國立高雄第一科技大學

工科博士班

**系統晶片設計實習**

K-means

指導老師：陳朝烈

班 級：電子4A

學生姓名：廖振東,劉峻瑜

學 號：u0252040,u0252038

**需求:**

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(Block ram)，將樣本點(座標值)寫至記憶體。
7. 程式中暫存器單元(Block ram)可以存取100個(24bit)樣本點值(約0.3bytes)。
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(Block ram)進行讀取。
11. 程式中運算電路單元從暫存器單元(Block ram)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中的重心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中的族群點消失判斷單元可以判別族群是否消失而隨機產生一全新的母坐標。
14. 程式中運算電路單元中的重心點相同判斷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. 程式中運算電路單元中的歐氏幾何單元進行最短距離計算(母族群座標與子樣本座標距離計算)。
16. 程式中運算電路單元中的分群單元進行子樣本座標分群，並傳回暫存器單元(Block ram)存取。
17. 此程式運作至少可以進行族群分類 K: 5個 ,標本100個。
18. 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）(使用)

Case2 Dijkstra 演算法 ->判斷最短距離 (參考)

1. **介面(內部)需求:**

**軟體:**

Zynq板內部cpu 讀取 uart 接受到的k值

**硬體:**

**BlockRam**

1. 輸入位址線(ADDRA 19bit) 輸入port, 輸出位址線(ADDRB 19bit) 輸出port。

2. 輸入資料線(S\_VALUE\_I 12bit) 輸入port, 輸出資料線(S\_VALUE\_O 12bit) 輸出port。

3.W/R控制線(dina 1bit ) 輸入port。

4.讀/寫clka,clkb (1bit ) 輸入port。

**vga**

1. 符合vga規格(r、g、b各4bit)的輸出port。

2. 包含 水平訊號(hc port 1bit )以及垂直訊號(vc port 1bit)輸出port。

3. 資料線(doutb 12bit)輸入port與位址線(addrb 19bit)輸出port。

4. 3.W/R控制線(dina 1bit ) 輸出port。

5. 分群結束完成訊號線(end 1bit) 輸出port。

**Random\_1(子樣本點)**

1. 32bit 內部暫存器(reg) 進行亂數運算。

2. 位址線(addrb\_Ran1 19bit)輸出port。

3. 資料線(doutb\_Ran1 24bit)輸出port

**Random\_2(族群數)**

1. 32bit 內部暫存器(reg) 進行亂數運算。

2. rx 輸入port 接收 k值(母族群座標個數)。

3. 暫存記憶體(reg) k個 (18 bit K\_VALUE\_I) (母族群座

標) 。

4. 暫存記憶體(reg) 額外母族群座標(product\_k 24bit)。

5. 母族群座標(K\_VALUE\_I 24bit)輸出port。

**Calculation 單元**

**外部:**

1. 輸入位址線(ADDRA\_Calcu 19bit) 輸入port, 輸出位址線(ADDRB\_Calcu 19bit) 輸出port。

2. 輸入資料線(S\_VALUE\_I\_Calcu 12bit) 輸入port, 輸出資料線(S\_VALUE\_O\_Calcu 12bit) 輸出port。

3. W/R控制線(dina 1bit ) 輸出port。

4. 分群結束完成訊號線(end\_Calcu 1bit) 輸出port。

5. 母族群點座標 (G\_K\_VALUE\_I 18bit) 輸入port。

6. 取得母族群點座標訊號線(G\_K\_VALUE\_O 1bit) 輸出port。

**內部:**

重心:

重心點相同判斷:

族群點消失判斷:

分群:

歐氏幾何:

除法:

1. 2個32bit 內部暫存器(tempa、tempb)及2個64bits內部暫存器(temp\_a、temp\_b)進行運算。

2. 被除數(divide 16bits)輸入port。

3. 除數(divisor 32bits) 輸入port。

4. 商數(quotient 32bits)輸出port。

5. 餘數(residue 32bits)輸出port。

平方:

1. 28個32bit內部暫存器進行乘法運算。

2. 2個樣本輸入port(mul\_a、 mul\_b 16bits)。

3. 乘法結果(mul\_out 32bits) 輸出port。

**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) 。

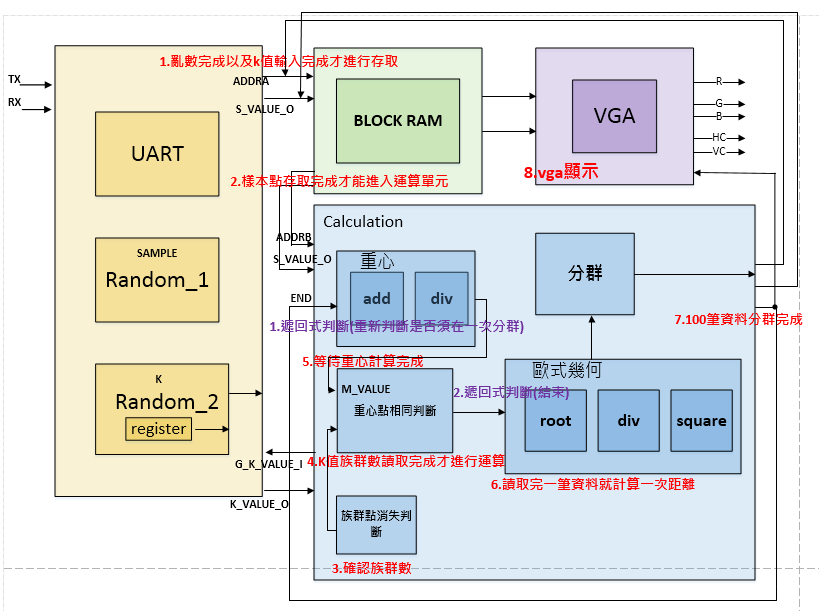
4. 撰寫一c程式 驗證uart 是否正常溝通。

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:**

**電路圖:** BLOCK DIAGRAM(簡易)

K-means

|  |  |  |  |
| --- | --- | --- | --- |
| 腳位名稱 | 腳位型態 | 資料型態 | 備註 |
| clk | Input | reg:1bit | 系統clk |
| rst | Input | reg:1bit | 重設 |
| Uart\_in | Input | reg:4bits | 使用者輸入之K值 |
| Uart\_outX | Output | reg:13bits | uart回傳分類結果X  前3bits為第幾個族群 |
| Uart\_outY | Output | reg:12bits | uart回傳分類結果Y  前3bits為第幾個族群 |
| VGA\_R | Output | reg:4bits | VGA輸出結果 |
| VGA\_G | Output | reg:4bits | VGA輸出結果 |
| VGA\_B | Output | reg:4bits | VGA輸出結果 |
| VGA\_HC | Output | reg:1bits | VGA輸出結果 |
| VGA\_VC | Output | reg:1bits | VGA輸出結果 |

UART

|  |  |  |  |
| --- | --- | --- | --- |
| 腳位名稱 | 腳位型態 | 資料型態 | 備註 |
| TX | Input |  | 外部介面透過UART進來 |
| RX | Output |  | 內部介面透過UART出去 |
| register | Buffer | reg:8bit | 暫存資料 |
| Interrupt\_TX | parameter | reg:1bit | 從UART進來的資料已經接收完畢 |
| Interrupt\_RX | parameter | reg:1bit | 透過UART丟出的資料已經準備完畢 |
| IRQ\_Tx(); //Tx中斷  IRQ\_Rx(); //Rx中斷  dataIn(); //將資料收進來  dataOut(); //將資料丟出去 | | | |

Random\_1

|  |  |  |  |
| --- | --- | --- | --- |
| 腳位名稱 | 腳位型態 | 資料型態 | 備註 |
| Random1\_clk | Input | reg:1bit | 系統clk |
| Random1\_rst | Input | reg:1bit | 重設 |
| ToBlockRam\_clka | Output | wire | 餵給Blocl\_Ram的clk |
| SampleX | Output | reg:12bits | 隨機樣本點X座標給Calculation  前3bits為第幾個族群  (此為測試port) |
| SampleY | Output | reg:12bits | 隨機樣本點Y座標給Calculation  前3bits為第幾個族群  (此為測試port) |
| ADDRA | Output | reg:19bits | 寫入Block\_Ram的位址線 |
| S\_VALUE\_I | Output | reg:12bits | 寫入Block\_Ram的資料線 |
| dina | Output | reg:1bits | 寫入Block\_Ram的控制線 |
| ram1 | Buffer | reg:32bits | 亂數占存器 |
| divclkcnt | Buffer | reg:32bits | 除頻占存器 |
| idle | parameter | reg:3bit | FSM狀態—初始 |
| beginRandom | parameter | reg:3bit | FSM狀態—產生亂數 |
| ifRepeat | parameter | reg:3bit | FSM狀態—判斷樣本是否重複 |
| savetoBuf | parameter | reg:3bit | FSM狀態—將亂數存入Buf |
| toBlockRam | parameter | reg:3bit | FSM狀態—將亂數丟入BlockRam |
| waitState | parameter | reg:3bit | FSM狀態—等待 |
| End | parameter | reg:3bit | FSM狀態—結束 |
| divclk1() //除頻  random 1() //取得亂數 | | | |

Random\_2

|  |  |  |  |
| --- | --- | --- | --- |
| 腳位名稱 | 腳位型態 | 資料型態 | 備註 |
| Random2\_clk | Input | reg:1bit | 系統clk |
| Random2\_rst | Input | reg:1bit | 重設 |
| Group\_quanI | Input | reg:3bits | 族群數量輸入 |
| Group\_quanO | Output | reg:3bits | 族群數量輸出給Calculation |
| Group\_coorX | Output | reg:9bits | 族群座標X給Calculation |
| Group\_coorY | Output | reg:9bits | 族群座標Y給Calculation |
| ram2 | Buffer | reg:32bits | 亂數占存器 |
| divclkcnt | Buffer | reg:32bits | 除頻占存器 |
| idle | parameter | reg: 3bit | FSM狀態--初始 |
| beginRandom | parameter | reg: 3bit | FSM狀態—產生亂數 |
| waitState | parameter | reg: 3bit | FSM狀態—等待 |
| ifRepeat | parameter | reg:3bit | FSM狀態—判斷族群是否重複 |
| savetoBuf | parameter | reg:3bit | FSM狀態—將亂數存入Buf |
| End | parameter | reg: 3bit | FSM狀態—結束 |
| divclk2() //除頻  random 2() //取得亂數  userkeyin() //讀取使用者輸入族群的數量 | | | |

Block\_Ram

|  |  |  |  |
| --- | --- | --- | --- |
| 腳位名稱 | 腳位型態 | 資料型態 | 備註 |
| Block\_Ram\_clka | Input | reg:1bit | Block\_Ram的clk |
| Block\_Ram\_clka | Input | reg:1bit | Block\_Ram的clk |
| ADDRA | Input | reg:19bit | 輸入位址線 |
| ADDRB | Output | reg:19bit | 輸出位址線 |
| S\_VALUE\_I | Input | reg:12bit | 輸入資料線 |
| S\_VALUE\_O | Output | reg:12bit | 輸出資料線 |
| dina | Input | reg:1bit | W/R控制線 |
| WriteData(); //寫入資料  ReadData(); //讀取資料 | | | |

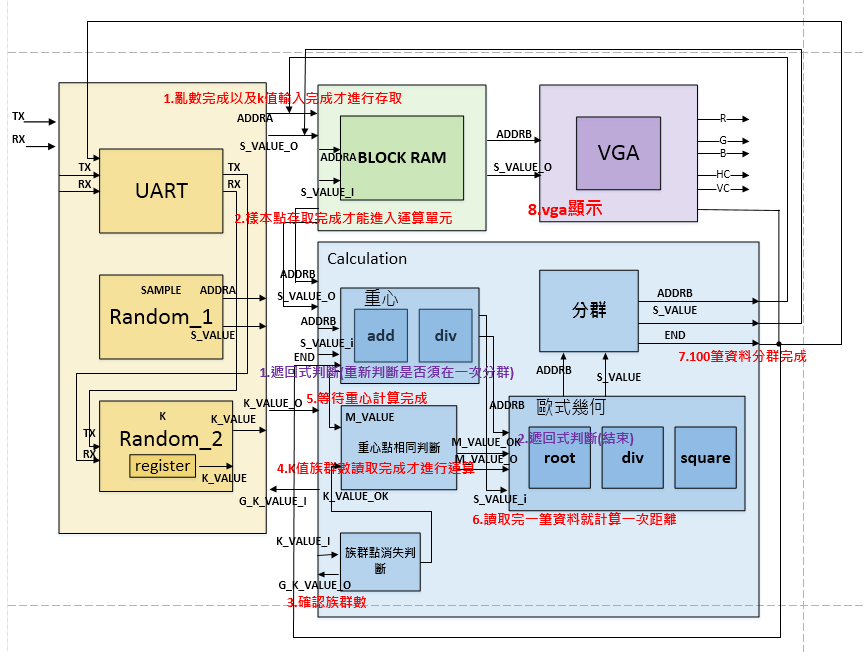
VGA

|  |  |  |  |
| --- | --- | --- | --- |
| 腳位名稱 | 腳位型態 | 資料型態 | 備註 |
| VGA\_clk | Input | reg:1bit | 系統clk |
| VGA\_rst | Input | reg:1bit | 重設 |
| Group\_coorX | Input | reg:9bits | Random\_2給的族群座標X |
| Group\_coorY | Input | reg:9bits | Random\_2給的族群座標Y |
| SampleX | Buffer | reg:12bits | 暫存樣本X |
| SampleY | Buffer | reg:12bits | 暫存樣本Y |
| ResultFlag | Input | reg:1bit | 是否分群完畢 |
| ToBlockRam\_clkb | Output | wire | 餵給Blocl\_Ram的clk |
| ADDRB | Output | reg:19bits | 讀取Block\_Ram的位址線 |
| S\_VALUE\_O | Input | reg:12bits | 讀取Block\_Ram的資料線 |
| dina | Output | reg:1bits | 讀取Block\_Ram的控制線 |
| VGA\_R | Output | reg:4bits | VGA輸出結果 |
| VGA\_G | Output | reg:4bits | VGA輸出結果 |
| VGA\_B | Output | reg:4bits | VGA輸出結果 |
| VGA\_HC | Output | reg:1bits | VGA水平訊號 |
| VGA\_VC | Output | reg:1bits | VGA垂直訊號 |
| idle | parameter | reg:2bit | FSM狀態—初始 |
| display | parameter | reg:2bit | FSM狀態—顯示 |
| End | parameter | reg:2bit | FSM狀態—結束 |
| getData(); //從Block\_Ram獲取分群結果  analysis(); //分析讀取到的結果  display (); //顯示分群結果 | | | |

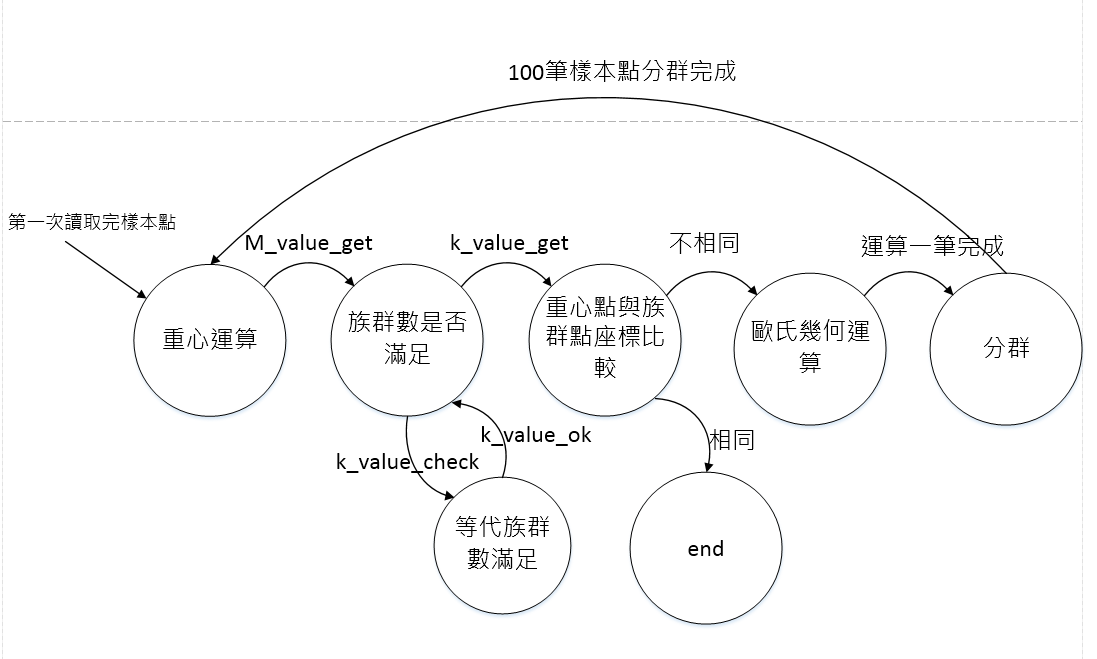
Calculation

|  |  |  |  |
| --- | --- | --- | --- |
| 腳位名稱 | 腳位型態 | 資料型態 | 備註 |
| Calculation \_clk | Input | reg:1bit | 系統clk |
| Calculation \_rst | Input | reg:1bit | 重設 |
| Group\_quanI | Input | reg:3bits | Random\_2給的族群數量(k值) |
| Group\_coorX | Input | reg:9bits | Random\_2給的族群座標X |
| Group\_coorY | Input | reg:9bits | Random\_2給的族群座標Y |
| ResultFlag | Output | reg:1bit | 告知Block\_Ram及VGA是否分群完畢 |
| ToBlockRam\_clkb | Output | wire | 餵給Blocl\_Ram的clk |
| ADDRA | Output | reg:19bits | 寫入Block\_Ram的位址線 |
| ADDRB | Output | reg:19bits | 讀取Block\_Ram的位址線 |
| S\_VALUE\_I | Output | reg:12bits | 寫入Block\_Ram的資料線 |
| S\_VALUE\_O | Input | reg:12bits | 讀取Block\_Ram的資料線 |
| dina | Output | reg:1bits | 寫入/讀取Block\_Ram的控制線 |
| SampleX | Buffer | reg:12bits | 暫存樣本X |
| SampleY | Buffer | reg:12bits | 暫存樣本Y |
| ResultX | Output | reg:13bits | 計算完的結果X給Block\_Ram及uart  前3bits為第幾個族群 |
| ResultY | Output | reg:12bits | 計算完的結果Y給Block\_Ram及uart  前3bits為第幾個族群 |
| SampleCnt | parameter | reg:7bit | 樣本數量計數器 |
| idle | parameter | reg:4bit | FSM狀態—初始 |
| getData | parameter | reg:4bit | FSM狀態—從BlockRam獲取樣本 |
| waitinput | parameter | reg:4bit | FSM狀態—等待使用者輸入族群(k) |
| Center | parameter | reg:4bit | FSM狀態—求重心 |
| Distance | parameter | reg:4bit | FSM狀態—求重心與各樣本的距離 |
| Grouped | parameter | reg:4bit | FSM狀態—分群 |
| Group\_disappear | parameter | reg:4bit | FSM狀態—判斷群是否消失 |
| newGroup | parameter | reg:4bit | FSM狀態—產生新的群 |
| pushData | parameter | reg:4bit | FSM狀態—將分群結果存入BlockRam  以及告知VGA以分群完畢 |
| End | parameter | reg:4bit | FSM狀態—結束 |
| pushData(); //寫入結果至Block\_Ram  getData(); //從Block\_Ram獲取樣本  Center(); //求重心  add(); //加法  div(); //除法  M\_value(); //判斷重心是否一致  Group\_disappear(); //判斷族群是否消失  newGroup(); //若族群消失，從樣本內隨機產生新的族群  Distance(); //歐氏幾何  add(); //加法  square(); //平方  div(); //除法  root(); //開根號  Grouped(); //分群 | | | |

**電路圖:** BLOCK DIAGRAM(詳細拉線)

****

**運算電路單元\_流程圖**:

****

Random\_1 FSM



Random\_2 FSM



VGA FSM



Calculation FSM



coding ----- Random\_1

驗證