

**PROBABILIDAD DE CRUCES DIHIBRIDOS ENTRE PADRE Y MADRE**

**INTEGRANTES:**  
**ANDRES FELIPE SALAZAR R.**  
**LAURA ARANGO MAYOR.**  
**TANIA C. OBANDO S.**  
**VERONICA TOFIÑO.**

**FACULTAD DE INGENIERIA**  
**CALI**

**VALLE DEL CAUCA**  
**2018**



**Pontificia Universidad**  
**JAVERIANA**  
Cali

## Definiciones:

**Gametos:** Células sexuales encargadas de la reproducción, gametos femeninos (óvulos), gametos masculinos (espermatozoides).

**Cigoto:** Célula que resulta de la unión de las células sexuales masculina y femenina y a partir de la cual se desarrolla el embrión de un ser vivo.

**Fenotipo:** Conjunto de caracteres visibles que un individuo presenta como resultado de la interacción entre su genotipo y el medio.

**Genotipo:** Conjunto de los genes que existen en el núcleo celular de cada individuo.

**Dominante y recesivo:** Este concepto alude al vínculo que establecen los alelos que forman parte de un mismo gen cuando uno de estos alelos consigue enmascarar la manifestación del fenotipo del otro alelo. El alelo que logra imponerse es el alelo **dominante**, mientras que el otro es el alelo **recesivo**.

**Nota:** Un alelo **dominante** y otro **recesivo**, para un determinado gen, tendrá el fenotipo **dominante**.

**Generaciones Filiales:** La Primera Generación Filial es la descendencia resultante del apareamiento controlado de la **generación parental**. Se da usualmente entre padres diferentes con genotipos relativamente puros.

- Cuando se cruzan dos individuos provenientes de la Primera Generación Filial (F1) su descendencia se llama Segunda Generación Filial (F2)
- Cuando se cruzan dos individuos provenientes de la Segunda Generación Filial (F2) su descendencia se llama: Tercera Generación Filial (F3).

## Introducción

El proyecto tiene como objetivo dar a conocer la probabilidad de que los hijos hereden ciertos rasgos físicos dominantes de los padres, de acuerdo con los cruces entre sus genotipos.

Para ello el proyecto se apoyará en dos Leyes de Mendel:

La primera establece que si se cruzan dos razas puras (un homocigoto dominante con un homocigoto recesivo) para un determinado rasgo, los descendientes de la primera generación serán todos híbridos iguales entre sí, fenotípicamente y genotípicamente, e iguales fenotípicamente a uno de los progenitores (de genotipo dominante), independientemente de la dirección del cruzamiento. Expresado con letras mayúsculas las dominantes (**A** = amarillo) y minúsculas las recesivas (**a** = verde), se representaría como se muestra a continuación:

♀ \ ♂	A	A
a	Aa	Aa
a	Aa	Aa

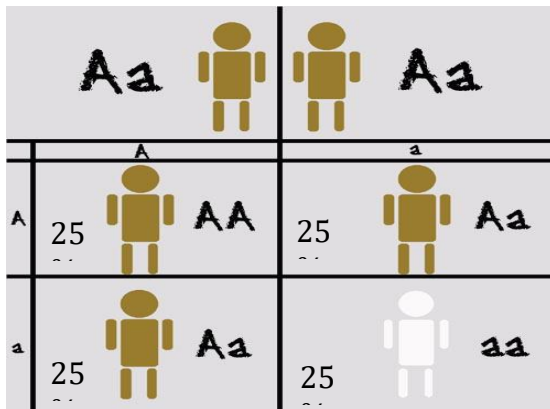
Todos los descendientes saldrían con el gen dominante.

La segunda indica que dos alelos de un mismo gen (Aa), se separan al formarse los gametos, es decir la A y la a se separan y terminan en cigotos distintos. Esto equivale a cruzar una Primera Generación Filial (Aa X Aa) con otra Primera Generación Filial (Aa X Aa), en otras palabras, cruzar individuos heterocigotos. como resultado obtuvo la Segunda Generación Filial.

EJEMPLO:

**A:** Cabello negro (Dominante)

**a:** Cabello rubio (Recesivo)



GENOTIPO	PROBABILIDAD	FENOTIPO
AA	25%	Cabello negro
Aa	50%	Cabello negro
aa	25%	Cabello rubio

El 75% saldría con el gen dominante y el 25% con el gen recesivo.

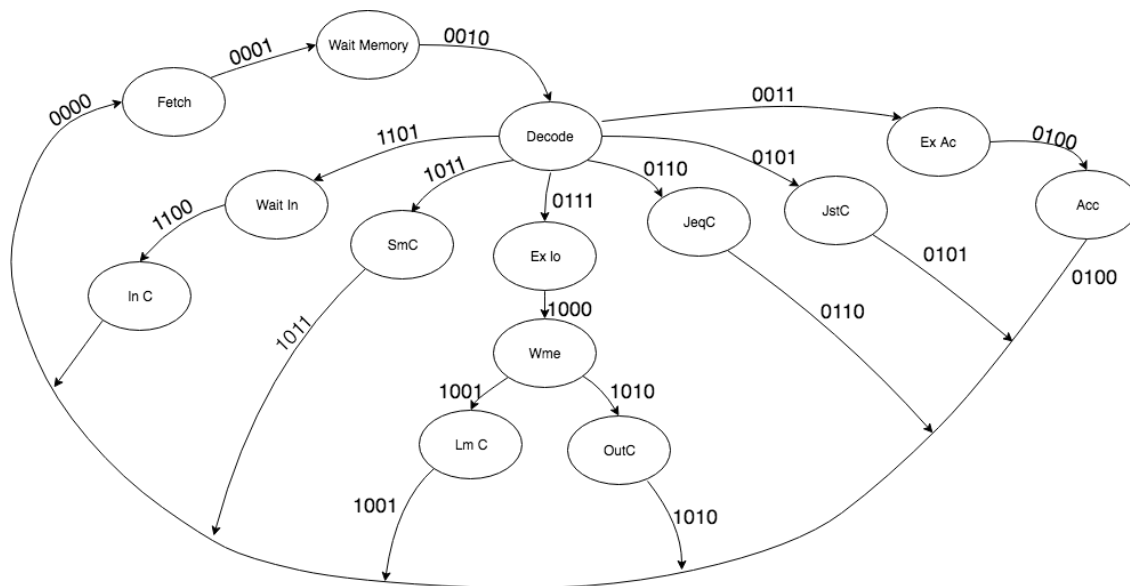
## Componentes anexados al programa:

### Maquina de estados:

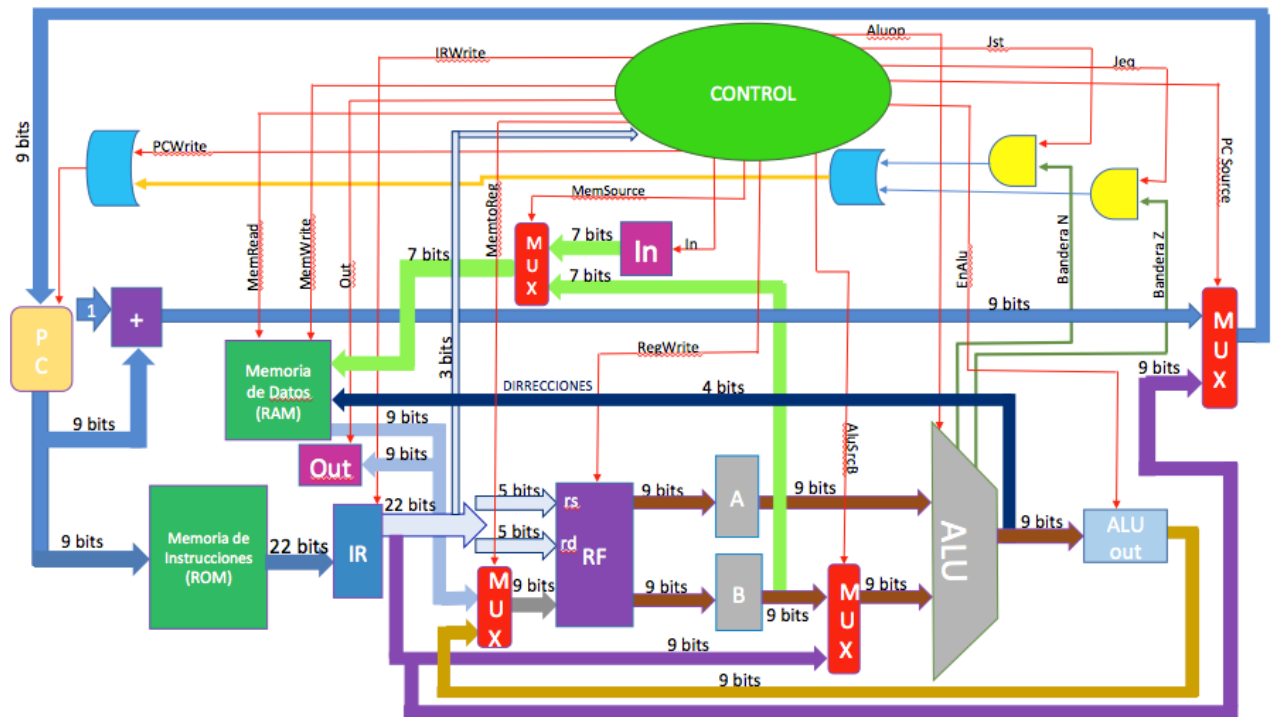
F	0000
WM	0001
DECODE	0010
EX AC	0011
ACC	0100
JSTC	0101
JEQC	0110
EX LO	0111
WME	1000
LMC	1001
OUTC	1010
SMC	1011
INC	1100

000	ADDC
010	SM
011	LM
100	JEQ
101	JST
110	IN
111	OUT

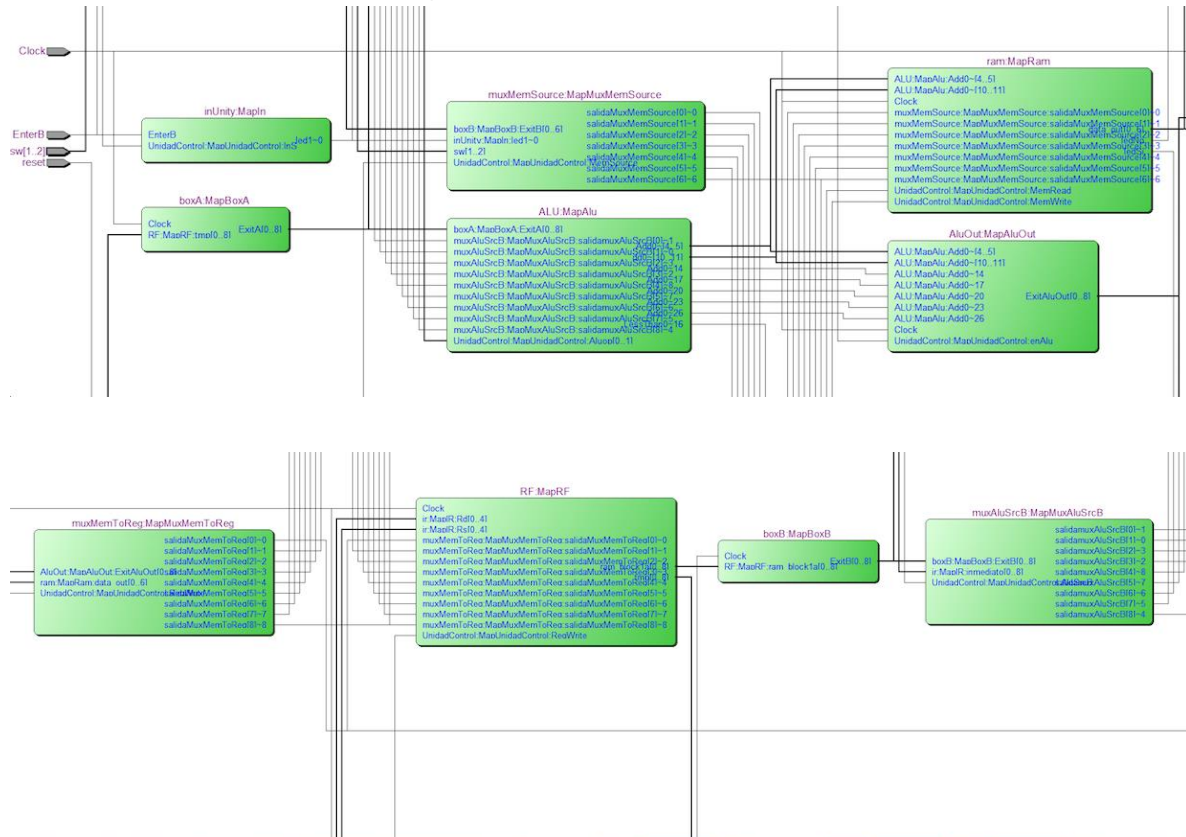
Para realizar la maquina de estados fue necesario diseñar la microarquitectura y simular el comportamiento de esta, se crearon varios estados de espera por los retardos que generaba la ROM y la RAM.



## Diagrama estructural:

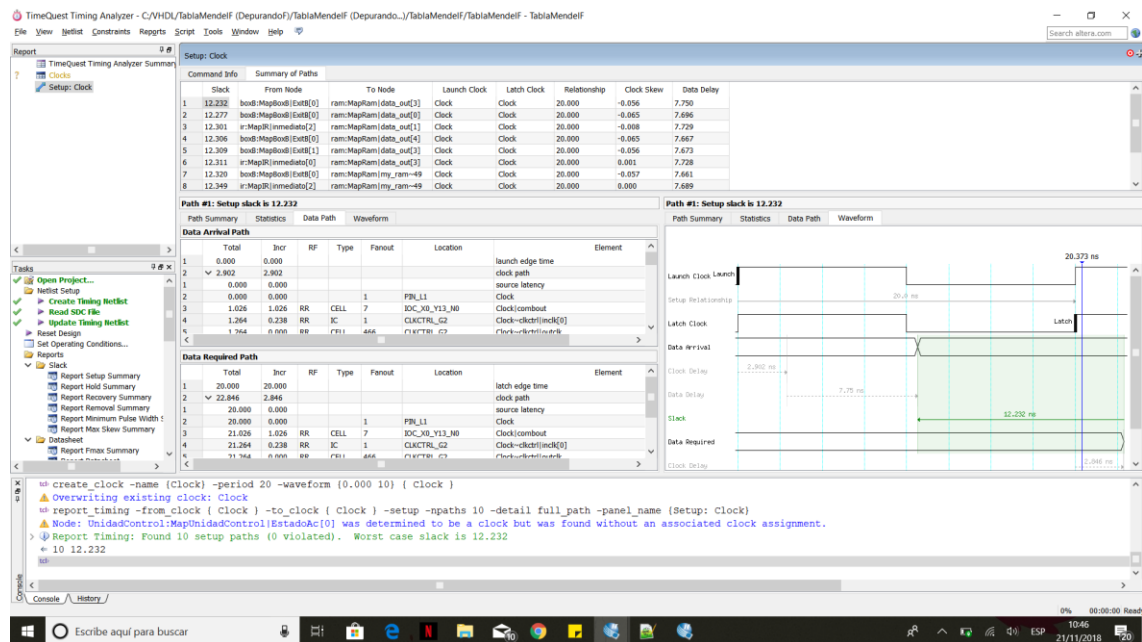


## Map viewer post mapping:





## Análisis de holgura de la arquitectura:



la ruta mas demorada en la arquitectura es la que toma desde el nodo BoxB hasta el nodo RAM con un delay de ya que esta ocupa el mayor delay que es de 7.750ns. Como la holgura que maneja nuestra arquitectura es de 20ns se puede concluir que teóricamente no alcanza a ocupar ni siquiera la mitad del periodo y se puede realizar la instrucción bajo el tiempo estipulado.

## Descripción y Formato de Instrucciones y Registros

### Instruction format:

I type instructions	Opcode	Rd	Rd	Immediate/Adress
# of bits	3	5	5	9

Data_Memory (Ram)	Data
# of bits (width)	9
# of bits (length)	16

Instruction_Memory (Rom)	Instruction
# of bits (width)	22
# of bits (length)	256

### Instruction set:

- Sm (Store memory): Carga un dato en DataMemory.
- Lm (Load memory): Carga un dato de Datamemory en un registro.
- Jeq (Jump equal): Salta si los dos registros son iguales.
- Jst (Jump smaller than): Salta si el dato que contiene un registro es más pequeño que el otro.
- Addc (Add constant): Suma la constante con el dato que contiene un registro.
- Input: Carga un inmediato ingresado por el usuario a la memoria de datos (en una parte específica que recibe los valores de los genotipos).
- Output: Manda el dato que contiene un registro a mostrarse en los displays.

### Ejemplos:

```
sm r16,(r10),000000000
lm r18,(r10),000000000
jeq r15,r17,000010101
jst r19,r21,101001011
addc r13,r14,000101000
input r10,r0,001000100
output r20,r0,100000001
```

### Register and memory format:



- La memoria de instrucciones consta de 22 bits de ancho y  $2^9$  posiciones de largo.
- Los registros posibles para usar son 32.
- El RF consta de  $2^5$  posiciones y de 9 bits de ancho.

**Diseño de interfaz:** Se implementaron 2 botones, el primer boton es enter, se utilizara para confirmar que el usuario ya ingreso en su totalidad los dato, el Segundo, es el boton de reset, el cual pone el sistema en fetch y todas las señales que envia la maquina de estados en cero.

A su vez se implementaron 2 displays 7 segmentos, en los cuales se muestran las probabilidades resultantes de los genotipos ingresados. Las cuatro probabilidades se mostraran una tras otra después de una breve espera.

Se establecieron 5 leds con multiples propisotos:

Led A(wait): identifica cuando la arquitectura se queda sobre el estado de wait memory a la espera que el usuario ingrese el respectivo dato

Led T(reset): indica que la arquitectura se lleva a estado fetch independientemente del estado en el que este

Led V(In): Se enciende al momento de que el botón de enter es presionado para guardar el dato sobre la RAM.

Led D(guardar): al momento que el led se enciende indica que los datos se guardan en la RAM.

## Conclusiones:

En el desarrollo del proyecto, pudimos identificar una serie de dificultades a superar, entre ellas:

- El diseño de la arquitectura, el cual fue replanteado varias veces con el objetivo de suplir las necesidades del proyecto y contar con una optima implementacion. Se tuvieron en cuenta las anotaciones realizadas por la profesora para los cambios que se realizaron.
- El diseño de las instrucciones, fue una parte importante del proyecto decidir cuantos tipos de instrucciones era necesario utilizar para realizar el codigo que ejecutaria la arquitectura diseñada y como estarian definidos estos tipos.
- Conectar los diversos componentes de la arquitectura, fue la parte del proyecto que mas nos tomo tiempo, puesto que nos toco analizar la ejecucion de los componentes en conjunto, con algunas instrucciones para coordinarlos y poder encontrar los errores para empezar a depurar.

- La realización del proyecto permitió adquirir conocimientos y comprender a fondo el funcionamiento de un procesador, desde entender una arquitectura, hasta diseñarla e implementarla.
- Se desarrollaron habilidades para manejar el lenguaje VHDL así como también el uso de la FPGA para la simulación física del proyecto.
- Se presentaron varios inconvenientes al sincronizar la arquitectura, ya que los componentes en muchas ocasiones no manejan los mismos tiempos y eso puede generar una malfunction de la arquitectura en general.