ARQUITECTURA DEL COMPUTADOR

TEMPORIZADOR FITNESS PROGRAMABLE

(ENTREGA FINAL)

PROFESOR: JOSÉ OLIDEN SÁNCHEZ

DAVID RIVERO URBANO

DIEGO F. GALARZA CH.

ELIZABETH C. MONCADA D.

NICOLÁS A. ORTIZ A

NOVIEMBRE 2018

ÍNDICE

1. TEMPORIZADOR FITNESS

Un temporizador fitness es una herramienta de fácil uso, que le permite a la persona llevar un registro organizado de su rutina diaria de ejercicios.; es un dispositivo de pequeño tamaño, compacto, que lleva un registro de sesiones de entrenamiento y ejercicios y dispone la informacion sobre cada sesion en su pantalla integrada. Actualmente aún se usan dispositivos temporizadores, pero se ha integrado esta funcionalidad mucho mas en el mundo de los smartphones, haciendo una simple aplicación que aprovecha las caracteristicas del telefono inteligente para disponer y mostrar la informacion requerida por el usuario durante su tiempo de ejercicio. El temporizador se encarga de mostrarle al usuario una jornada que ha programado el mismo; dicha jornada consta de un tiempo de ejercicios y un descanso, y además también se encarga de mostrar el manejo del tiempo sobre las rutinas ejecutadas.

El dispositivo temporizador programa en minutos y/o segundos el entrenamiento del usuario y cuenta con ayudas visuales para determinar el inicio y fin de cada momento dentro de una sesión; las sesiones de entrenamiento tienen una duración máxima de una hora. El usuario puede seleccionar el entrenamiento que desee y se le indica un conteo regresivo de la sesion seleccionada, ademas de indicarse tambien el tiempo por cada ejercicio ejecutado y el tiempo de descanso tomado.

1.1 REQUERIMIENTOS DETALLADOS:

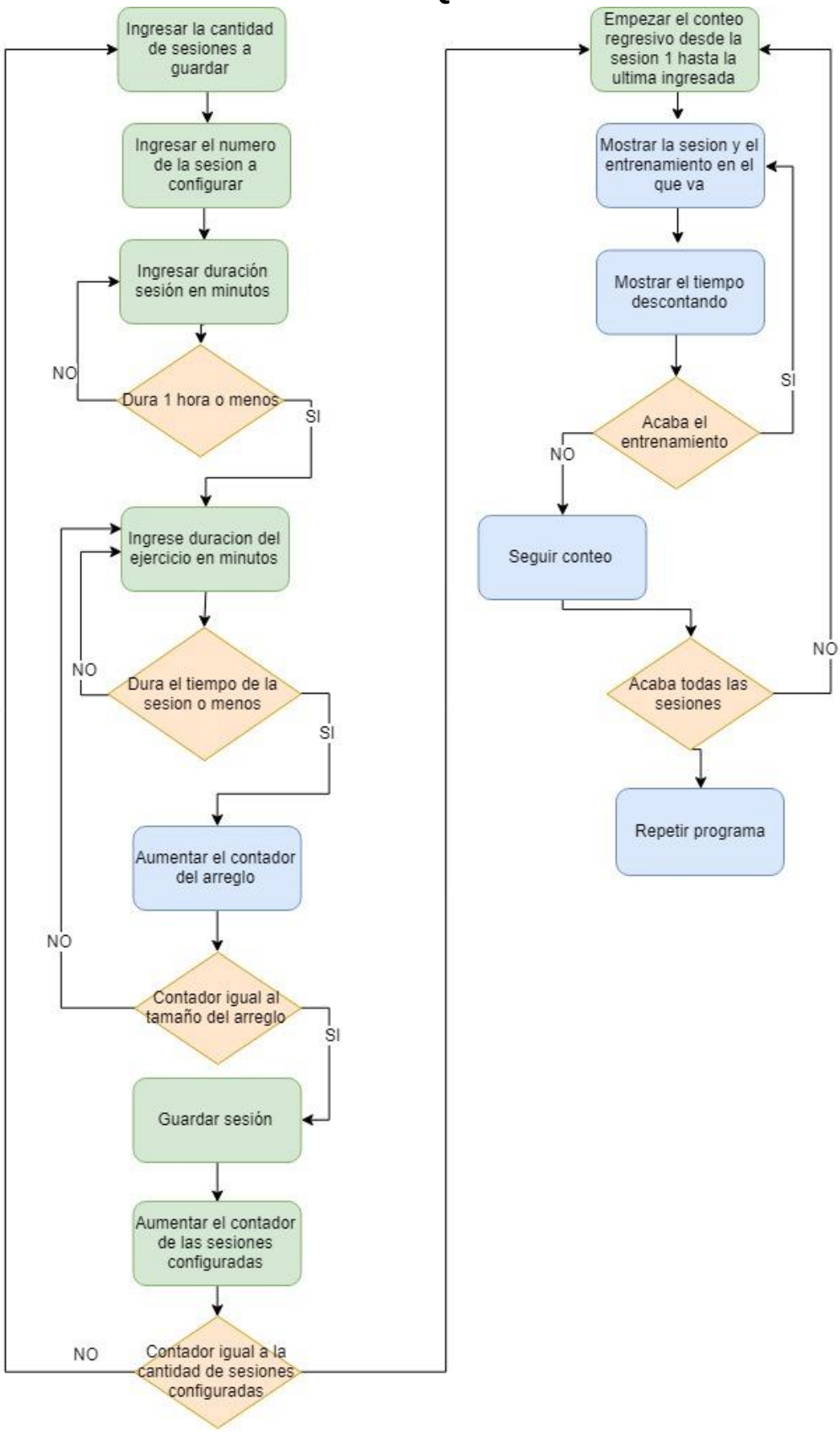
Como sabemos nuestro dispositivo va a ser implementado desde el dispositivo FPGA, usando claramente las herramientas MIPS y QUARTUS II de Altera.

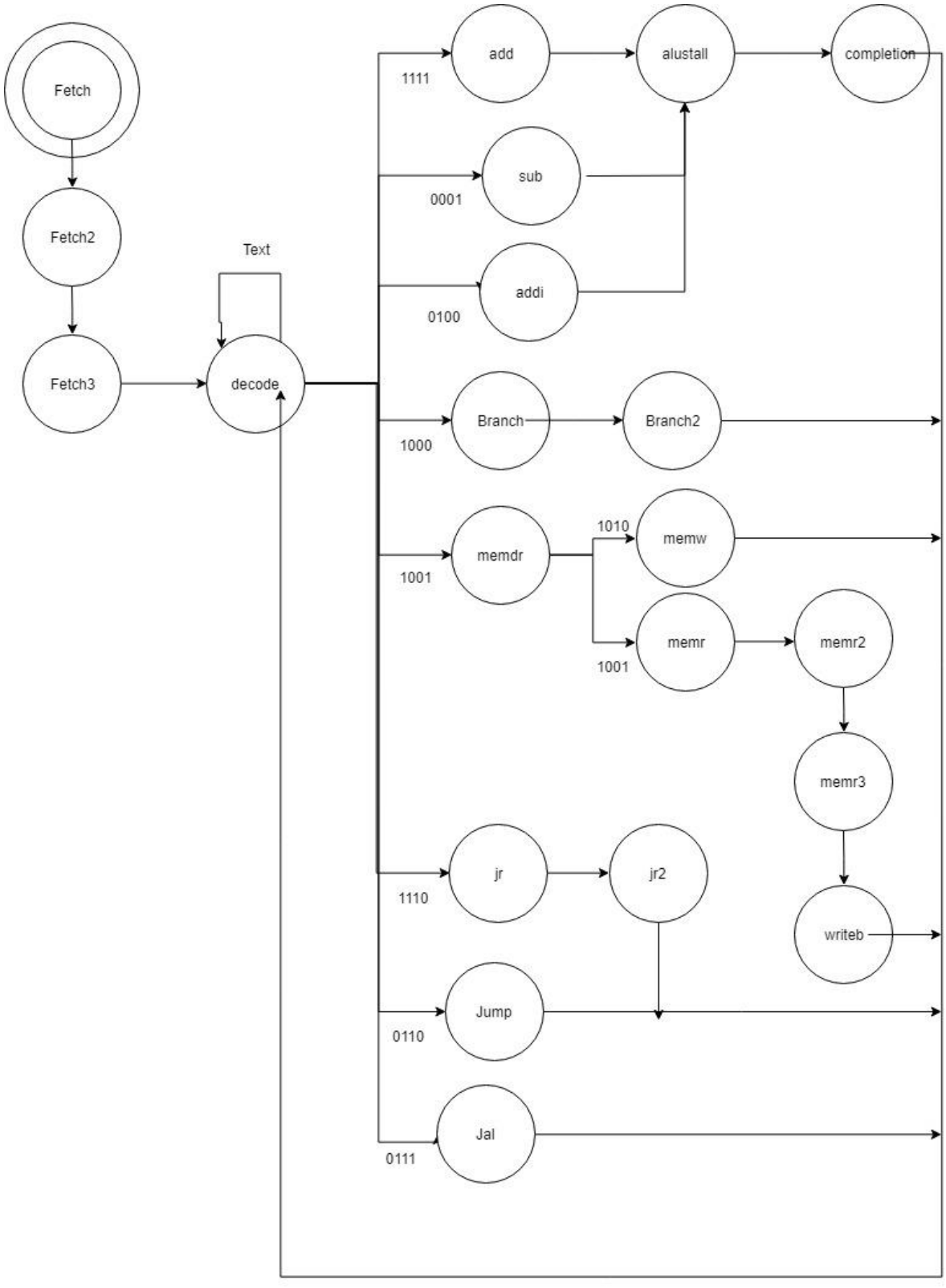
Nuestro dispositivo tiene una interfaz de visualización donde el usuario puede de cierta forma observar el número de sesiones de su entrenamiento, la propia duración de esas sesiones. El dispositivo le informa al usuario el inicio de cada sesión, además del número de circuitos de ejercicio y la duración de este. Y por último le informa el número de descansos con los que contara, la duración de cada uno y el momento en el que inicie.

Cuando el usuario se encuentra en una sesión de entrenamiento o tomando un descanso, cuando se le es indicado, el dispositivo le informa por medio de un contador regresivo al usuario, el tiempo que le queda para terminar su determinada sesión o ya sea su descanso.

El dispositivo cuenta con una memoria para almacenar las sesiones que el usuario desee, las cuales son mayor a 5 sesiones, con una duración de al menos 1 hora cada una. Dichas sesiones son programadas por el usuario manualmente indicando un número de sesiones y su respectivo tiempo de una hora o menos. Por cada sesión, se especifica el inicio y duración de los circuitos de ejercicios y los descansos. Tras haber seleccionado la sesion indicada se le muestra al usuario el conteo regresivo de la informacion almacenada previamente, hasta dar finalización al entrenamiento completo en el tiempo indicado inicialmente por el usuario. Para una guia mas precisa del funcionamiento de nuestro dispositivo temporizador, se le brinda al usuario un manual con un instructivo paso a paso sobre como usar nuestro programa en el dispositivo FPGA.

2. DIAGRAMA DE FLUJO Y MAQUINA DE ESTADOS




Fetch
Fetch2
Fetch3
Text
decode
add
1111
alustall
completion
sub
0001
addi
0100
Branch
1000
Branch2
memdr
1001
1010
memw
memr
1001
memr2
memr3
jr
1110
jr2
writeb
Jump
0110
Jal
0111

3. CRITERIOS DE DISEÑO

Debido a la familiaridad con la arquitectura multiciclo trabajada durante el curso, esta será la que se implementó para la realización del proyecto. Además, a esta se le añadirán los formatos de instrucciones que se utilizan en MIPS, el cual incluye tres tipos, las cuales son R, I, J.

TIPO R

OP	RS	RT	RD	FUNC
20:17	16:13	12:9	8:5	10:6

TIPO I

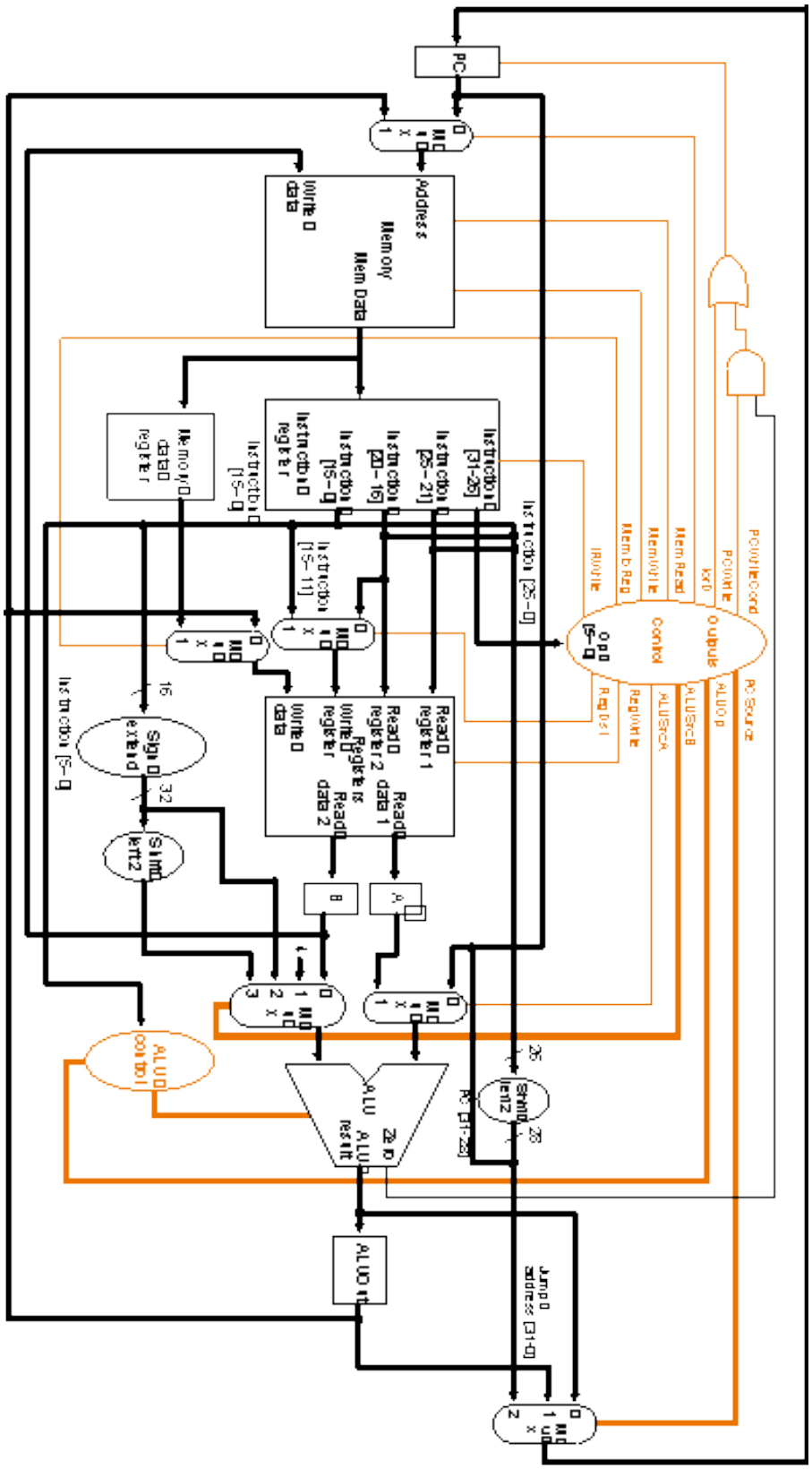
OP	RS	RT	ADDRESS
20:17	16:13	12:9	8:0

TIPO J

	RS	RT	ADDRESS
20:17	16:13	12:9	8:0

- El campo de OP especifica el operador, que denota la instrucción que se utilizará.
- El campo RS especifica un registro con la cual se operará.
- El campo RT especifica un registro con la cual se operará si existe un registro RD. En el caso de que no exista, este funcionará como registro destino.
- El campo RD especifica un registro destino, en el que se guardarán las operaciones hechas por los demás registros.
- El campo de Funct especifica la operación aritmética básica que se utilizará.
- La Memoria de Instrucciones cuenta con 29 posiciones de 21 bits cada uno. En cada una de estas posiciones se guardará una de las instrucciones implementadas.
- El tamaño de los buses de direccionamiento y de palabras será de 32 bits, para adecuarse a los formatos de instrucciones utilizadas en MIPS.

4. DIAGRAMA ESTRCUTURAL DE LA ARQUITECTURA DEL PROCESADOR



5. DESCRIPCIÓN Y FORMATO DE INSTRUCCIONES (ISA)

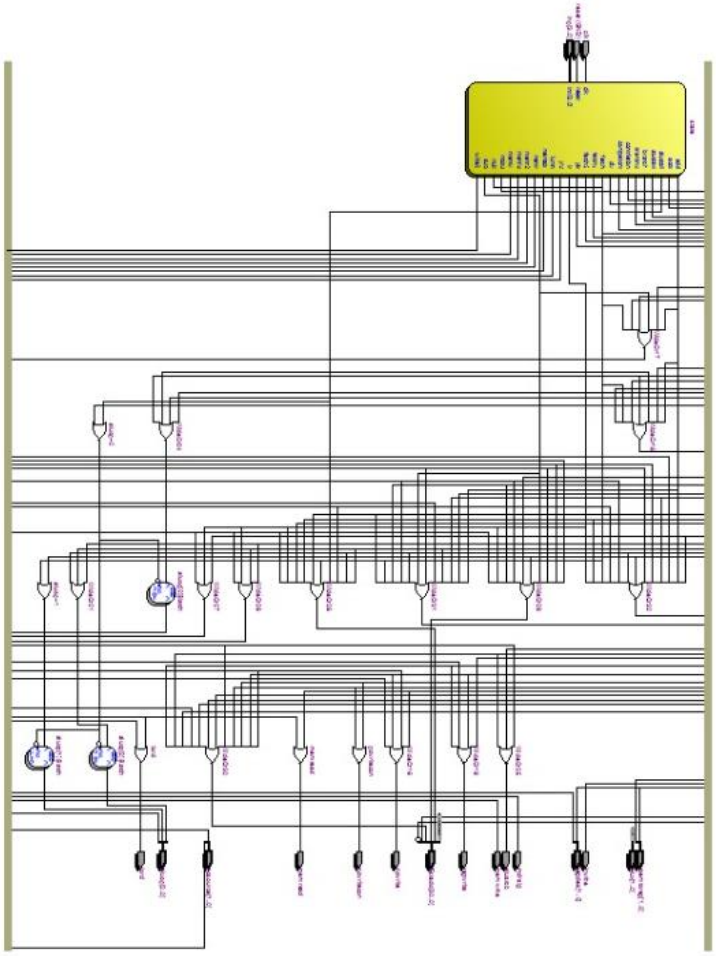
Para el desarrollo de nuestro proyecto utilizamos el formato de instrucciones de MIPS, lo cual nos permitió tener un manejo más fácil y rápido de funciones y una estructura más adecuada para el temporizador, pues gracias a los conocimientos obtenidos en las clases de Arquitectura del Computador y en las monitorias de la misma asignatura, logramos obtener un dominio sobre dicho formato de instrucciones.

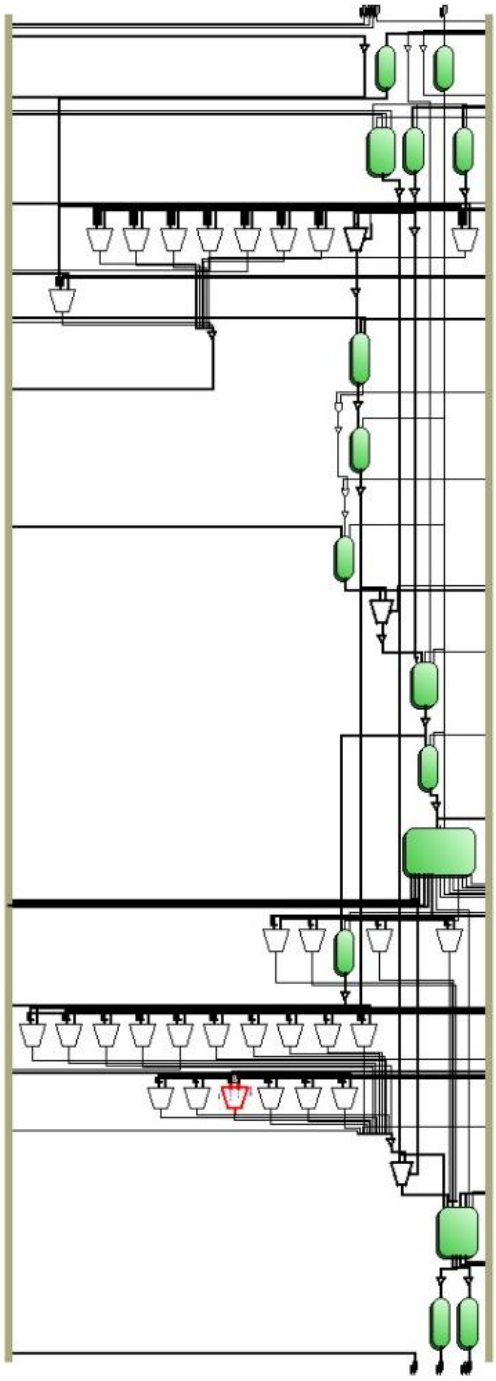
El temporizador cuenta esencialmente con las instrucciones basicas de MIPS, las cuales están descritas más a detalle en el cuadro adjunto al final de la página; dichas instrucciones nos permitieron realizar comparaciones en los distintos eventos de cada parte del temporizador, así como verificar el correcto funcionamiento de los 7-segmentos para mostrar el trascurrir de los segundos de la manera mas fiable y cercana a un temporizador real.

Cuadro de instrucciones manejadas en el Proyecto.

NAME MNEMONIC		FORMAT	OPERATION
Add	add	R	R[rd]=R[rs]+R[rt]
Add Inmediate	addi	I	R[rd]=R[rs]+SignExtImm
Branch On Equal	beq	I	if (R[rs]=R[rt]) PC=PC+4+BranchAddr
Branch On Not Equal	bne	I	if (R[rs]!=R[rt]) PC=PC+4+BranchAddr
Load Word	lw	I	R[rt]=M[R[rs]+SignExtImm]
Store Word	sw	I	M[R[rs]+SignExtImm]=R[rt]
Jump	j	J	PC=JumpAddr

6. DIAGRAMAS FUNCIONALES RESULTANTES





7. CONCLUSIONES

- Nuestro programa está enfocado en facilitar el entrenamiento y ejercicio físico de la persona, por lo cual sirve como herramienta de orden secuencial de los ciclos de deporte del usuario, permitiendo programar cada sesión de manera independiente con sus respectivos entrenamientos y descansos.

- Para el funcionamiento de nuestro programa desarrollado en MIPS, tuvimos muchas dificultades en el planteamiento del problema, por lo que primero decidimos crear dicho programa en el lenguaje de programación Python, para poder entender bien su funcionamiento base y así después transcribirlo en MIPS para probar su funcionamiento directamente en lenguaje ensamblador.

- Logramos desarrollar una arquitectura multiciclo funcional que recibiera instrucciones basicas de MIPS y las ejecutara en tiempo real en el dispositivo FPGA, para así realizar pruebas con nuestro codigo del programa hecho en MIPS.

- El codigo desarrollado en Python funciona correctamente por lo que logramos entender como debía funcionar nuestro código en MIPS y desarrollarlo haciendo uso de arreglos para guardar tiempos de cada rutina separando decimales y asi clasificar cada número de tal forma que sea recibido por nuestra multiciclo y sea representado de manera correcta en los 7-segmentos.

- Tenemos también un algoritmo que permite interpretar cualquier instrucción de nuestro ISA de MIPS en binario de tal forma que convierte nuestro disositivo en uno mas eficiente y da una respuesta mas rapida ante el usuario.

- Debido a la principal limitante en nuestro proyecto, la tarjeta FPGA, no desarrollamos un dispositivo compacto, funcional y adecuado a la vista del usuario, pero a través del uso de la FPGA podemos observar con mas claridad el funcionamiento correcto de nuestro programa.