir el número de pasos de una persona a partir de una red neuronal previamente entrenada y proyectar los resultados en un monitor con puerto VGA.
- Diseñar programación en lenguaje C que ayude a la estimación de número de pasos.

DURACIÓN: 60 minutos

MATERIALES Y HERRAMIENTAS:

- Software Quartus 17.0 versión Estándar.
- Software Eclipse Kepler.
- Tarjeta de desarrollo DE10-Standard.

INTRODUCCIÓN

El proyecto se basa en un sistema embebido capaz de estimar el número de pasos que realizará el usuario en un día determinado, a partir de una red neuronal entrenada con los datos propios del usuario; para lo cual se ha implementado la tarjeta DE10 Standard con FPGA SoC.

Para la realización del proyecto, se utilizará el programa Quartus junto con sus herramientas Qsys y Eclipse que permitirá la creación tanto del software como del hardware respectivamente realizando la interconexión de los bloques que se detallan a continuación: Procesador Nios II, System ID, VGA controller, On chip Memory, Internal Timer, entre otros.


La propuesta cuenta con un sistema multinúcleos, el primer CPU está encargado del almacenamiento de la red neuronal y realiza la predicción del número de pasos a dar; siempre y cuando se ingrese por teclado los siguientes datos: día a predecir, mes a predecir, día de la semana, hora a predecir, temperatura promedio y pasos del día

anterior. El segundo CPU realizará la proyección de los datos a partir de un puerto VGA y un monitor.

Para el entrenamiento de la Red Neuronal utilizada en este trabajo, se realizó el conteo de pasos de un estudiante sano a través de una aplicación móvil y el podómetro incorporado en el Smartphone, dicha aplicación nos muestra el total de pasos en un día entero. De esta manera se logró obtener la cantidad de pasos dados en aproximadamente 2 meses (noviembre y diciembre).

DESCRIPCION DE LA PRACTICA: Inicialmente se debe añadir el archivo de Qsys al proyecto y compilarlo, en el que se encuentra la arquitectura de un sistema multi-núcleo. Se abre la plataforma de Eclipse y se añade la programación en lenguaje C, que contiene la red neuronal y predicción de número de pasos.

PROCEDIMIENTO:

1. Proceda a crear una carpeta en el escritorio con el nombre PROYECTO (escriba sin espacios). Luego copie y pegue todos los archivos que se encuentran en la carpeta compartida, en la carpeta que fue creada en el escritorio.
2. Ejecute el programa Quartus Prime 17.0 Standard Edition que se encuentra en el escritorio representado por el ícono .
3. Proceda a crear un nuevo Proyecto seleccionando la opción **File→New Project Wizard** como se observa en la ilustración 1. Aparecerá la ventana de Introducción, luego da clic en **Next**.

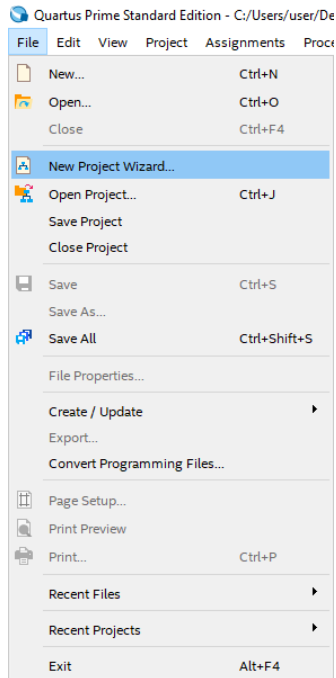


Ilustración 1: Creación de un nuevo proyecto

4. En la primera línea escoja la ruta donde va a crear su proyecto, en este caso, la ruta será la carpeta PROYECTO que fue creada en el procedimiento 1 y que se encuentra en el escritorio. En la siguiente línea escoge como nombre del proyecto PROYECTO, tal como se visualiza en la ilustración 2. Luego da clic en Next.

Ilustración 2: Nombre del proyecto

5. Se mostrará la ventana para seleccionar el tipo de proyecto. Se escogerá la opción de Proyecto vacío (Empty project), tal como se visualiza en la ilustración 3. Luego da clic en **Next**.

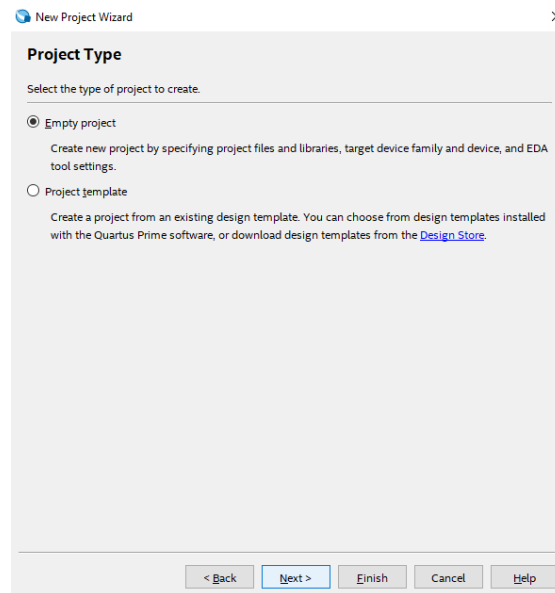


Ilustración 3: Tipo de proyecto

6. Se mostrará la ventana para adhesión de archivos. En esa ventana seleccione la opción **Add All**, para que se agreguen todos los archivos que fueron copiados a la carpeta del proyecto y que se realizó en el procedimiento 1. En la ilustración 4 se observan los archivos añadidos. Por último, haga clic en **Next**.

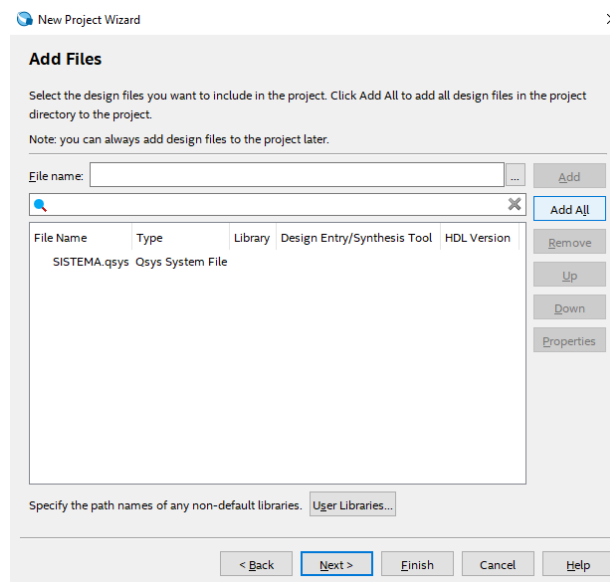


Ilustración 4: Agregar archivos al proyecto

7. En la siguiente ventana aparecerán todas las familias de chips FPGA. Proceda a escoger la familia **Cyclone V (E/GX/GT/SX/SE/ST)** y seleccione el nombre del

chip FPGA SoC **5CSXFC6D6F31C6**, tal como se muestra en la ilustración 5. Luego haga clic en **Next**.

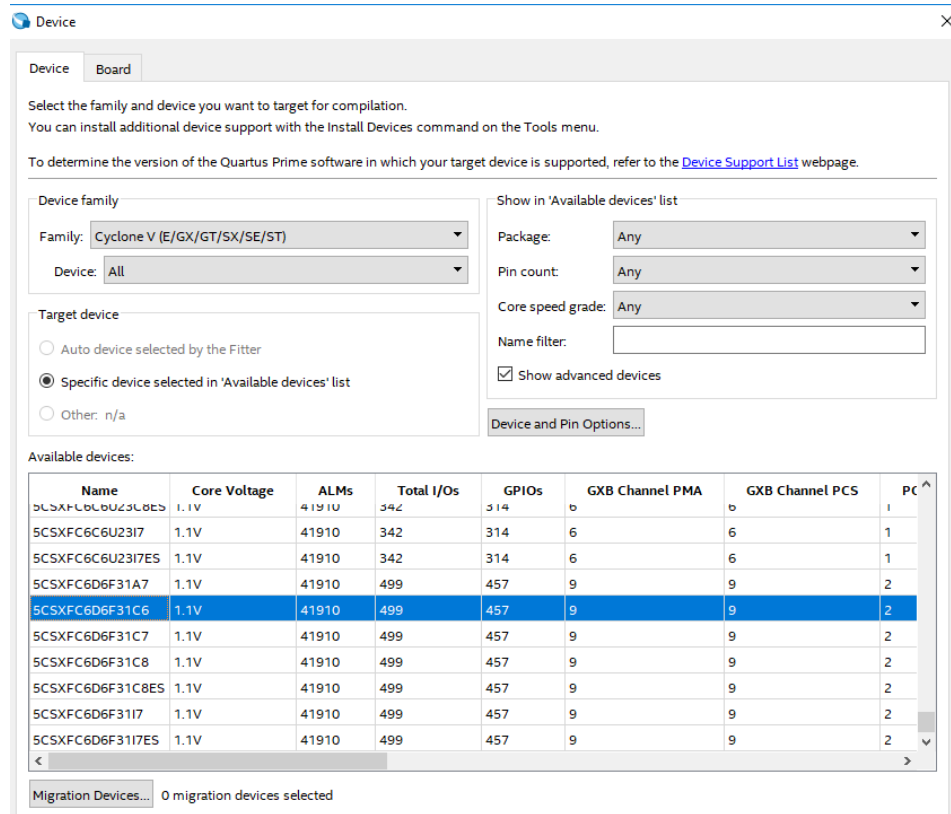


Ilustración 5: Familia y modelo del chip FPGA SoC

8. Debido a que no se realizarán simulaciones en la presente actividad, no se seleccionará alguna herramienta de simulación como se muestra en la ilustración 6. Haga clic en **Next**.

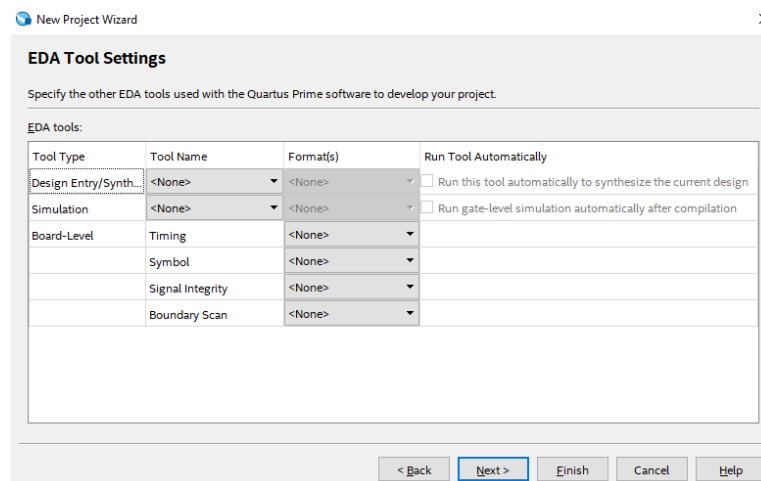


Ilustración 6: Herramientas para simulación.

9. Al finalizar la creación del proyecto, aparecerá la ventana **Summary** para resumir todas las especificaciones que tendrá nuestro proyecto. Para finalizar haga clic en **Finish**.

10. Haga doble clic al archivo **SISTEMA.qsys** dentro del programa **Quartus Prime**. Verifique que se encuentra las siguientes conexiones entre cada componente, como se muestra en la ilustración 7.

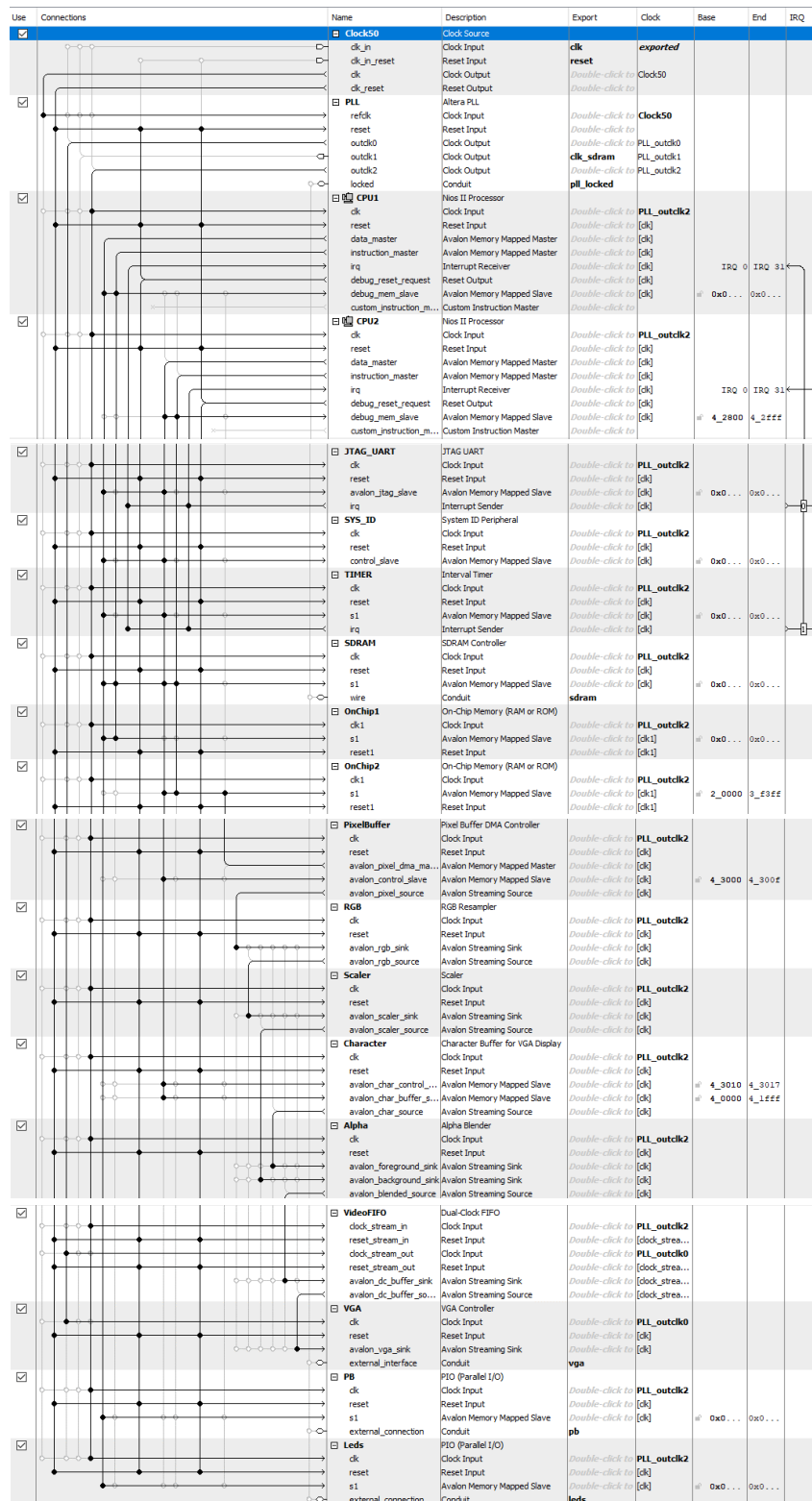


Ilustración 7: Diseño del Qsys realizado

11. Haga clic en **Generate HDL** que se encuentra en la parte inferior derecha, tal como se muestra en la ilustración 8.

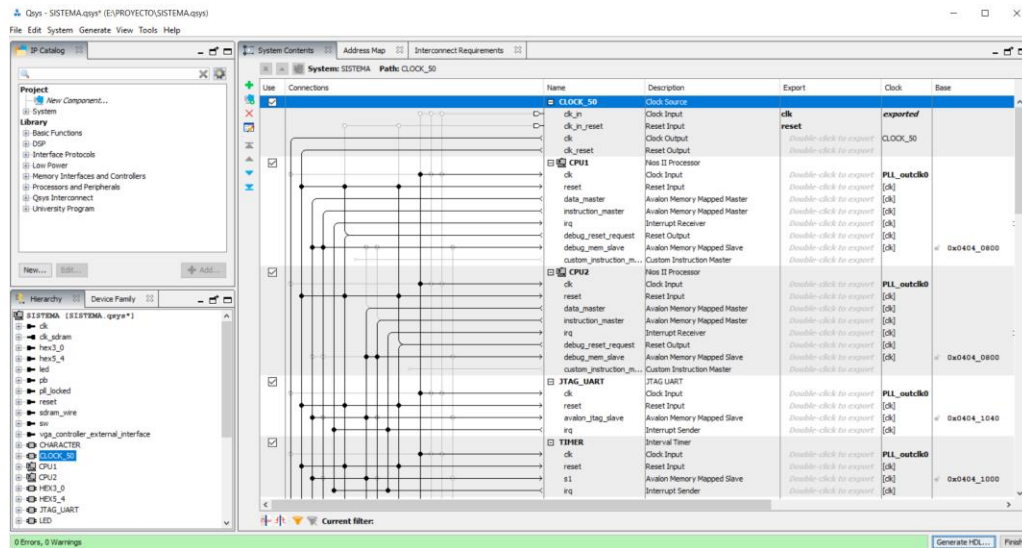


Ilustración 8: Herramienta de Qsys.

12. En la ventana **Generation**, seleccione la creación de archivos VHDL en la sección **Synthesis**, tal como se muestra en la ilustración 9. Por último, haga clic en **Generate**, y cuando finalice haga clic en **Finish**.

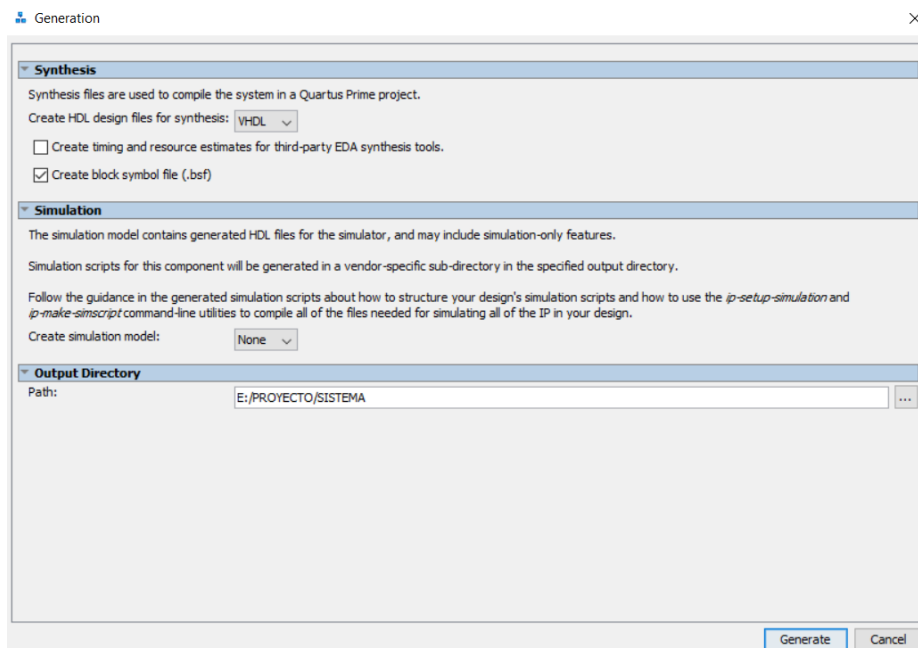



Ilustración 9: Ventana Generation del Qsys

13. Proceda a compilar el archivo **SISTEMA.qsys** estableciéndolo previamente como más alta jerarquía (dando clic derecho en el mismo y seleccionando la opción **Set as Top-Level Entity**), como se muestra en la ilustración 10. Luego seleccione el ícono .

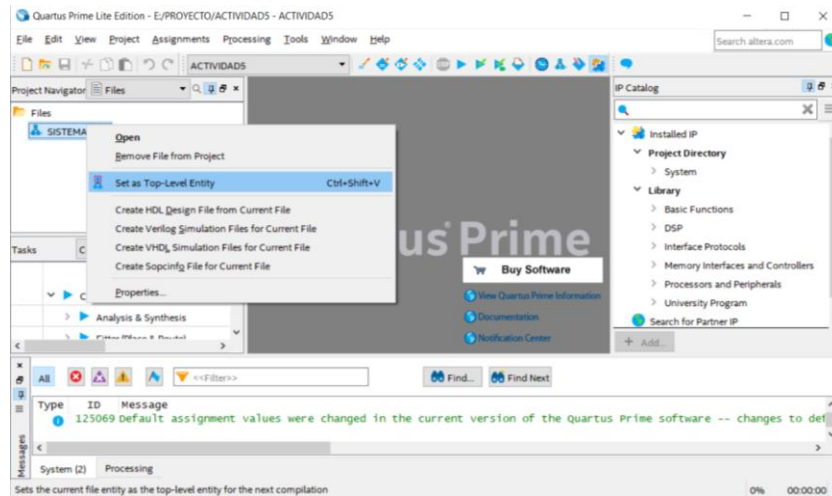




Ilustración 10: Compilación del proyecto

14. A continuación, se procede a la asignación de pines. En la ventana de Assignments elegir la opción de Pin Planner, como se muestra en la ilustración 11.

Ilustración 11: Asignación de pines

15. Nuevamente, proceda a compilar el archivo **SISTEMA.qsys** estableciéndolo previamente como más alta jerarquía (dando clic derecho en el mismo y seleccionando la opción **Set as Top-Level Entity** y seleccione el ícono .
16. Proceda a conectar la tarjeta de desarrollo **DE10-Standard** tanto a la PC (cable **USB-Blaster**) como a la fuente de alimentación.
17. Una vez que se haya completado la compilación, diríjase al ícono  que se encuentra en la barra de tareas del programa **Quartus Prime**, y servirá para programar la tarjeta de desarrollo **DE10-Standard**.
18. En la ventana **Programmer** se visualizará un archivo, haga clic derecho en el mismo y elimínelo escogiendo la opción **Delete**, tal como se visualiza en la ilustración 12.

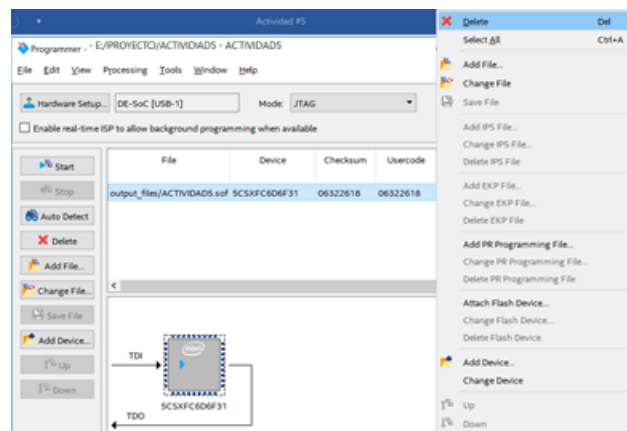
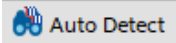


Ilustración 12: Ventana Programmer

19. Una vez eliminado el archivo y que no se encuentre ningún otro archivo en la ventana Programmer, proceda a escoger la opción **Auto Detect** representada por la siguiente botonera . Seleccione la opción **5CSXFC6D6** tal como se muestra en la ilustración 13. Haga clic en OK.

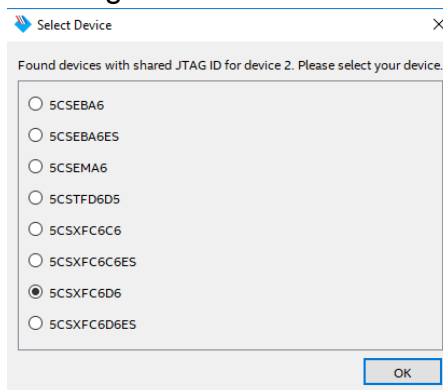


Ilustración 13: Ventana Auto Detect

20. Aparecerán dos archivos: **SOCVHPS** y **5CSXFC6D6**. Haga clic derecho en el segundo y escoja la opción **Change File** tal como se muestra en la ilustración 14, y reemplácelo por el archivo **ACTIVIDAD5.sof** que se encuentra en la carpeta **Output_Files** tal como se muestra en la ilustración 15.

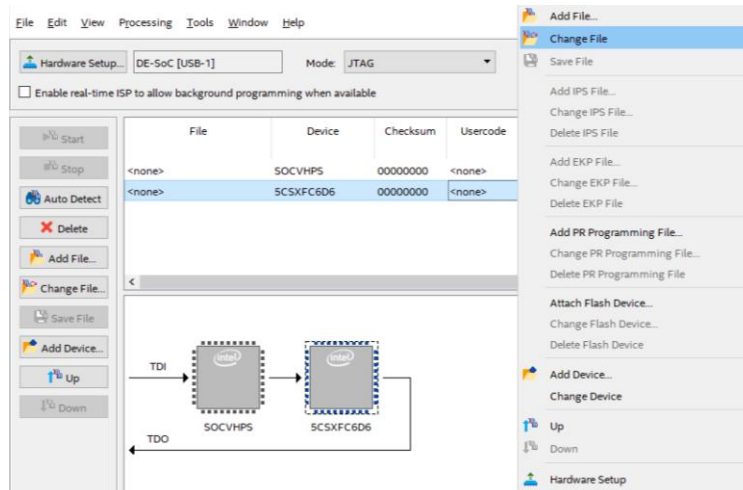


Ilustración 14: Cambio de archivo en la ventana Programmer.

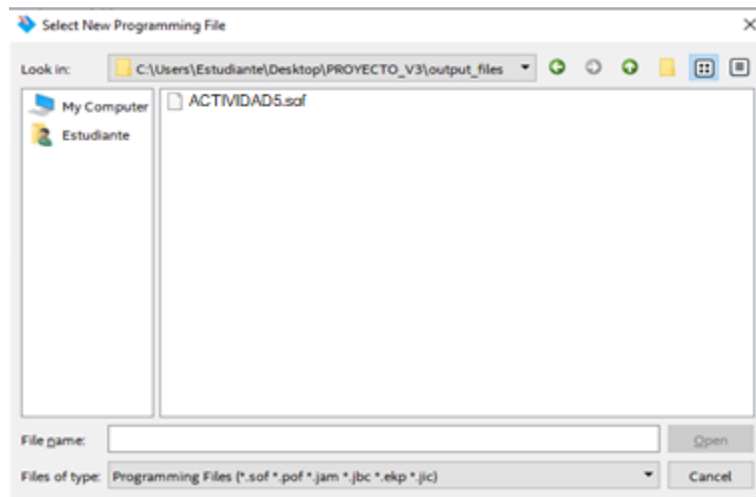


Ilustración 15: Elección del archivo. sof

21. Haga clic en la casilla correspondiente a la columna **Program/Configure** del archivo **ACTIVIDAD5.sof**, y luego seleccione **Start** para que se programe el chip FPGA. Para que la programación sea exitosa se debe de alcanzar el 100%, tal cual se muestra en la ilustración 16.

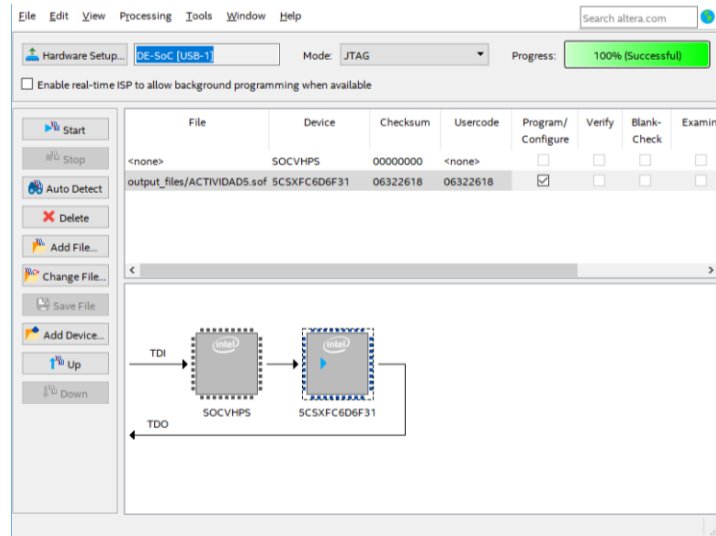


Ilustración 16: Programación exitosa del chip FPGA

22. En este momento se acaba de construir físicamente la arquitectura de computadora embebida con los 2 microprocesadores Nios II en el chip FPGA.

23. Realizamos la programación del microprocesador Nios II utilizando lenguaje C. En menu **Tools**, seleccione la opción **Nios II Software Build Tools for Eclipse**, tal como se muestra en la ilustración 17.

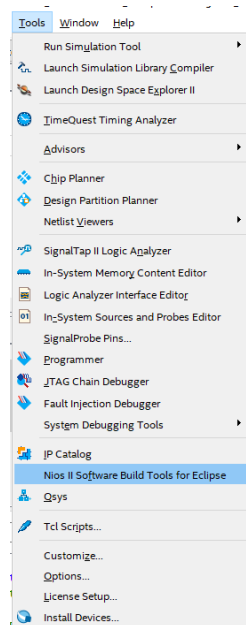


Ilustración 17 Selección de la herramienta Eclipse.

24. Seleccionamos la carpeta **SISTEMA** que se encuentra dentro de nuestra carpeta de proyecto para alojar el nuevo proyecto de Eclipse, como se muestra en la ilustración 18.

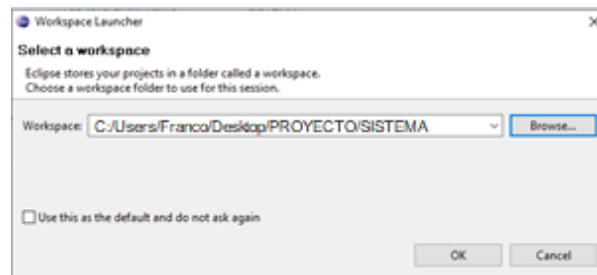


Ilustración 18: Ubicación del proyecto en la plataforma Eclipse

25. Inmediatamente se abra el entorno de programación Eclipse, procedemos a crear un nuevo proyecto, seleccionamos **File → New → Nios II Application and BSP from Template** tal como se ve en la ilustración 19.

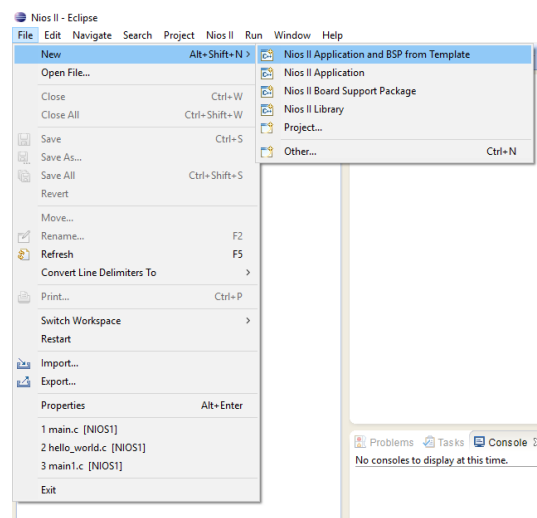


Ilustración 19: Selección de una nueva Aplicación para Nios II.

26. En la opción **SOPC Information File name**, buscamos el archivo **Sistema.sopcinfo** que tendrá toda la información de la computadora embebida, como se muestra en la ilustración 20.

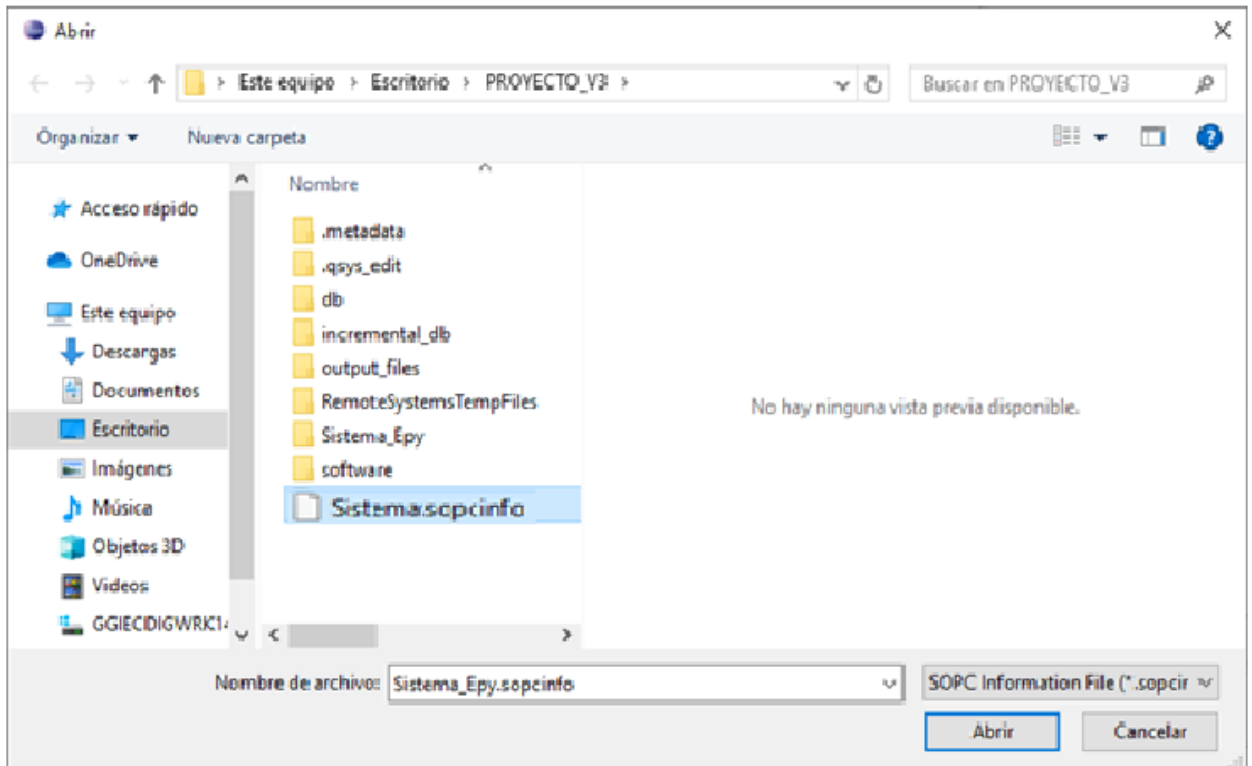


Ilustración 20: Elección del archivo .sopcinfo

27. En la opción **CPU name** escogemos el núcleo CPU1 disponibles en la computadora embebida. En **Project name** elegimos el nombre del proyecto, utilice el nombre **NIOS1**. En **Project Template** seleccionamos la opción **Hello World** y clic en **FINISH**, tal como se muestra en la ilustración 21.

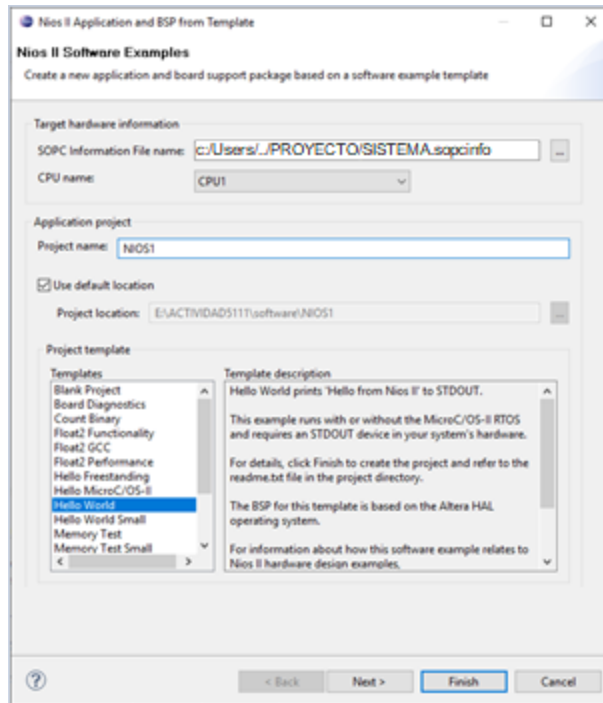


Ilustración 21 Creación de la aplicación del primer núcleo en software.

28. Inmediatamente se abra el entorno de programación Eclipse, procedemos a crear un nuevo proyecto, seleccionamos **File → New → Nios II Application and BSP from Template** tal como se ve en la ilustración 22.

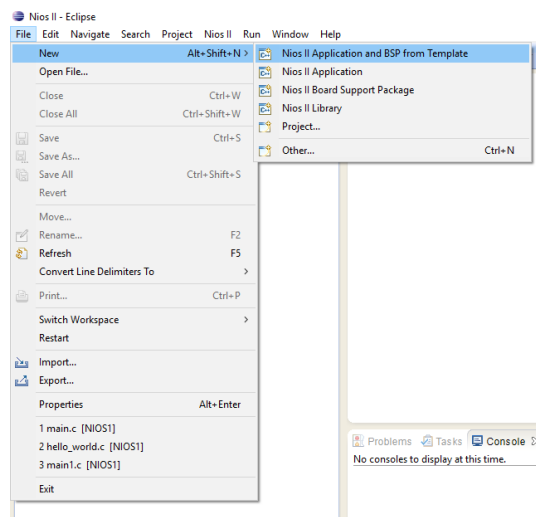


Ilustración 22: Selección de una nueva Aplicación para Nios II.

29. En la opción **SOPC Information File name**, buscamos el archivo **Sistema.sopcinfo** que tendrá toda la información de la computadora embebida.
30. En la opción **CPU name** escogemos el núcleo CPU2 disponibles en la computadora embebida. En **Project name** elegimos el nombre del proyecto, utilice el nombre **NIOS2**. En **Project Template** seleccionamos la opción **Hello World** y clic en **FINISH**, tal como se muestra en la ilustración 23.

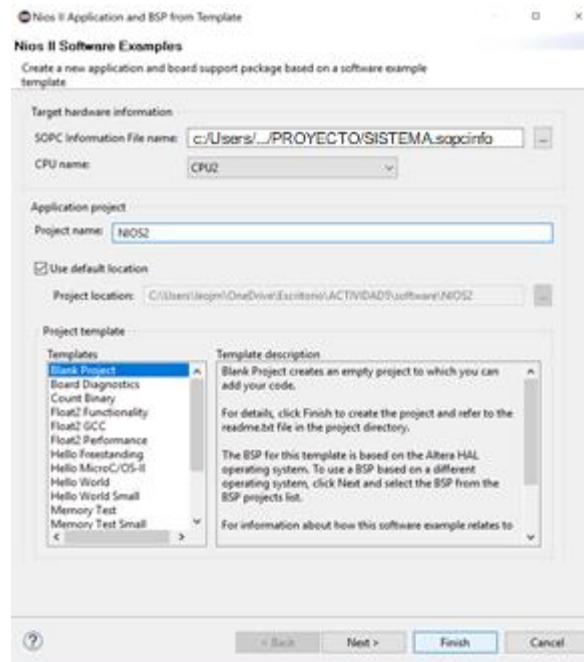


Ilustración 23 Creación de la aplicación del primer núcleo en software.

31. En la subventana de **Project Explorer** visualizamos 4 carpetas: Nios1 Nios2 y Nios1_bsp y Nios2_bsp. En la carpeta de NIOS1 y Nios2 podemos crear los archivos de programación en lenguaje C/C++ para nuestros microprocesadores, y las carpetas restantes contienen todos los componentes del hardware como se observa en la ilustración 24.

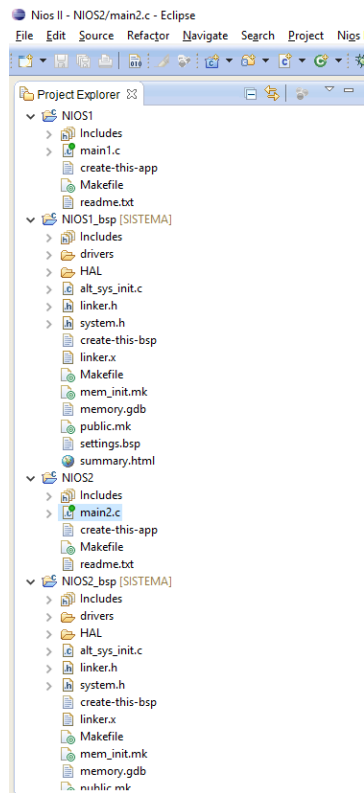


Ilustración 24: Ventana del navegador de proyectos.

32. Proceda a hacer clic derecho a la carpeta **NIOS1_bsp[SISTEMA]** y seleccione la opción **Properties**, tal como se muestra en la ilustración 25

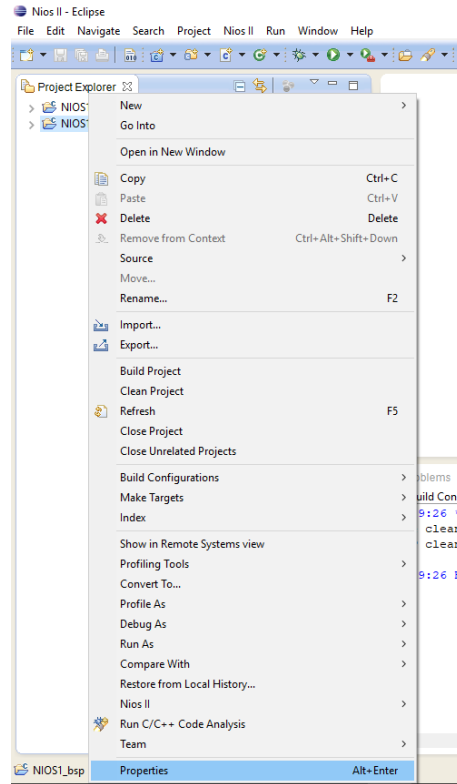


Ilustración 25: Propiedades del archivo NIOS1_bsp.

33. Inmediatamente se abrirá la ventana de propiedades, diríjase a la categoría **Nios II BSP Properties**, y deshabilite la casilla **Support C++** y habilite la casilla **Reduce device drives**. Tal como se muestra en la ilustración 26. Luego dar clic en OK.

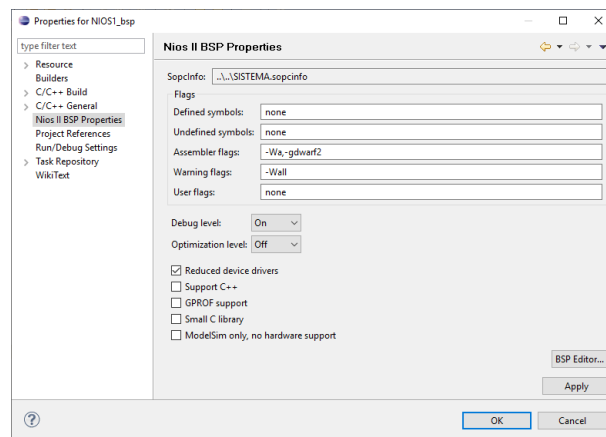


Ilustración 26: Selección para librerías reducidas de la carpeta NIOS1_bsp.

34. Copiamos la programación del archivo **RedNeuronal.txt** que se encuentra en la carpeta del proyecto, en el archivo **hellow_world.c** de la carpeta **NIOS1**. Asimismo, copiamos la programación del archivo **ProyeccionVGA.txt** que se encuentra en la carpeta del proyecto, en el archivo **hellow_world.c** de la carpeta **NIOS2**.
35. Proceda a hacer clic derecho a la carpeta **NIOS1_bsp[SISTEMA]**, seleccione la opción **NIOS II** y luego seleccione la opción **BSP EDITOR**, tal como se visualiza en la ilustración 27.

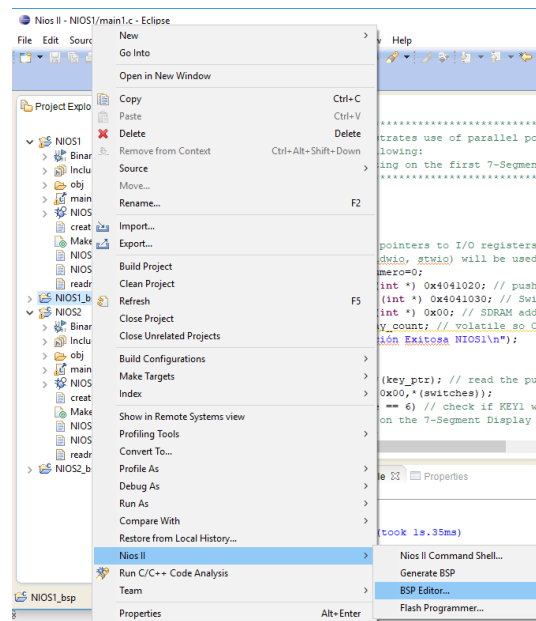


Ilustración 27: **BSP EDITOR** de la carpeta **NIOS1_bsp**

36. Asegúrese que tanto **NIOS1** y **NIOS2** se encuentren ejecutando desde sus espacios de memoria, como se visualiza en la ilustración 28. En el caso de **NIOS1** se ejecuta desde **onchip_mem1**, y **NIOS2** se ejecuta desde **onchip_mem2**.

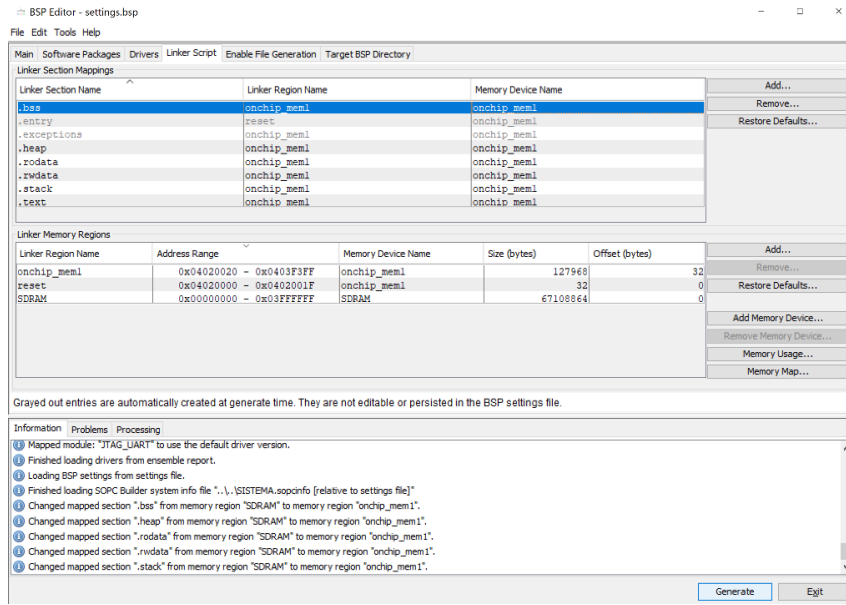


Ilustración 28: Revisión de los espacios de memoria de cada procesador.

37. Luego, se procede a guardar y compilar todo el sistema nuevamente.



Ilustración 29: Compilación de todo el sistema

38. Inicialmente, se ejecutará el procesador **NIOS1** que contiene la red neuronal con la predicción de pasos. Ingrese los datos, en **Nios II Console**, correspondientes a los pasos a predecir por la Red Neuronal y observe los resultados, tal como se muestra en la ilustración 30.

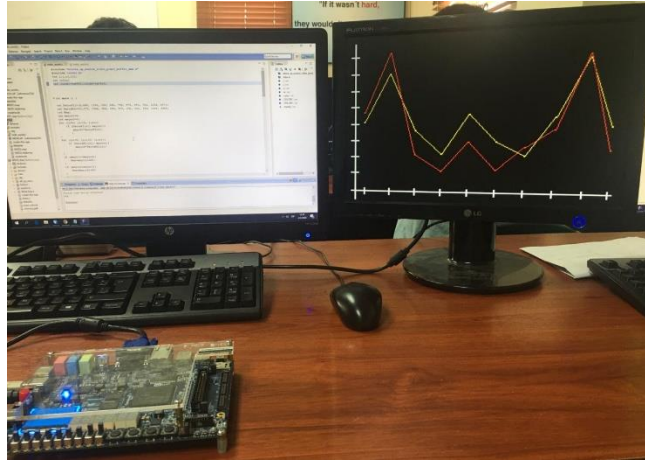


Ilustración 32: Proyección de datos estadísticos por puerto VGA.