

## SISTEMAS DIGITALES II

### CLASE DE PROBLEMAS # 6

**CAPÍTULO DEL CURSO:** FUNDAMENTOS DE MICROCONTROLADORES Y SISTEMAS EMBEBIDOS

**TEMA DE LA ACTIVIDAD:** SIMULACIÓN DE UN SISTEMA DIGITAL BASADO EN MICROCONTROLADOR

**OBEJTIVOS DE APRENDIZAJE:**

- Ejecutar una memoria rom de una arquitectura de un microcontrolador a través de un archivo de texto con extensión .mif
- Analizar la simulación en un diagrama de tiempo de las principales señales que conforman la arquitectura de un microcontrolador.
- Diseñar el diagrama ASM del bloque de control del microcontrolador en vhdl, utilizando las señales del microcontrolador para la verificación de los estados de la máquina secuencial sincrónica.

**DURACIÓN:** 100 minutos

**MATERIALRES Y HERRAMIENTAS:**

- Software Intel Quartus Prime versión 19.1

**MARCO TEÓRICO:**

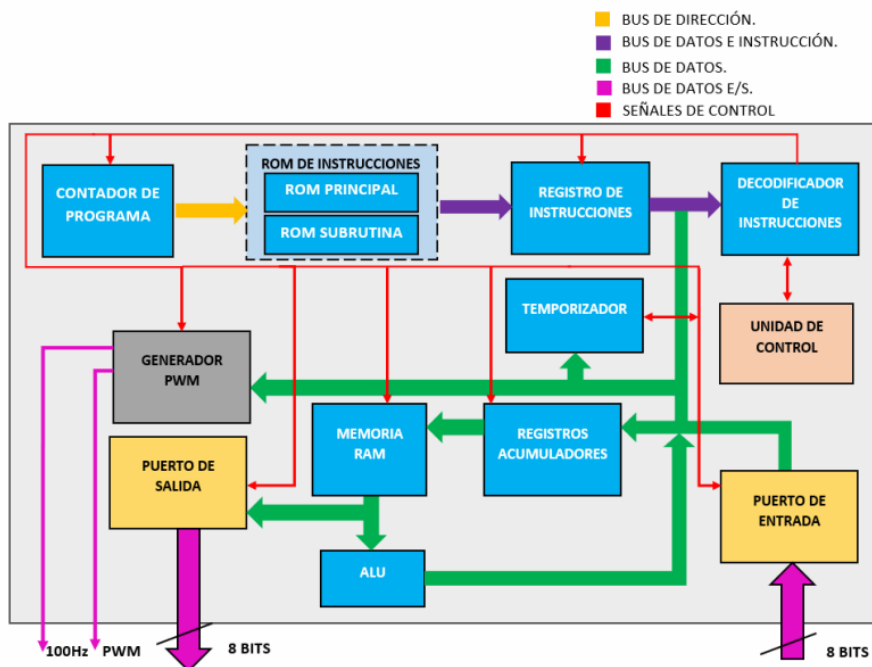


Ilustración 1. Diagrama de bloques de la arquitectura diseñada del microcontrolador

Un microcontrolador con arquitectura tipo Harvard es una arquitectura de computadora en la que físicamente se encuentran separadas las instrucciones y los datos. La arquitectura de

**Elaborador por:**

Ing. Arnold Pinto & Ing. Carlos Espinoza

**Editado por:**

Mgs. Alisson Constantine Macías

Jefe del Laboratorio Sistemas Digitales Avanzados

este microcontrolador está básicamente formada por una memoria ROM que contiene el archivo del programa donde se almacenan las instrucciones y una memoria RAM que permitirá almacenar la información ya sea ingresada por el usuario a través de los pines de entrada del microcontrolador o generada por las operaciones que ejecute la Unidad Aritmética Lógica ALU. Adicional está conformada por bloques de registros que permite almacenar valores para propósito particular, por bloques multiplexores, un contador de programa y un decodificador de instrucciones. Se cuenta con una Unidad de Control, que constituye un bloque, el cual permitirá realizar el proceso secuencial del microcontrolador que es buscar, decodificar, y ejecutar la instrucción, como se detalla en la ilustración 1.

El set de instrucciones guardado en la memoria ROM del microprocesador consta de 19 bits, divididos en: 1 bits para el acarreo de entrada, 6 bits para el código de la instrucción, 8 bits de Propósito General, 2 bits para direccionamiento, 1 bit para fuente y 1 bit para indicar el destino, como se muestra en la ilustración 2.

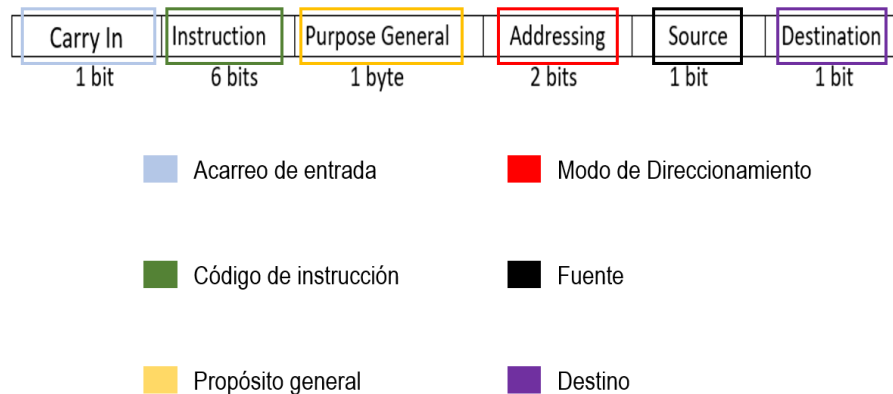


Ilustración 2. Set de Instrucciones

El microcontrolador posee 2 grupos de instrucciones, el primer grupo conformado por aquellas que no utilizan la ALU (Ilustración 3) y el segundo grupo por aquellas instrucciones que requieren de la ALU (Ilustración 4).

No utilizan la ALU	Descripción	Mnemotécnico	Código
	Salto si la bandera de cero es verdadera	JIFZ	000000
	Salto si la bandera de acarreo es verdadera	JIFC	000010
	Salto incondicional	JUMP	000100
	Llamada a subrutina	CALL	000110
	Retorno de Subrutina	RETURN	001000
	Almacena el valor del acumulador en la RAM	STORE	001010
	Lectura en puerto	INPUT	001100
	Escritura en puerto	OUTPUT	001110
	Fin	FIN	010100
	Clear	CLEAR	010110
	Activa PWM	PWM	011000
	Retardo	DELAY	011110

Ilustración 3. Set de Instrucciones que no utilizan la ALU

**Elaborador por:**  
 Ing. Arnold Pinto & Ing. Carlos Espinoza  
**Editado por:**  
 Mgs. Alisson Constantine Macías  
 Jefe del Laboratorio Sistemas Digitales Avanzados


## Utilizan la ALU

Descripción	Mnemotécnico	Código	Afecta	
			Cero	Acarreo
Suma	ADD	100000	1	1
Suma con acarreo	ADDC	100010	1	1
Resta	SUB	100100	1	1
Resta con acarreo	SUBC	100110	1	1
Incrementar	INC	101000	1	1
Decrementar	DEC	101010	1	1
Desplazamiento a la izquierda (sin acarreo)	SHL	101100	1	0
Desplazamiento a la derecha (sin acarreo)	SHR	101110	1	0
Desplazamiento circular a la izquierda	ROL	110000	1	1
Desplazamiento circular a la derecha	ROR	110010	1	1
Operadores Lógicos	AND	110100	1	0
	OR	110110	1	0
	XOR	111000	1	0
	NOT	111010	1	0
Cargar valor en el acumulador	LOAD	111100	1	0
Mover valores de acumuladores	MOVE	010100	1	0
Multiplicación	MULT	100001	1	1
División	DIV	100011	1	1
Porcentaje PWM	PDC	100101	1	1

Ilustración 4. Set de Instrucciones que Utilizan la ALU

**DESCRIPCIÓN DE LA PRÁCTICA:** Se analizarán los bloques del sistema digital al igual que el diagrama de tiempo donde se visualizarán las principales señales que componen la arquitectura del microcontrolador.

### PROCEDIMIENTO:

1. Proceda a crear una carpeta en el escritorio con el nombre **PracticaMicro** (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 19.1 Lite 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 Figura 1. Aparecerá la ventana de Introducción, luego da clic en **Next**.

**Elaborador por:**

Ing. Arnold Pinto & Ing. Carlos Espinoza

**Editado por:**

Mgs. Alisson Constantine Macías

Jefe del Laboratorio Sistemas Digitales Avanzados

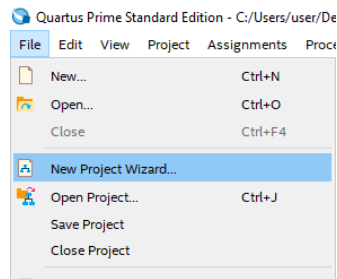


Figura 1. Creación de un nuevo proyecto.

- En la primera línea escoja la ruta donde va a crear su proyecto, en este caso, la ruta será la carpeta **PracticaMicro** que fue creada en el procedimiento 1 y que se encuentra en el escritorio. En la siguiente línea escoja como nombre del proyecto **MicroPrueba**, tal como se visualiza en la Figura 2. Luego de clic en **Next**.

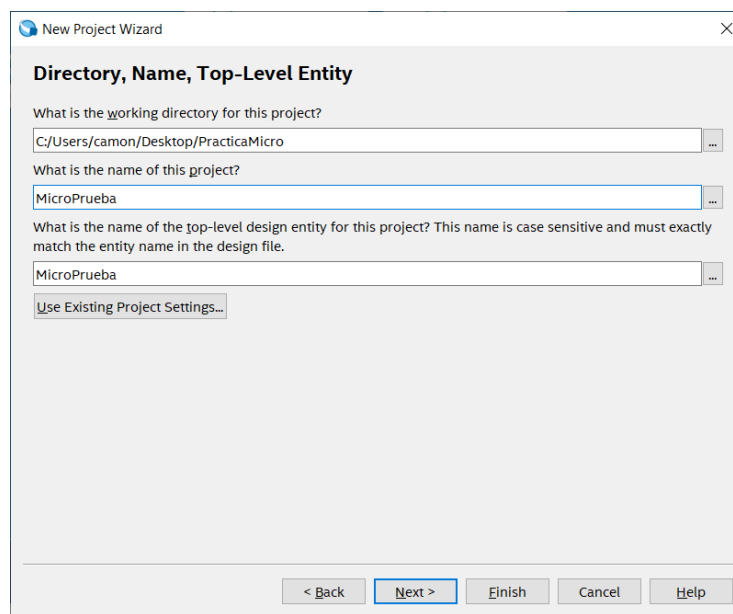


Figura 2. Nombre y Ubicación del Proyecto.

- 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 Figura 3. Luego de clic en **Next**.

**Elaborador por:**

Ing. Arnold Pinto & Ing. Carlos Espinoza

**Editado por:**

Mgs. Alisson Constantine Macías

Jefe del Laboratorio Sistemas Digitales Avanzados

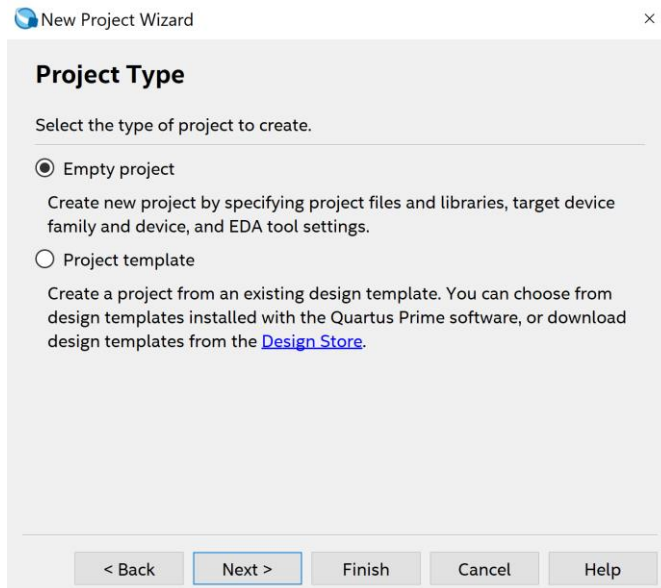


Figura 3. Tipo del proyecto.

- 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. Por último, haga clic en **Next**.

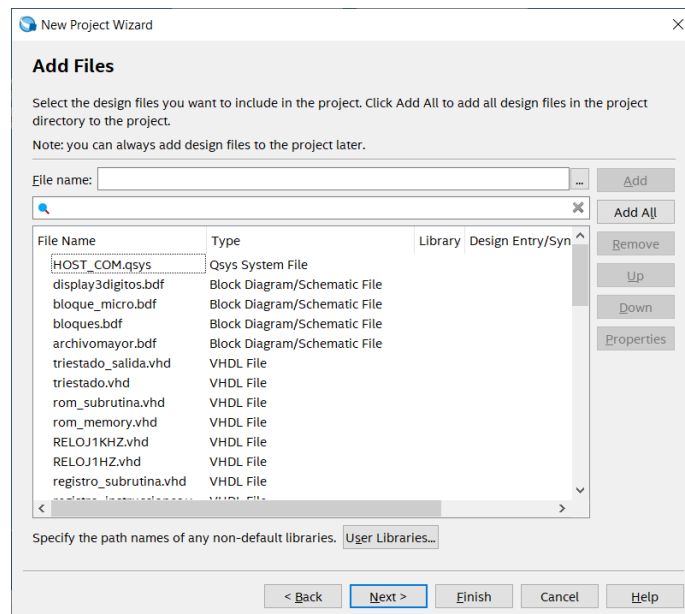


Figura 4. Adhesión de archivos.

- 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 Figura 5. Luego haga clic en **Next**.

**Elaborador por:**

Ing. Arnold Pinto & Ing. Carlos Espinoza

**Editado por:**

Mgs. Alisson Constantine Macías

Jefe del Laboratorio Sistemas Digitales Avanzados

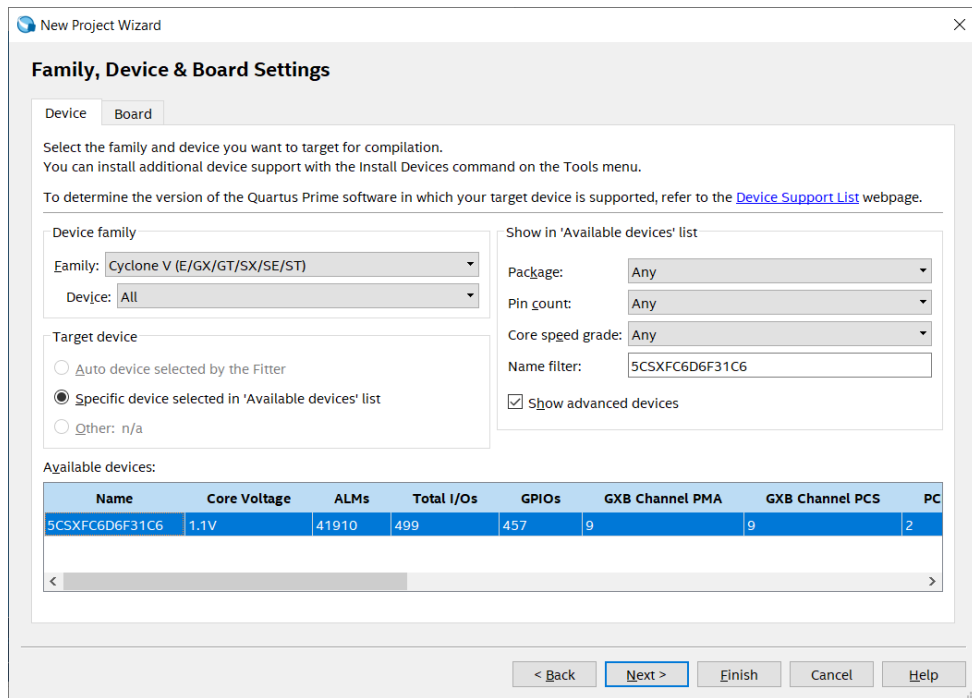


Figura 5. Familia y modelo del chip FPGA SoC.

8. A continuación, en la siguiente ventana recuerde habilitaremos de **ModelSim-Altera** en la columna **Tool Name** de Simulation y en la columna **Format(s)** VHDL, como se muestra en la figura 6. Luego click en **FINISH**.

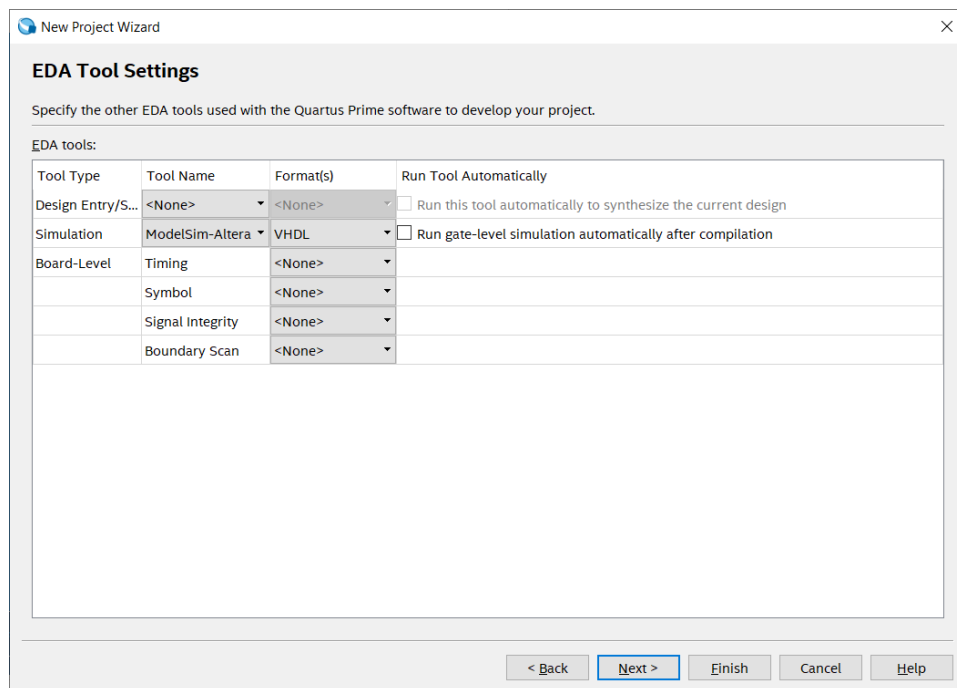


Figura 6. Selección del modo ModelSim-Altera y formato VHDL para la simulación del proyecto

9. A continuación, diríjase a la carpeta donde se encuentra guardado el proyecto y seleccione el archivo con extensión .mif llamado **informacion.mif** tal como se muestra en la figura 7. Ábralo como una nota de texto o con Notepad++.

**Elaborador por:**

Ing. Arnold Pinto & Ing. Carlos Espinoza

**Editado por:**

Mgs. Alisson Constantine Macías

Jefe del Laboratorio Sistemas Digitales Avanzados


10. Copie y pegue lo que se encuentra en el archivo **CODIGO1.txt** en el archivo **información.mif**

```

1 WIDTH=19;
2 DEPTH=256;
3
4 ADDRESS_RADIX=UNS;
5 DATA_RADIX=BIN;
6
7 CONTENT
8 BEGIN
9 0 : 0111100000010001100; ---LOAD CON DIRECCIONAMIENTO INMEDIATO---
10 1 : 0001010000000000100; ---STORE---
11 2 : 0111100000000111100; ---LOAD CON DIRECCIONAMIENTO INMEDIATO---
12 3 : 000101000000010100; ---STORE---
13 4 : 0111110000000000000; ---MOVE
14 5 : 0111100000000000100; ---LOAD---
15 6 : 0100001000000000000; ---MULT---
16 7 : 0001010000000100100; ---STORE--ADD 0X02
17 8 : 0111100000000101100; --LOAD CON DIRECCIONAMIENTO INMEDIATO--
18 9 : 0111110000000000000; ---MOVE
19 10 : 0100001000000100000; ---MULT---
20 11 : 0001010000001100100; --STORE-- ADD 0X06
21 12 : 1011000000000000001; ----ACTIVA PWM
22 13 : 0100101000001100000; ----CALCULO DEL DUTY% PWM
23 14 : 0001010000001110100; ----STORE--EN LA DIRECCIÓN 0X07
24 15 : 0001110000001110000; ---OUTPUT
25 16 : 0010100000001110000; ---FIN

```

Figura 7. Selección y grabado del archivo con extensión .mif que contiene la información del programa

11. Proceda a compilar el archivo **bloque\_micro.vhd** 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 Figura 8. Luego seleccione el ícono .

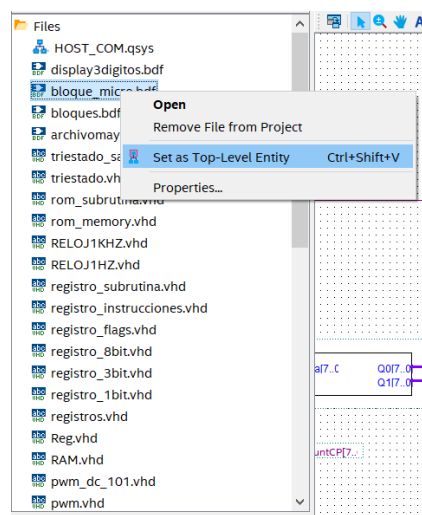


Figura 8. Asignación de la más alta jerarquía al archivo **bloque\_micro.bdf**

**Elaborador por:**  
 Ing. Arnold Pinto & Ing. Carlos Espinoza  
**Editado por:**  
 Mgs. Alisson Constantine Macías  
 Jefe del Laboratorio Sistemas Digitales Avanzados

12. Cree un nuevo archivo de simulación, para ello diríjase a **File**→**New**→**University Program VWF** como se muestra en la figura 9.

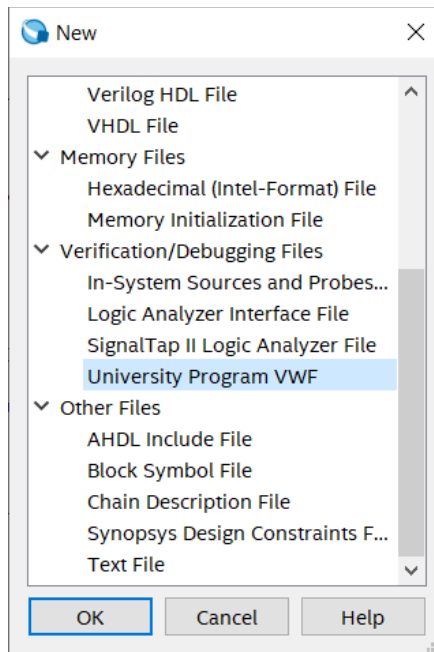


Figura 9. Creación del archivo de University Program VWF

13. A continuación, proceda a colocar todas las señales en orden de prioridad dentro del diagrama de tiempo como se observa en la siguiente figura 10.

**Elaborador por:**

Ing. Arnold Pinto & Ing. Carlos Espinoza

**Editado por:**

Mgs. Alisson Constantine Macías

Jefe del Laboratorio Sistemas Digitales Avanzados



	Name	Value at 0 ps
in	clock	B 0
in	reset_MSS	B 0
in	start	B 0
out	> estados	U 1
out	> Q_CountCP	U 0
out	> Add_RAM	U 0
out	> In_DecoInst	B 000000
in	> Dato_In	U 0
out	C	B X
out	clear	B 0
out	EnAcum1	B 0
out	EnAcum2	B 0
out	EnAdd	B 0
out	EnB	B 0
out	EnC	B 0
out	EnCall	B 0
out	EnCp	B 0
out	EnDeco	B 0
out	EnRins	B 0
out	EnSub	B 0
out	fin	B 0
out	modo	B 0
out	op	B 0
out	> Pins_Salida	B 00000000
out	> InA_ALU	U 0
out	> InB_ALU	U Z
out	> Q_Mux1	U X
out	> Q_Mux2	U 0

Figura 10. Señales correspondientes al archivo bloque\_micro.bdf

**Elaborador por:**

Ing. Arnold Pinto & Ing. Carlos Espinoza

**Editado por:**

Mgs. Alisson Constantine Macías

Jefe del Laboratorio Sistemas Digitales Avanzados

14. Para la señal del **clock** establezca un clock de 10 [ns], mientras que a la señal **reset\_MSS** establézcalo en un valor alto y para la señal **start** asígnele un valor alto por un tiempo aproximado de 100 ns, tal como se indica en la figura 10.

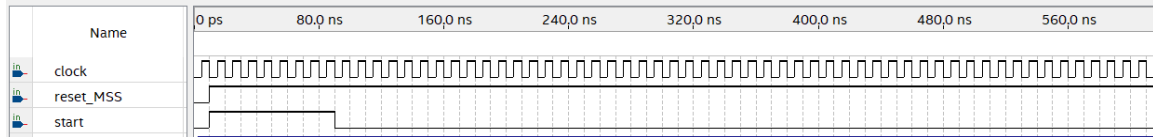



Figura 1. Asignación de valores lógicos a las principales señales de entrada del archivo *bloque\_micro.bdf*

15. Ahora guarde el archivo de la simulación con el nombre que viene asignado por default como **Waveform.vwf** y proceda a realizar la simulación del archivo dando click en el ícono **Run Functional Simulation** .
16. De las señales de la simulación del **Simulation Waveform Editor** que se obtienen, proceda a contestar la preguntas.

**PREGUNTAS:**

1. Adjunte una captura de pantalla del reporte de la simulación de la ventana del Simulation Waveform Editor y realice la respectiva explicación de cada una de las señales.
2. Adjunte captura de pantalla de la partición funcional (visor RTL) del archivo *bloque\_micro.bdf*.
3. En base a la pregunta anterior determine los bloques utilizados del microcontrolador y realice una breve descripción de cada uno de ellos en donde indica su función.
4. Elabore el diagrama ASM de la unidad de control del microcontrolador en base al archivo *waveform.vwf*, indicando para que sirve cada uno de los estados.
5. Del bus de señal denominado **In\_DecoInst** enliste cada una de las instrucciones que ejecuta el programa y en base a la hoja de instrucciones que posee el microcontrolador, identifique cada una de ellas.
6. Del primer código de programa grabado en la rom, calcule el tiempo de ejecución del programa por medio de la siguiente ecuación, tenga en cuenta el clock, y las instrucciones generadas por el archivo *waveform.vwf*.

$$tiempo\ ejecución = (\#instrucciones) \left( \frac{ciclos}{instruccion} \right) (Periodo\ del\ clock)$$