**成員:**

劉峻瑜、廖振東

**題目:**

K-means

**需要:**

1. **功能:**

1. 以硬體電路(soc)實現K-means分群之動作。

2. 使用者可以透過操作將樣本數進行分群(unsupervised learning)。

3. 使用者可以由外部輸入(1~5)之k值(族群數)。

4. 程式透過亂數隨機產生k值之坐標值。

5. 程式透過亂數產生至少100個樣本點，並且透過uart顯示出來。

6. 程式可以進行最短路徑判別、重心點座標運作。

7. 程式可以判別族群是否消失而隨機產生。

7. 程式可以進行動態記憶體配置存取k值。

8. 程式可以不斷運作計算直到重心點座標相同。

9. 此運作至少可以進行族群分類 K: 5個 ,標本100個。

10. 輸出介面可以透過

Case1 可以使用vga顯示

Case2 uart讀值

**2.介面(內部)需求:**

1. 使用暫存器(block ram) 進行存取樣本數與族群數。

2. 設計一個主狀態機進行整體流程運作。

3. 設計一個副狀態機進行最短路徑判斷、分群以及重心點計算。

4. 符合vga規格的輸出port。

5. 演算法運算:

Case1歐氏距離（Euclidean distance）

Case2 Dijkstra 演算法 ->判斷最短距離

6. ip設計

(1)根號運算 (2)平方運算 (3) 除法運算 (4)族群是否消失。

7. ip設計

設計一較有較佳亂數能力之ip。

**3.介面(外部)需求:**

1.USB接口

(1)鍵盤 (2)uart

2.vga 接口

**4.效能規格:**

1.可以在一定的時間內完成判斷。

**5.限制:**

1. 使用zynq-7020之開發粄。

2. 歐氏距離使用 verilog 語言。

3. Dijkstra 演算法使用VHDL語言。

4. Uart Driver 使用c語言。

5. VGA 顯示模塊 使用Verilog語言。

6. ram 在1MB之內。

驗證:

1. 撰寫一c#程式 驗證演算法是否正確計算最短路徑以及分群。

2. 撰寫一ram.v 驗證ram是否能夠正常存取 。

3. 撰寫一vga.v 驗證vga是否能夠正常顯示(verilog) 。

4. 撰寫一c程式 驗證uart 是否正常溝通。

5. 撰寫一random.v 驗證是否有較佳亂數能力。

**分析(breakdown):**

設計: