**成員:**

劉峻瑜、廖振東

**題目:**

K-means

**需要:**

1. **功能:**
2. 以硬體電路(soc)實現K-means分群之動作。
3. 使用者可以透過操作將樣本數進行分群(unsupervised learning)。
4. 使用者可以使用uart介面單元，並透過鍵盤輸入(1~5)個之k值(族群數)。
5. 程式中透過亂數產生器(random\_1)隨機產生至少100個樣本點(座標值)，並透過UART顯示出來。
6. 程式中亂數產生器(random\_1)產生之樣本點(座標值)，透過暫存器單元，將樣本點(座標值)寫至記憶體。
7. 程式中暫存器單元可以進行動態記憶體配置存取樣本點值。
8. 程式中透過亂數產生器(random\_2)隨機產生出k值個數之座標值。
9. 亂數產生器(random\_2)內包含一暫存器(BUFFER)用於進行k值座標之存取。
10. 程式中運算電路單元可以對暫存器單元進行讀取。
11. 程式中運算電路單元可以將讀取到的樣本點(座標值)、族群(座標值)進行最短距離判別，將最靠近族群的各個樣本(座標值)，納入族群，成為一新集合。
12. 程式中運算電路單元可以計算新集合之重心點座標，成為新族群之座標
13. 程式中運算電路單元可以判別族群是否消失而隨機產生一全新的族群。
14. 程式中運算電路單元可以不斷運作直到所有樣本點皆已找到歸屬之族群。
15. 此程式運作至少可以進行族群分類 K: 5個 ,標本100個。
16. 輸出介面可以透過

Case1 可以使用vga顯示

Case2 uart讀值

1. 演算法運算:

Case1歐氏距離（Euclidean distance）

Case2 Dijkstra 演算法 ->判斷最短距離

1. **介面(內部)需求:**

**軟體:**

**Uart 輸入介面**

1.tx,rx port

**硬體:**

**vga**

1. 符合vga規格(r、g、b個4bit)的輸出port。

2. 包含 水平訊號(hc port )以及垂直(vc port)訊號。

**Random\_1(樣本點)**

1. 32bit 暫存器 進行亂數運算。

2. 100個(11 bit)樣本點的位址。

**Random\_2(族群數)**

1. 32bit 暫存器 進行亂數運算。

2. rx port 接收 產生 k個 族群座標。

3. product\_k port 額外增加族群座標。

**Block ram**

1. 100個(11 bit)樣本點的位址。

2. CLKA PORT 接收資料CLKB PORT 輸出資料。

3. 2條 位址線 2條 資料線。

**Calculation 單元**

1. 100個(11 bit)樣本點的位址。

2. CLK PORT

3. 2條 位址線 3條 資料線。

加法:

除法:

平方:

**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.進行一次32bit除法在10 cycls內完成。

2.進行一次4bit加法在1 cycle內完成。

3.進行一次32bit平方在2 cycle內完成。

4.亂數樣本點產生在20 cycle內完成。

**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:

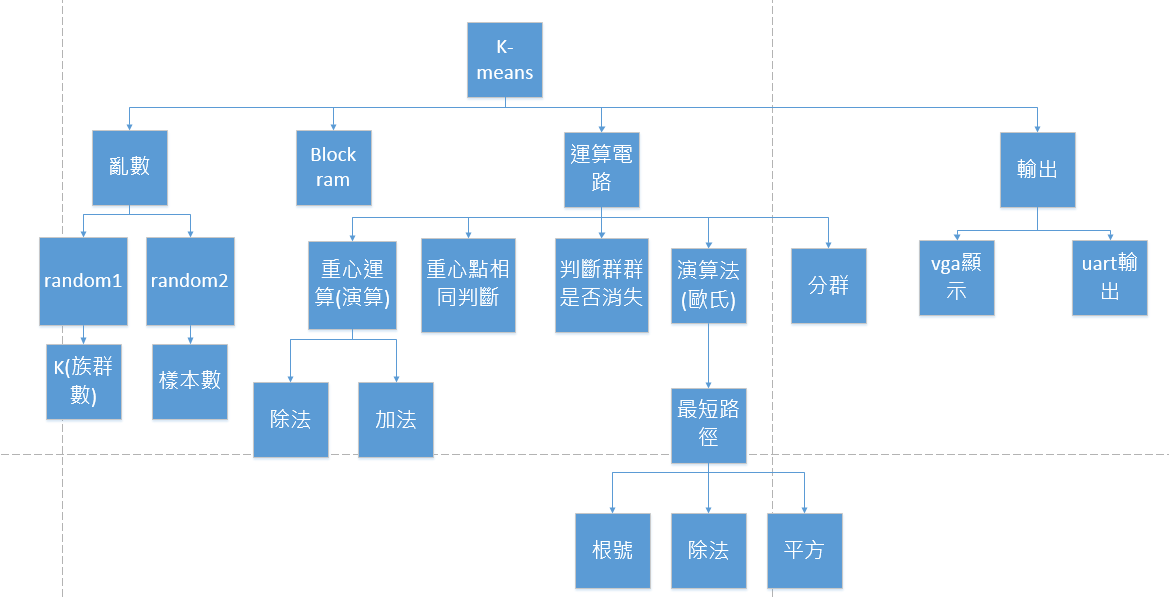
1. 撰寫一c#程式 驗證演算法是否正確計算最短距離以及分群。

2. 撰寫一ram.v 驗證ram是否能夠正常存取 。

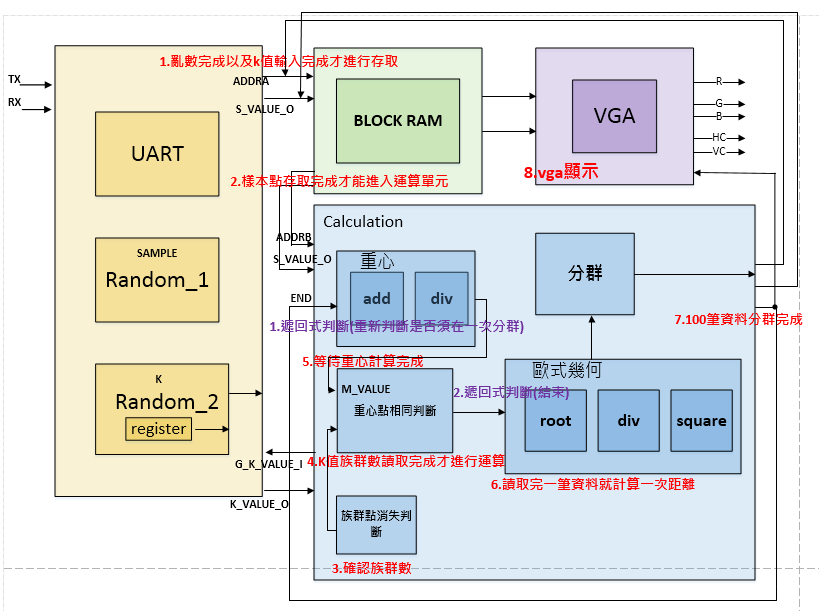
3. 撰寫一vga.v 驗證vga是否能夠正常顯示(verilog) 。

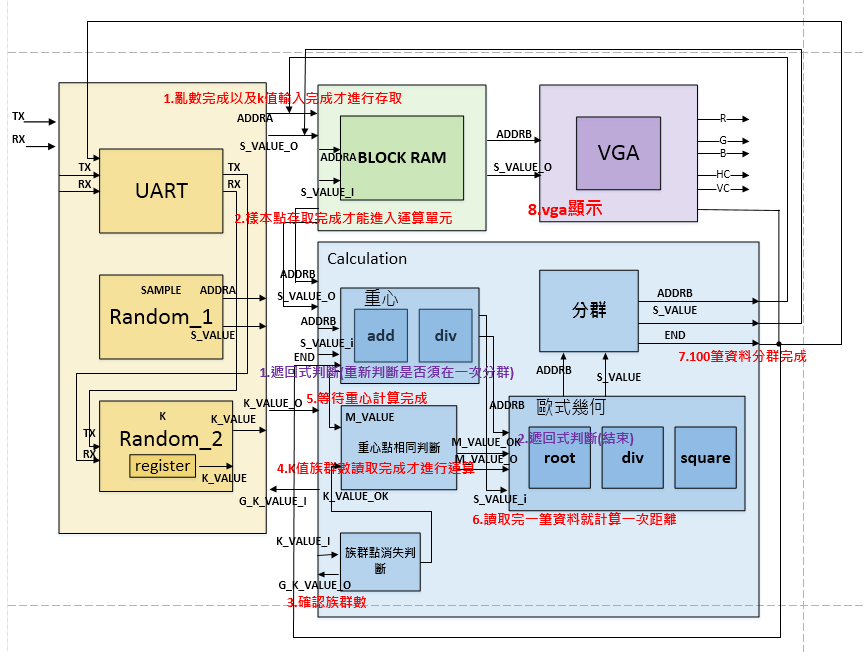
4. 撰寫一c程式 驗證uart 是否正常溝通。

5. 撰寫一random.v 驗證是否有較佳亂數能力。

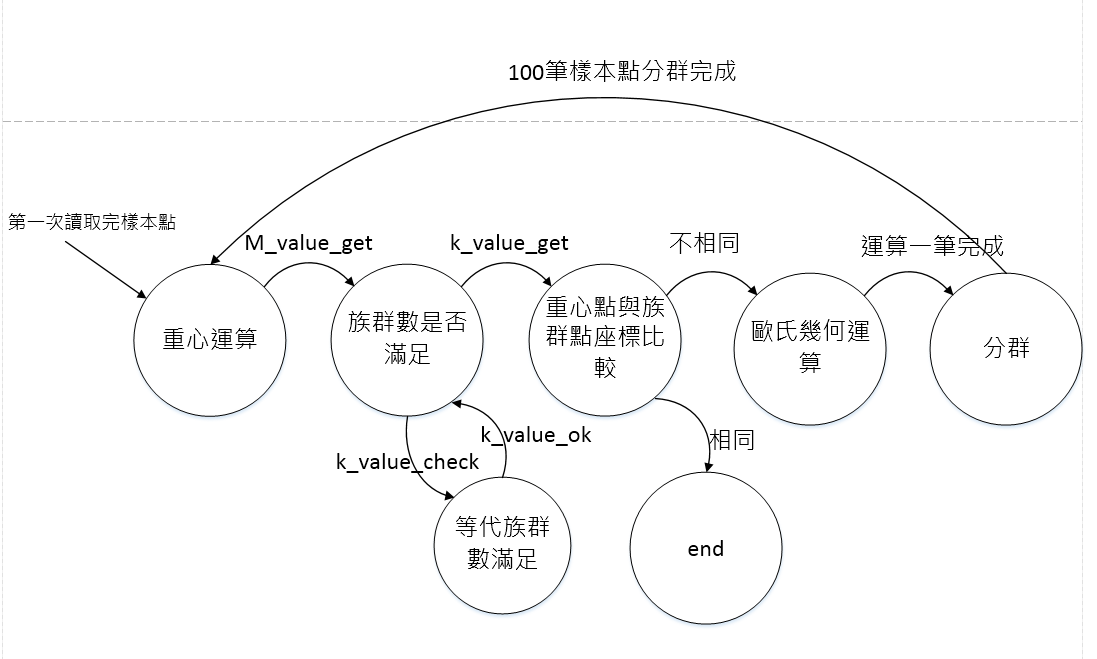
**分析(breakdown):**

**設計:**

** 架構圖:** BLOCK DIAGRAM(簡易)

**架構圖:** BLOCK DIAGRAM(詳細拉線)

**Fsm**:

運斷電路單元: