

組合電路設計 (Spuggy0919)

最近在學習 Arduino，有些電路是以前數位邏輯的知識，組合電路是數位電路設計的基礎，學好組合電路數位電路就學好一大半了，組合電路可用於解碼，定址，PLA 控制，甚至 CPU 控制單元都用的到，20 年前因工作的需要，爲了設計一款客制 CPU，寫了一個真假表轉換成化簡成式子的程式，找了許久找不回來，一時想起便再寫一次，已經忘了當初如何寫的只約略記得資料結構用 00，01，11 分別表示輸出的 True，False，及 Don't Care 三種狀態，一個 Long 用 32bits，兩個 bit 代表一個輸出，共可處理 16 個輸出，同時將真假表放入一個陣列，限定 1K，只有 10bits 輸入，這樣輸入檔案格式便定義好了。

例一輸入檔案 7seg.txt

%%說明 7 segment 共陽極 True Table

%7 segment common +5V True Table

%000 gfedcba

0000 0000001

0001 1001111

0010 0010010

0011 0000110

0100 1001100

0101 0100100

0110 0100000

0111 0001111

1000 0000000

1001 0000100

程式流程，對每個輸出求下列化簡式子 EQ

1. 輸入項數 k 從 1 到 m\

1.1 將從 m 輸入中取 k (組合 $C(m, k)$) 的每一項，做含蓋檢查(Minimal K-map)

1.1.1 Check 該**輸入組合項**，所含蓋輸出爲真的有幾個(OneCnt)例如 pq' 爲 k=2 的其中一個**輸入組合項**

1.1.2 若 pq' 所含蓋皆爲 1 或 X 表示該項是屬於**化簡項**，加入**式子集合** pq' in EQ

1.1.3 並將 TrueTable 輸出爲 1 且被此子項所含蓋(共 OneCnt 個)的全設爲 X (Don't Care)

1.1.4 Truetable 是否 true 項都被含蓋(轉爲 Don'tCare) ，若是化簡結束

2. 輸出化簡式子 EQ

程式執行

Minimize 7seg.txt 7sego.txt 1<enter>

輸出檔案 7sego.txt

%Minimize ver 1.0 spuggy0919@gmail.com

%7 segment True Table

% Input Symbols:3210

% :pqrs

%Output Symbols:0123456

% :ABCDEFG

A=(r) || (p) || (q&&s) || (!q&&!s)

B=(p) || (!r&&!s) || (q&&!s) || (q&&!r)

C=(!q) || (r&&s) || (!r&&!s)

D=(p) || (r&&!s) || (q&&!s) || (!q&&r) || (q&&!r)

E=(r&&!s) || (!q&&!s)

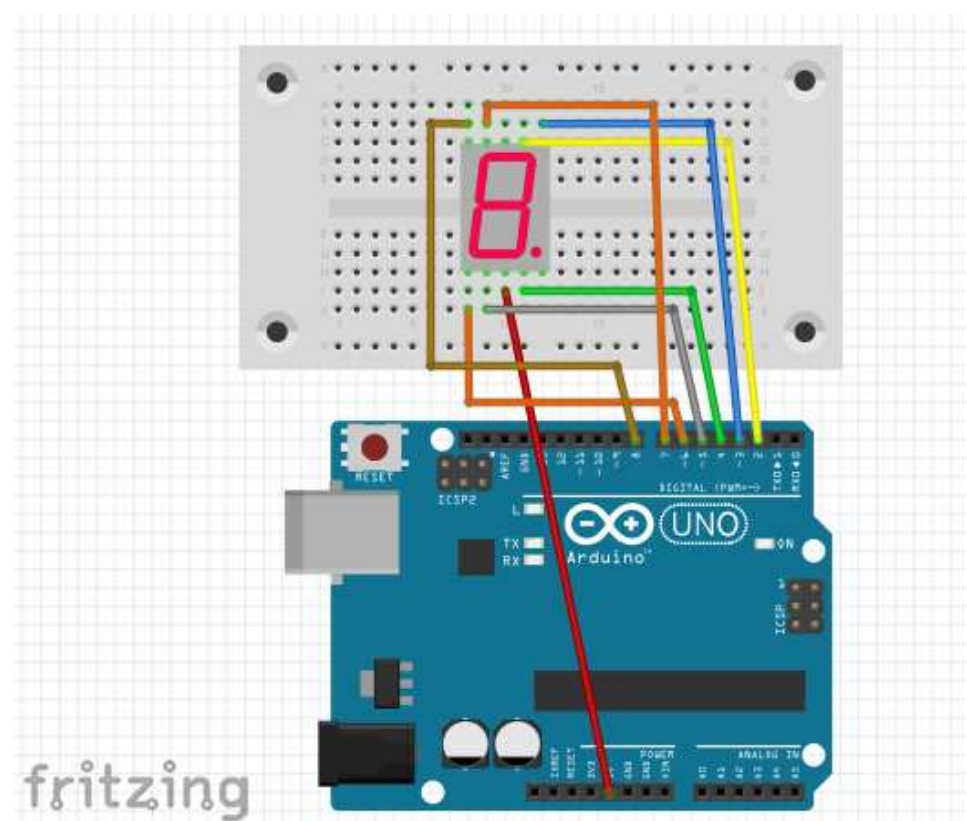
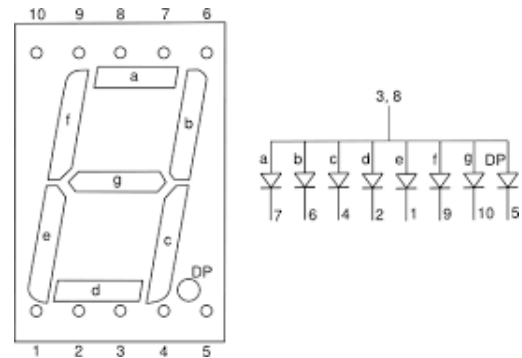
F=(s) || (!r) || (q)

G=(p) || (r&&!s) || (!q&&!s) || (!q&&r) || (q&&!r&&s)

七段顯示的解碼器

將 Pin 2, 3, 4, 5, 6, 7, 8 分別對應到七段顯示的 a, b, c, d, e, f, g

5v 接 Comon Pin



Sketch 程式可以用 **Blink** 範例修改便可完成此解碼器

```
/*
  7 Segment
  Turns on an Digit on for one second , then next for one second ,
  repeatedly.

  This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led[7] = {2, 3, 4, 5, 6, 7, 8}; //abcdefg

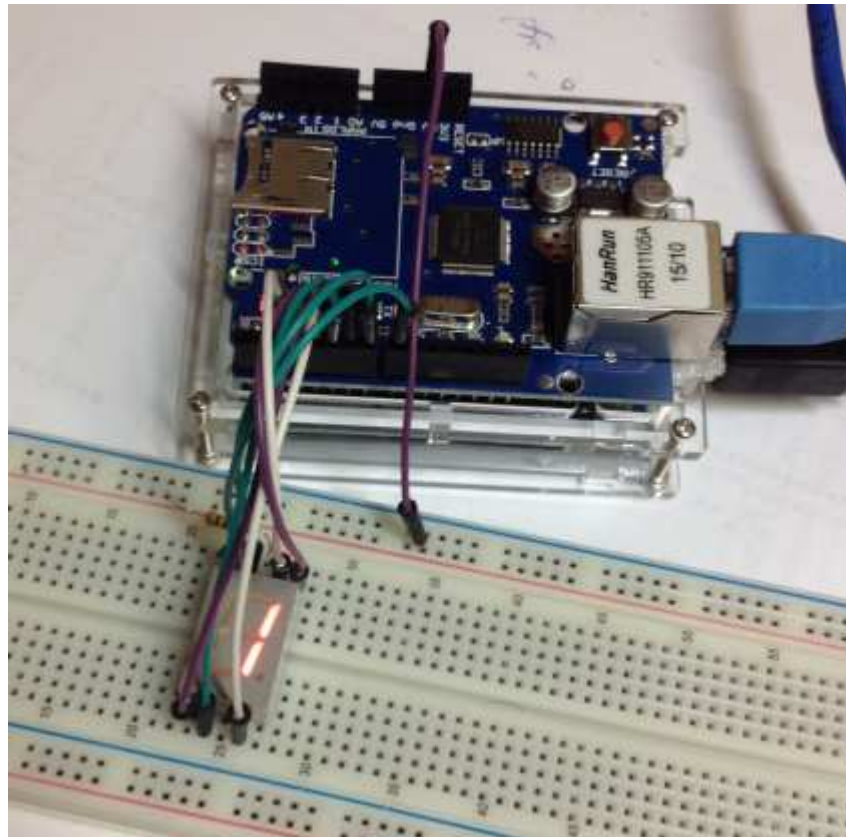
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  Serial.begin(9600);
  int i;
  for(i=0;i<7;i++){
    pinMode(led[i], OUTPUT);
    digitalWrite(led[i], HIGH);
  }
}

// the loop routine runs over and over again forever:
void loop() {
  int i, j, digit;
  boolean p, q, r, s;
  int out[7];
  for(digit=0;digit<10;digit++) {
    // digit=8;
    p=(digit&0b1000) ? true: false;
    q=(digit&0b0100) ? true: false;
    r=(digit&0b0010) ? true: false;
    s=(digit&0b0001) ? true: false;
    if (p) Serial.print("1"); else Serial.print("0");
    if (q) Serial.print("1"); else Serial.print("0");
    if (r) Serial.print("1"); else Serial.print("0");
    if (s) Serial.print("1"); else Serial.print("0");
```

```

Serial.print(" ");
out[0]=((q&&!r&&!s) || (!p&&!q&&!r&&s)) ? HIGH:LOW; //A
out[1]=((q&& !r&&s)|| (q&&r&& !s)) ? HIGH:LOW; //B
out[2]=(!q && r&& !s)? HIGH:LOW; //C
out[3]=((q&& r&&s)|| (q&&!r&&!s)|| (!p&&!q&&!r&&s)) ?
HIGH:LOW; //d
out[4]=(s|| (q&& !r))? HIGH:LOW; //E
out[5]=((r&&s)|| (!q&&r)|| (!p&&!q&&s)) ? HIGH:LOW; //F
out[6]=((q&& r&& s)|| (!p&& !q&& !r))? HIGH:LOW; //G
for(j=0;j<7;j++) {
    if (out[j] == HIGH) Serial.print("1");
    else Serial.print("0");
    digitalWrite(led[j], out[j]);
}
delay(1000);          // wait for a second
Serial.print("\n");
}
}

```

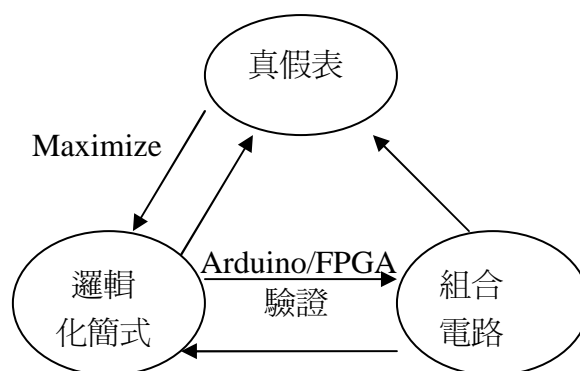


結論

組合邏輯電路每一個輸出是一個邏輯運算式，此化簡式子可等效轉換成一個邏輯電路由 NOT、AND、OR 三種 Gates 組成的線路，也可以由邏輯化簡式反推真假表，邏輯電路也可以推回真假表，但真假表無法直觀轉換成組合電路，必須經過化簡程序才能節省 Gate 數目，並讓電路精簡。

經過此設計，化簡後的式子轉成 C 在 Arduino 的模擬中驗證成果正確，我們可以将 Minimize 當成設計的一環，將真假表轉成化簡邏輯式子，再將此電路用 arduino 或 FPGA 做成組合電路進行驗證。

化簡是組合電路的重要環節，在此借助筆者寫的 Minimize 將之串起來。



還沒完還有後篇

後篇 (同步、非同步)

例二: bcd 計數回圈組合化簡

輸入 bcdcount.txt

%bcd count loop True Table

0000 0001

0001 0010

0010 0011

0011 0100

0100 0101

0101 0110

0110 0111

0111 1000

1000 1001

1001 0000

1010 XXXX

1011 XXXX

1100 XXXX

1101 XXXX

1110 XXXX

1111 XXXX

執行

Minimize.exe bcdcount.txt bcdcounto.txt 1 <enter>

產生 C 格式輸出

Minimize ver 1.0 spuggy0919@gmail.com

%bcd count loop True Table

% Input Symbols:3210

% :pqrs

%Output Symbols:0123

% :ABCD

A=(p&&!s)||(q&&r&&s)

B=(q&&!s)||(q&&!r)||(!q&&r&&s)

C=(r&&!s)||(q&&!r&&s)||(!p&&!r&&s)

D=(!s)

程式片段，可以將之前的 For loop 換掉，將前面四式 ABCD，改成 pqrs，不要忘記加；pqrs 的宣告轉為 static。沒有 for 也能依序顯示 0-9 神奇吧！這也是序向電路的雛型，將輸出接回輸入。

```
// the loop routine runs over and over again forever:
static boolean p,q,r,s;
void loop() {
  int i,j;
  int out[7];
  p=(p&&!s)|| (q&&r&&s);
  q=(q&&!s)|| (q&&!r)|| (!q&&r&&s);
  r=(r&&!s)|| (q&&!r&&s)|| (!p&&!r&&s);
  s=(!s);
  if (p) Serial.print("1"); else Serial.print("0");
  if (q) Serial.print("1"); else Serial.print("0");
  if (r) Serial.print("1"); else Serial.print("0");
  if (s) Serial.print("1"); else Serial.print("0");
  Serial.print(" ");
  out[0]=((q&&!r&&!s) || (!p&&!q&&!r&&s)) ? HIGH:LOW; //A
  out[1]=((q&&!r&&s)|| (q&&r&&!s)) ? HIGH:LOW; //B
  out[2]=(!q && r&& !s)? HIGH:LOW; //C
  out[3]=((q&&r&&s)|| (q&&!r&&!s)|| (!p&&!q&&!r&&s)) ?
HIGH:LOW; //d
  out[4]=(s|| (q&&!r)) ? HIGH:LOW; //E
  out[5]=((r&&s)|| (!q&&r)|| (!p&&!q&&s)) ? HIGH:LOW; //F
  out[6]=((q&&r&&s)|| (!p&&!q&&!r)) ? HIGH:LOW; //G
  for(j=0;j<7;j++) {
    if (out[j] == HIGH) Serial.print("1");
    else Serial.print("0");
    digitalWrite(led[j], out[j]);
  }
  delay(1000); // wait for a second
  Serial.print("\n");
}
```

這樣還有個 Bug 因為 p 被變更時影響到了 r，所以要將 pqrs 做同步變更這叫做同步(Synchronize)電路，這個叫非同步(Asynchonize)，因為 r 被立即變更的 p 影響結果數字顯示成 234567892... 01 不見了。下列改為同步方式就正確了。

```
static    boolean p=false,q=false,r=false,s=false;
void loop() {
    int i,j;
    int out[7];
    boolean pp,qq,rr,ss;
    pp=(p&&!s)||(q&&r&&s);
    qq=(q&&!s)||(q&&!r)||(!q&&r&&s);
    rr=(r&&!s)||(q&&!r&&s)||(!p&&!r&&s);
    ss=(!s);
    p=pp; q=qq; r=rr; s=ss;
    if (p) Serial.print("1"); else Serial.print("0");
    if (q) Serial.print("1"); else Serial.print("0");
    if (r) Serial.print("1"); else Serial.print("0");
    if (s) Serial.print("1"); else Serial.print("0");
    Serial.print(" ");
    out[0]=((q&&!r&&!s) || (!p&&!q&&!r&&s)) ? HIGH:LOW; //A
    out[1]=((q&&!r&&s)||(!q&&r&&!s)) ? HIGH:LOW; //B
    out[2]=(!q && r&& !s)? HIGH:LOW; //C
    out[3]=((q&&r&&s)||(!q&&!r&&!s)||(!p&&!q&&r&&s)) ? HIGH:LOW; //d
    out[4]=(s||(!q&&r)) ? HIGH:LOW; //E
    out[5]=((r&&s)||(!q&&r)||(!p&&!q&&s)) ? HIGH:LOW; //F
    out[6]=((q&&r&&s)||(!p&&!q&&!r)) ? HIGH:LOW; //G
    for(j=0;j<7;j++) {
        if (out[j] == HIGH) Serial.print("1");
        else Serial.print("0");
        digitalWrite(led[j], out[j]);
    }
    delay(1000);                // wait for a second
    Serial.print("\n");
}
```


例三: binary OP 組合化簡

輸入 in.txt

%binary gate True Table

00 0101010101010101

01 0011001100110011

10 0000111100001111

11 0000000011111111

化簡結果 out.txt

%binary gate True Table

%00 0101010101010101

%01 0011001100110011

%10 0000111100001111

%11 0000000011111111

%

% Input Symbols:10

% :pq

%Output Symbols:5432109876543210

% :ABCDEFGHJKLMNPO

A=0

B=p'q'

C=p'q

D=p'

E=pq'

F=q'

G=p'q+pq'

H=q'+p'

I=pq

J=pq+p'q'

K=q

L=q+p'

M=p

N=q'+p

O=q+p

P=1