บทที่ 4

การออกแบบและพัฒนาโปรแกรมรวมภาพและเครื่องควบคุมแว่นผลึกเหลว

4.1 การออกแบบและพัฒนาโปรแกรมรวมภาพ

คุณสมบัติของโปรแกรม

4.1.1 การรวมภาพแบบแผนที่บิท (Bitmap)

การรวมภาพในไฟล์แบบแผนที่บิท ในที่นี้หมายถึง การอ่านค่าจุดในแนวนอนของภาพที่หนึ่ง โดยอ่านเท่ากับขนาดความกว้างของภาพ ซึ่งเป็นเส้นคี่ และค่าจุดในแนวนอนของภาพที่สอง โดยอ่านเท่ากับขนาดความกว้างของภาพโดยอ่านเส้นคู่ แล้วนำจุดเส้นคี่และเส้นคู่นั้นมาประกอบกันเป็นภาพที่สาม จะทำการอ่านเช่นนี้จนหมดเส้นคี่ของภาพที่หนึ่ง และหมดเส้นคู่ของภาพที่สอง จะได้ภาพที่สามเป็นภาพใหม่ ซึ่งมีจำนวนเส้นในแนวนอนเท่ากับภาพที่หนึ่งและภาพที่สอง

4.1.2 โปรแกรมรวมภาพแบบแผนที่บิท

ความต้องการของโปรแกรม

- ไฟล์ฟอร์แมตแบบแผนที่บิต
- จำนวนสีของภาพ 16 สี ถึง 16 ล้านสี
- สีของภาพต้องเป็นตารางสีเดียวกัน
- ขนาดของภาพที่หนึ่งและภาพที่สอง กว้างคูณยาว เป็นจุดต้องเท่ากัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.1.3 การทำงานของโปรแกรม

จะอ่านไฟล์แผนที่บิตภาพที่หนึ่งและภาพที่สอง มาตรวจดูว่าเป็นไฟล์ฟอร์แมตของแผนที่บิทหรือไม่ ถ้าฟอร์แมตถูกต้องจะทำการอ่านข้อมูลส่วนต่าง ๆ และตารางสีของภาพที่ 1 มาเก็บไว้เป็นไฟล์ฟอร์แมตของเอาท์พุทไฟล์ จากนั้นจะนำเอาข้อมูลภาพของทั้งสองไฟล์มารวมกันแบบแทรกสอด ซึ่งในโปรแกรมนี้จะดำเนินการรวมภาพทั้งหมดในเมมโมรี่ จึงทำให้เวลาที่ใช้ในการรวมภาพรวดเร็วขึ้น

4.1.4 ซอร์สโปรแกรม (source program)

```
/*********************************************************/
/* Merge 2 Bitmap Program...                             */
/*   Now work if.....                                    */
/*         1. same offset                                */
/*         2. same resolution                            */
/*         3. even width (easy to divide)                */
/*********************************************************/
#include <stdio.h>

#define  ARGUMENT_NO  4

main(int argc, char *argv[])
{
 FILE  *file1, *file2, *out_file;
 long start_offset, file_size[2], pic_size[2];
 long pic_width, pic_height[2];
 long i, j;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
if(argc != ARGUMENT_NO){
      printf("\n Usage:   merge  file1  file2  merge_file\n");
      return(1);
    }
    if ((file1 = fopen(argv[1], "rb")) == NULL){
      fprintf(stderr, "Cannot open input file.\n");
      return 1;
    }
    if ((file2 = fopen(argv[2], "rb")) == NULL){
      fprintf(stderr, "Cannot open input file.\n");
      return 1;
    }
    if ((out_file = fopen(argv[3], "wb")) == NULL){
      fprintf(stderr, "Cannot open output file.\n");
      return 1;
    }
//   read offset (assume that they have the same offsets...)
  fseek(file1, 10L, SEEK_SET);
  fread(&start_offset, sizeof(long), 1, file1);

//   read file_size  (assume that they have the same sizes...)
  fseek(file1, 2L, SEEK_SET);
  fread(&file_size[0], sizeof(long), 1, file1);

//   compute image size
  pic_size[0] = file_size[0] - start_offset;
  pic_size[1] = 2*pic_size[0];
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
file_size[1] = start_offset + pic_size[1];

 //    read picture height.
fseek(file1, 22L, SEEK_SET);
    fread(&pic_height[0], sizeof(long), 1, file1);
    pic_height[1] = 2 * pic_height[0];
 //   compute picture width.
    pic_width = pic_size[0] / pic_height[0];

    rewind(file1);
    rewind(out_file);
    for(i = 0; i < start_offset; i++){          // copy header
      fputc(fgetc(file1), out_file);
    }

    fseek(out_file, 2L, SEEK_SET);                  // change size
    fwrite(&file_size[1], sizeof(long), 1, out_file);
    fseek(out_file, 22L, SEEK_SET);
    fwrite(&pic_height[1], sizeof(long), 1, out_file);
    fseek(out_file, 34L, SEEK_SET);
    fwrite(&pic_size[1], sizeof(long), 1, out_file);

    fseek(out_file, start_offset, SEEK_SET);      // point to start pic
    fseek(file1, start_offset, SEEK_SET);
    fseek(file2, start_offset, SEEK_SET);

    for(i = 0; i < pic_height[0]; i++){                // copy contents
```

```
        for(j = 0; j < pic_width; j++)
                fputc(fgetc(file1), out_file);
            for(j = 0; j < pic_width; j++)
                fputc(fgetc(file2), out_file);
        }
    fclose(file1);
        fclose(file2);
        fclose(out_file);
        return 0;
}
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

I17212194

4.1.5 การใช้งานโปรแกรม MERGE BMP.EXE

C:\>MERGEBMP [SOURCE1] [SOURCE2] [TARGET]

SOURCE1 คือ ไฟล์ฟอร์แมต BMP ภาพที่หนึ่ง

SOURCE2 คือ ไฟล์ฟอร์แมต BMP ภาพที่สอง

TARGET คือ ไฟล์ฟอร์แมต BMP ภาพที่สาม ซึ่งเป็นผลลัพธ์ จากการ merge กันของภาพที่หนึ่งและภาพที่สอง

4.1.6 โปรแกรมสร้างเส้นสีขาวเส้นสุดท้าย

โปรแกรมสร้างเส้นสีขาวเส้นสุดท้ายนี้ ทำหน้าที่สร้างเส้นสีขาวเส้นสุดท้ายในแต่ละฟิลด์ของภาพ ซึ่งพัฒนาขึ้นจากภาษาวิชวลเบสิก เวอร์ชั่น 4 ดังซอร์สโปรแกรมข้างล่าง

4.1.7 ซอร์สโปรแกรม (sorce program)

```
VERSION 4.00
Begin VB.Form WLC
   Appearance      =   0  'Flat
   BackColor       =   &H00000000&
   BorderStyle     =   0  'None
   Caption         =   "Form1"
   ClientHeight    =   1365
   ClientLeft      =   2175
   ClientTop       =   2970
   ClientWidth     =   6450
   BeginProperty Font
      name            =   "MS Sans Serif"
      charset         =   1
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
        weight          =   700
          size          =   8.25
          underline     =   0   'False
          italic        =   0   'False
          strikethrough =   0   'False
        EndProperty
        ForeColor       =   &H00FFFFFF&
        Height          =   1770
        Icon            =   "WLC.frx":0000
        Left            =   2115
        LinkTopic       =   "Form1"
        ScaleHeight     =   91
        ScaleMode       =   3   'Pixel
        ScaleWidth      =   430
        Top             =   2625
        Width           =   6570
    End
    Attribute VB_Name = "WLC"
    Attribute VB_Creatable = False
    Attribute VB_Exposed = False
    Private Declare Function SetWindowPos Lib "user" (ByVal h%, ByVal hb%, ByVal x%, ByVal
    y%, ByVal cx%, ByVal cy%, ByVal f%) As Integer
    ' The above Declare statement must appear on one line.

    Const SWP_NOMOVE = 2
```

```
Const SWP_NOSIZE = 1
Const FLAGS = SWP_NOMOVE Or SWP_NOSIZE
Const HWND_TOPMOST = -1
Const HWND_NOTOPMOST = -2

Private Sub Command1_Click()
ScaleMode = 3
DrawWidth = 4
Line (0, 10)-(100, 10), RGB(255, 255, 255)
Line (0, 1)-(Width * 0.67, 1), RGB(255, 255, 255)

End Sub

Private Sub Form_Load()

success% = SetWindowPos(WLC.hWnd, HWND_TOPMOST, 0, 0, 0, 0, FLAGS)
Rem success% <> 0 When Successful

Width = Screen.Width
Height = 2 * Screen.TwipsPerPixelY
Left = 0
Top = Screen.Height - Height
Show
ScaleMode = 3
DrawWidth = 1
Line (0, 1)-(ScaleWidth * 0.67, 1), RGB(255, 255, 255)
End Sub
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
Private Sub Form_Paint()
ScaleMode = 3
DrawWidth = 1
Line (0, 1)-(ScaleWidth * 0.67, 1), RGB(255, 255, 255)
End Sub
```

<u>เครื่องมือที่ใช้</u>

1. เครื่องไมโครคอมพิวเตอร์ไอบีเอ็ม หรือไอบีเอ็มคอมแพททิเบิล ที่มีไมโครโปรเซสเซอร์หมายเลข 80386 ขึ้นไป มีหน่วยความจำตั้งแต่ 4 เมกกะไบท์ และมีหน่วยขับจานแม่เหล็กขนาด 1.2 เมกกะไบท์ หรือ 1.44 เมกกะไบท์ อย่างน้อย 1 หน่วย

2. ซอฟท์แวร์ไมโคร์ซอฟท์วินโดว์ หรือไมโครซอฟท์วินโดว์ฟอร์เวอร์คกรุ๊ฟ รุ่น 3.X

3. จอภาพและวงจรแสดงผลแบบวีจีเอ ที่แสดงผลได้ 16 สี ขึ้นไป

4. ไมโครซอฟท์เมาส์ (mouse) และแผงแป้นอักขระ (keyboard)

5. เครื่องพิมพ์แบบอิงก์เจ็ต (inkjet)

4.2 <u>การออกแบบและพัฒนาเครื่องควบคุมแว่นผลึกเหลว</u>

คุณสมบัติของเครื่อง

สามารถตรวจจับรหัสเส้นขาวเส้นสุดท้ายได้ โดยใช้ดิพสวิทช์ (dip switch) เป็นตัวเลือกจำนวนเส้น แต่จะเป็นจำนวนเส้นที่เท่าใดนั้นขึ้นอยู่กับความละเอียดของโหมดการแสดงภาพใน

ขณะนั้น แล้วใช้ต่อกับแว่นผลึกเหลวเพื่อควบคุมการเปิดปิดแว่นให้ตรงกับการแสดงภาพจริงตามที่ตรวจจับได้

<u>เครื่องมือที่ใช้</u>

1. เครื่องไมโครคอมพิวเตอร์ไอบีเอ็ม หรือไอบีเอ็มคอมแพททิเบิล ที่มีไมโครโปรเซสเซอร์หมายเลข 80386 ขึ้นไป มีหน่วยความจำตั้งแต่ 4 เมกกะไบท์ และมีจานขับแม่เหล็กขนาด 1.2 เมกกะไบท์ หรือ 1.4 เมกกะไบท์ อย่างน้อย 1 หน่วย

2. ซอฟท์แวร์ไมโครซอฟท์วินโดว์ หรือไมโครซอฟท์วินโดว์ฟอร์เวอร์คกรุ๊ฟ รุ่น 3.X หรือไมโครซอฟท์วินโดว์รุ่น 95

3. ซอฟท์แวร์ชื่อ WLC.EXE

4. จอภาพและวงจรแสดงผลแบบวีจีเอ ที่สามารถใช้งานได้กับWLC.EXE

5. ไมโครซอฟท์เม้าส์และแผงแป้นอักขระ

6. อุปกรณ์วงจรรวมชนิดต่าง ๆ

7. ออสซิลโลสโคป (oscilloscope)

8. เครื่องกำเนิดสัญญาณรูปแบบต่าง ๆ ที่สามารถปรับความถี่และขนาดความสูงของสัญญาณได้

9. เครื่องจ่ายกำลังไฟฟ้า (power supply)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

10. แผงทดลองเอนกประสงค์

4.2.1 การออกแบบเครื่องควบคุมแว่นผลึกเหลว

จากทฤษฎีที่เกี่ยวข้องกับการสแกนแบบแทรกสอดที่กล่าวมาแล้ว จะเห็นได้ว่าในการสแกนภาพ 1 ภาพ หรือ 1 เฟรม (frame) นั้น จะถูกแบ่งการสแกนออกเป็น 2 ฟิลด์ แบบเส้นเว้นเส้น คือสแกนเส้นคี่ก่อนแล้วจึงกลับมาสแกนเส้นคู่อีก โดยเริ่มต้นภาพที่เป็นฟิลด์ที่เรากำหนดให้แว่นด้านขวาทึบแสง และในฟิลด์คู่เรากำหนดให้แว่นด้านซ้ายทึบแสง ซึ่งมีความถี่ในการเปิดปิดดังกล่าวเท่ากับความถี่ของสัญญาณหักเหทางแนวตั้ง โดยมีค่าเปลี่ยนแปลงตามโหมดความละเอียดของการแสดงภาพ จากการสแกนแบบนี้เอง จึงทำให้สามารถแสดงภาพของตาซ้ายและขวาได้ โดยการรวมภาพของตาซ้ายและขวาเข้าด้วยกันแบบเส้นเว้นเส้น ดังนั้น เมื่อนำมาแสดงในโหมดการทำงานดังกล่าว ก็จะได้ภาพซ้าย-ขวาที่สลับกันโดยหลักการสแกนแบบแทรกสอดนั่นเอง แต่การแสดงภาพ 3 มิติ ด้วยวิธีนี้ จะมีผลให้ความละเอียดของภาพลดลงครึ่งหนึ่งด้วยเช่นกัน

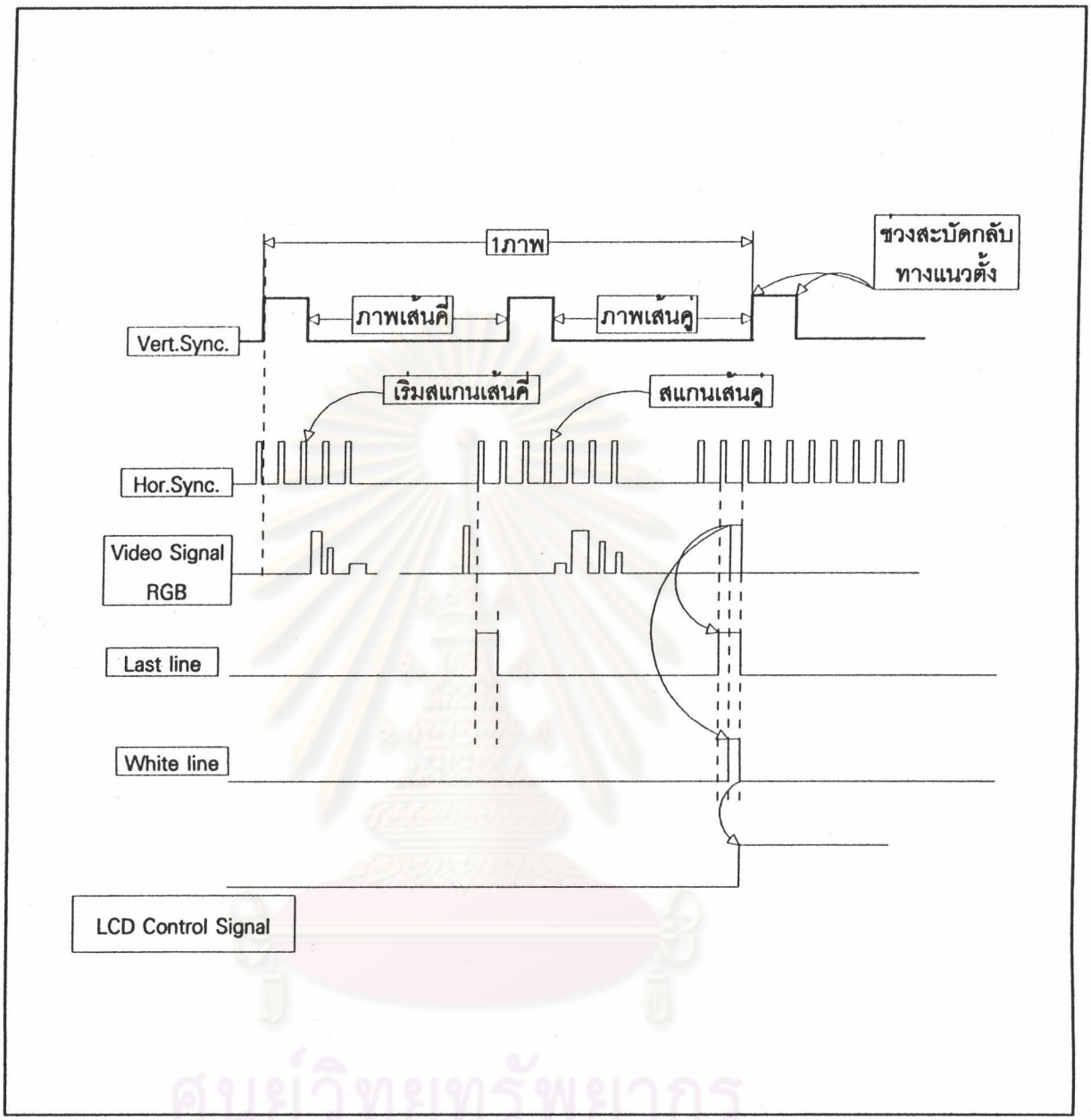
ในรูปที่ 4.1 เป็นรูปคลื่นที่แสดงความสัมพันธ์ทั้งหมดของการแสดงภาพที่ได้จากการ์ดแสดงภาพแบบวีจีเอ โดยใช้โปรแกรม WLC ทำหน้าที่สร้างเส้นสีขาวเส้นสุดท้ายขึ้นมาในโหมดการสแกนแบบแทรกสอด ซึ่งเป็นเส้นขาวของฟิลด์คู่ โดยจากรูปจะเห็นว่าในหนึ่งรูปคลื่นของสัญญาณหักเหทางแนวตั้ง จะประกอบไปด้วยรูปคลื่นของสัญญาณหักเหทางแนวนอนจำนวนมาก ซึ่งเท่ากับครึ่งหนึ่งของความละเอียดของการแสดงภาพทางแนวนอน และในสัญญาณหักเหทางแนวนอนแต่ละรูปจะประกอบไปด้วยสัญญาณภาพซึ่งมีระดับของสัญญาณที่แตกต่างกัน สูงต่ำไม่เท่ากัน โดยระดับสัญญาณสูงนั้นจะมีลักษณะเป็นจุดที่สว่างหรือจุดสีขาว ซึ่งสัญญาณซึ่งเป็นเส้นสีขาวเส้นสุดท้ายนี้จะมีความสูงของระดับสัญญาณสูงสุดคือ ประมาณ 0.7 โวลท์ และมีความกว้างของรูปคลื่นประมาณ 12 ไมโครเซกกันด์ จะเห็นว่า ถ้าสามารถจับเส้นสุดท้ายของการสแกนในทุก ๆ ฟิลด์ได้ แล้วนำมาเปรียบเทียบกับสัญญาภาพอีกครั้ง ก็สามารถตรวจจับรหัสเส้นขาวได้แล้ว ซึ่งสัญญาณภาพเส้นสุดท้ายนี้ก็คือ รหัสเส้นขาวนั่นเอง ในความเป็นจริงรหัสเส้นขาวนั้นจะเกิดขึ้นที่เส้นสุดท้ายของทุกฟิลด์ ทั้งเส้นคี่และคู่สลับกันไปตลอดเวลา ในการตรวจจับรหัส

เส้นสุดท้ายนี้ เราใช้เส้นสุดท้ายของฟิลด์คู่ คือเส้นขนาด 2/3 ของจอ แล้วนำมาใช้สร้างเป็นสัญญาณควบคุมการเปิดปิดแว่นผลึกเหลว ซึ่งจากการทดลองทำให้เราทราบว่า สัญญาณภาพเส้นสุดท้ายของแต่ละฟิลด์นั้นไม่ได้เกิดที่จำนวนเส้นเดียวกัน ดังนั้น การออกแบบวงจรจึงไม่จำเป็นต้องเปรียบเทียบว่าเป็นเส้นคู่หรือคี่ เพราะสามารถเลือกได้จากการนับที่จำนวนเส้นเลย เช่น ในโหมดการแสดงภาพที่ความละเอียด 1024 คูณ 768 จุด จะได้ค่าของเส้นสุดท้ายที่นับจากฮอร์ซิงค์เป็น 404 สำหรับฟิลด์คี่ และ 405 สำหรับฟิลด์คู่ เป็นต้น

จากรูปสัญญาณทั้งหมดจะเห็นว่า ที่ขอบขาขึ้นของสัญญาณหักเหทางแนวตั้งจะไปตรงกับขอบขาลงของสัญญาณหักเหทางแนวนอน โดยความกว้างของสัญญาณหักเหทางแนวตั้งจะมีค่าประมาณ 3 ลูก ของสัญญาณหักเหทางแนวนอน ซึ่งความกว้างของสัญญาณหักเหทางแนวตั้งจะใช้เป็นช่วงสะบัดกลับในการแสดงภาพของแต่ละฟิลด์ จากนั้นจะเริ่มมีสัญญาณภาพเข้ามาหลังจากที่สัญญาณหักเหทางแนวนอนเกิดเล็กน้อย ซึ่งไม่ตรงกับขอบของสัญญาณหักเหทางแนวนอน คือจะอยู่ในช่วงของสัญญาณหักเหทางแนวนอนนั่นเอง ส่วนสัญญาณเส้นขาวเส้นสุดท้ายจะเกิดใกล้ ๆ กับสัญญาณหักเหทางแนวตั้งลูกถัดไป ซึ่งเราใช้สัญญาณดังกล่าวนี้ไปกำหนดเกี่ยวกับวงจรควบคุมแว่นผลึกเหลว

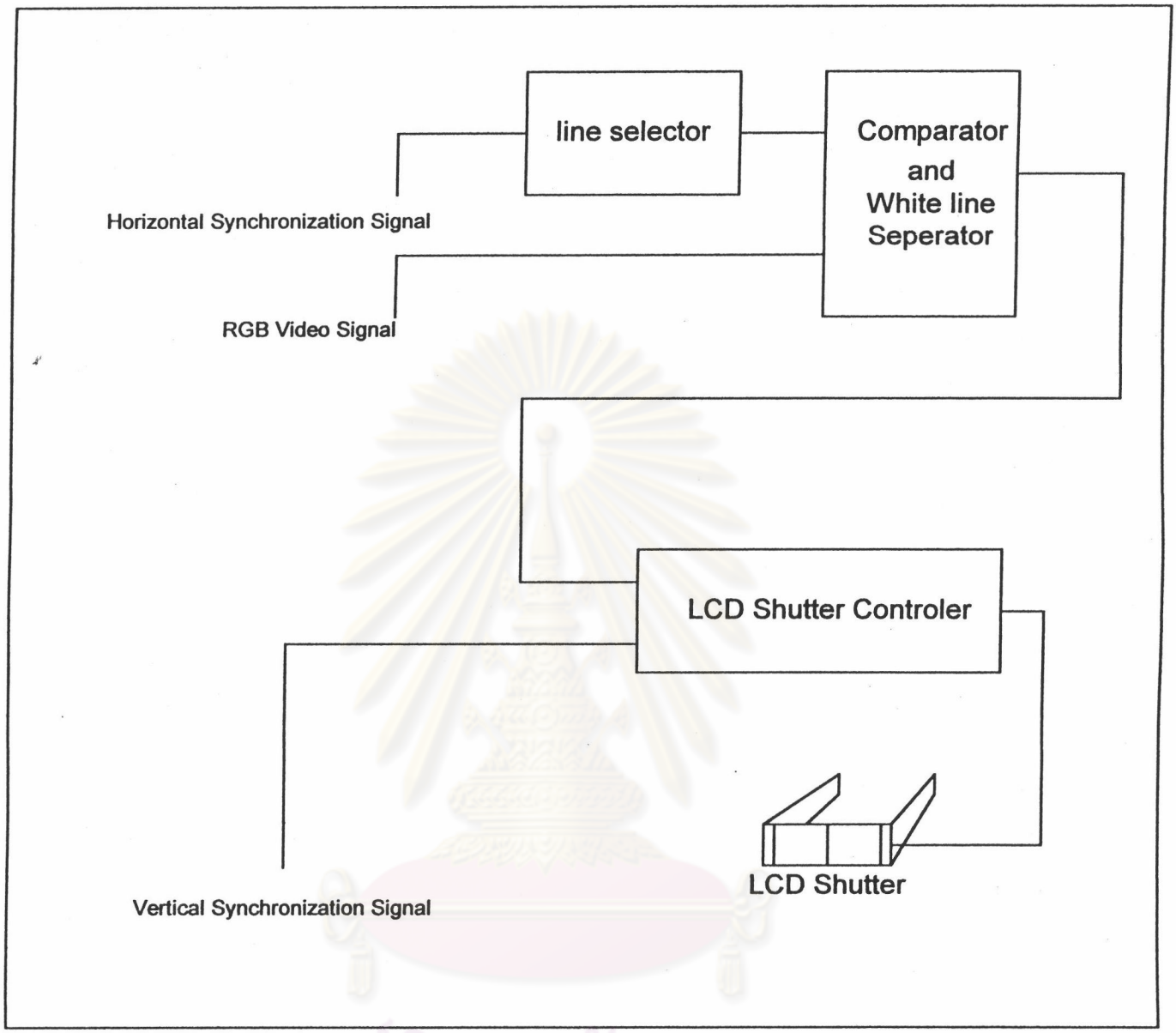
สำหรับแนวทางในการออกแบบ แสดงในรูปที่ 4.2 กล่าวคือ จะใช้การนับสัญญาณหักเหทางแนวนอนเป็นหลัก แล้วใช้สัญญาณหักเหทางแนวตั้งเป็นตัวกำหนดให้เริ่มต้นนับ ซึ่งในการนับลูกคลื่นของสัญญาณหักเหทางแนวนอนนั้น จะต้องใช้จำนวนเส้นที่ได้จากความละเอียดในการแสดงภาพ รวมกับช่วงเวลาก่อนการแสดงภาพอีก 560 us และรวมความกว้างของสัญญาณหักเหทางแนวตั้งอีก 110 us เข้าไปด้วย เนื่องจากเริ่มนับที่จุดนี้ จึงทำให้จำนวนเส้นของการนับเส้นสุดท้ายมีค่ามากกว่าความเป็นจริง ซึ่งสามารถอธิบายได้จากรูปที่ 4.3

รูปที่ 4.3 แสดงช่วงเวลาที่เกิดของสัญญาณที่ต่างกัน จะเห็นได้ว่าสัญญาณภาพนั้นไม่ได้เริ่มต้นเกิดตอนที่สัญญาณหักเหทางแนวตั้งเริ่มขึ้นทันที แต่จะทิ้งช่วงเวลาไปอีก 560 us และจบลงก่อนที่สัญญาณหักเหทางแนวตั้งลูกใหม่เกิดอีก 20 us ในขณะที่สัญญาณหักเหทางแนวนอนมีขนาด 28.1 us แต่ช่วงเวลาของการหักเหทางแนวตั้งนั้นยาวครอบคลุมทั้งหมด ดังนั้นถ้านับจำนวนของเส้นหักเหทางแนวนอนเทียบกับการหักเหทางแนวตั้ง จึงมีจำนวนเส้นมากกว่าจำนวนเส้นที่สแกนจริง ๆ ที่จอภาพ



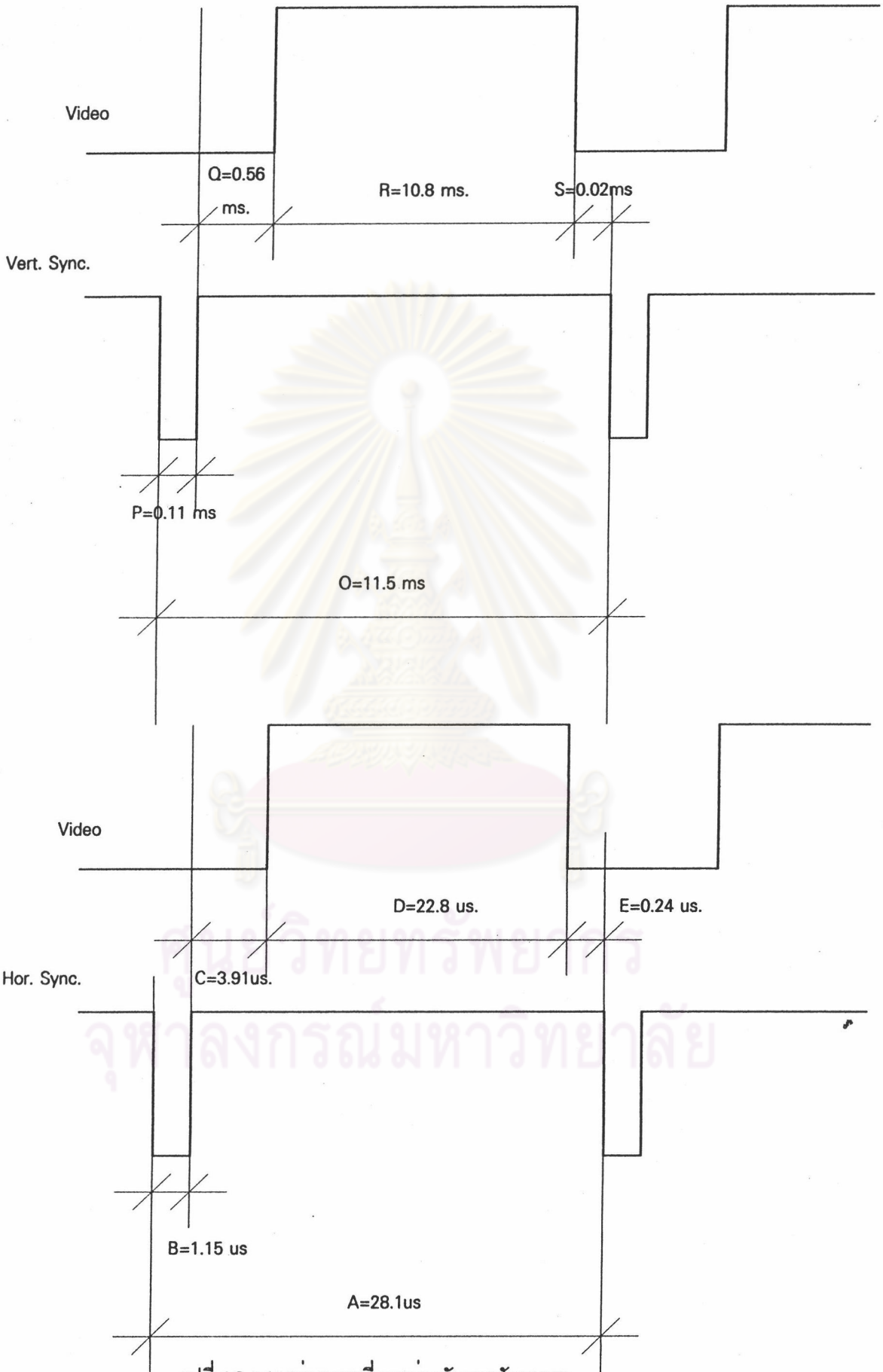
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่4.1 แสดงรายละเอียดของสัญญาณที่ใช้ในการออกแบบวงจร



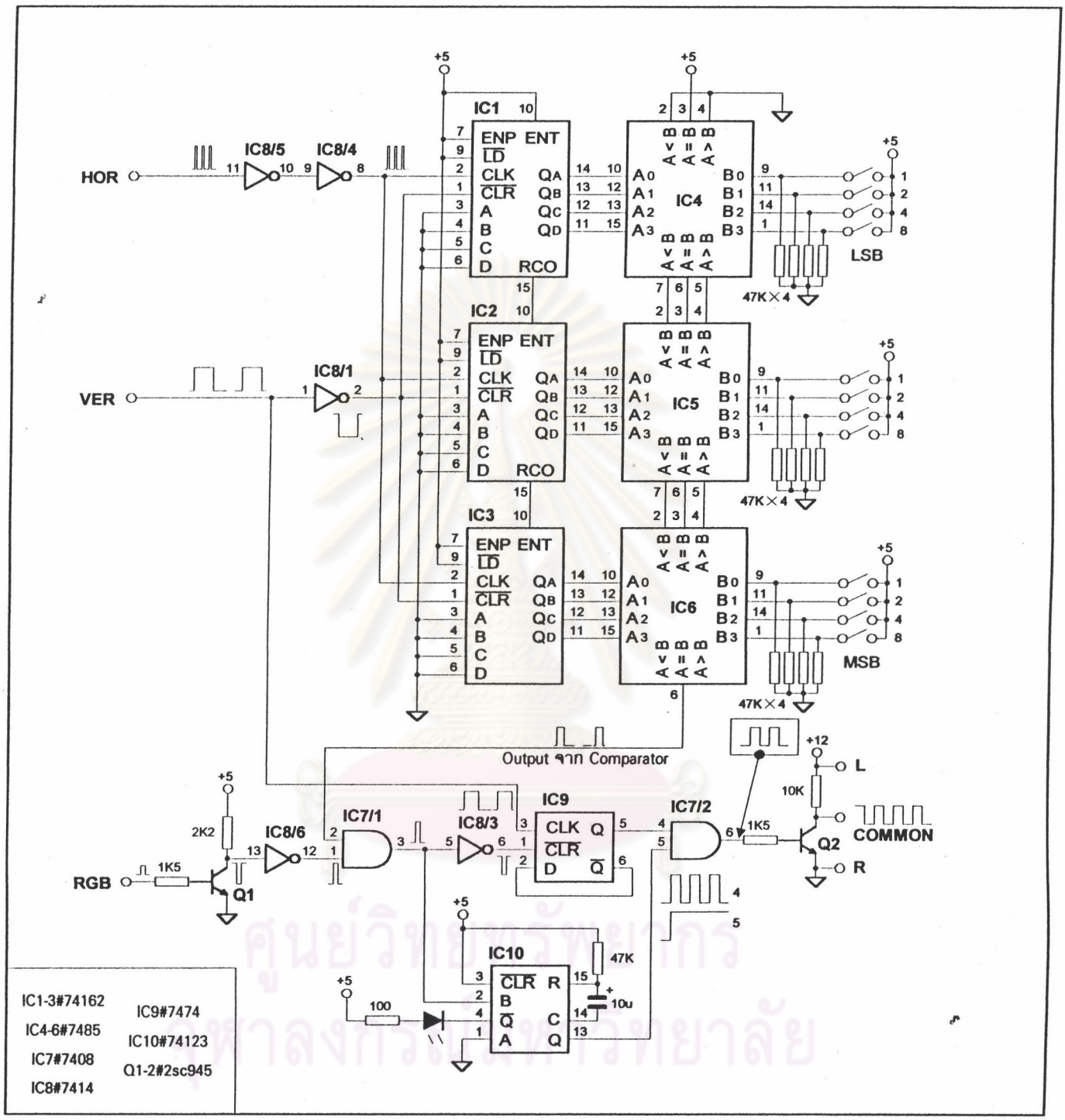
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 4.2 แสดงแนวการออกแบบเครื่องควบคุมแว่นผลึกเหลว



รูปที่ 4.3 แสดงช่วงเวลาที่แตกต่างกันของสัญญาณ
จาก Operating Manual ของ Monitor CVP-5468NI

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.4 แสดงวงจรตรวจจับเส้นขาวเส้นสุดท้าย

การทำงานของวงจร

รูป 4.4 เป็นวงจรตรวจจับรหัสเส้นขาว (white line code) แล้วนำมาสร้างสัญญาณควบคุมการทำงานของแว่นผลึกเหลว เพื่อให้การเปิดปิดของแว่นข้างซ้ายและขวาสัมพันธ์กันกับเฟรมของภาพที่แสดงบนจอภาพ ที่ความละเอียด 1024 คูณ 768 ในโหมดอินเทอเลส หลักการตรวจจับรหัสเส้นขาวกระทำโดยตรวจจับสัญญาณฮอซิงค์ลูกสุดท้ายของเฟรมแล้วนำมา AND กับสัญญาณ RGB ซึ่งจะเป็นสัญญาณของสีขาวในเส้นสุดท้ายของเฟรมเช่นกัน เอาท์พุตที่ได้จะนำมาเป็นสัญญาณซิงค์ให้แว่นตาเปิดปิดตรงกับเฟรมของภาพ

เราใช้สัญญาณหักเหทางแนวนอนมาผ่านขาไอซี 8/5 ซึ่งเป็นไอซีเบอร์ 7414 ทำหน้าที่เป็นอินเวอร์เตอร์ (inverter) และชมิดทิกเกอร์ (schmittigger) เพื่อจัดรูปของสัญญาณให้มีรูปร่างที่ถูกต้อง เพื่อใช้เป็นสัญญาณนาฬิกาให้วงจรนับ (counter) ซึ่งวงจรนับที่ใช้เป็นแบบซิงโครนัสเคาน์เตอร์ (synchronous counter) ส่วนสัญญาณหักเหทางแนวตั้งจะผ่านไอซี 8/1 เพื่อกลับเฟสของสัญญาณเพื่อใช้ในการเคลียร์ (clear) เคาน์เตอร์ เพื่อให้เริ่มนับหลังจากที่สัญญาณหักเหทางแนวตั้งสิ้นสุด จนกว่าจะเจอสัญญาณหักเหทางแนวตั้งลูกใหม่ เมื่อวงจรเคาน์เตอร์ถูกเคลียร์ด้วยสัญญาณหักเหทางแนวตั้งแล้ว สัญญาณหักเหทางแนวนอนที่ป้อนเข้าขา CLK วงจรจะเริ่มนับและเอาท์พุทของเคาน์เตอร์จะปรากฏที่ขา QA, QB, QC, QD ของไอซี 1 ก่อน ส่วนไอซี 2, 3 จะยังไม่ทำงาน เนื่องจากที่ขา ENT ของไอซี 2 และ 3 มีสถานะเป็น low ทำให้วงจรเคาน์เตอร์ยังไม่ทำงาน ส่วนไอซี 1 นั้น ต่อขาของ ENT เข้ากับ VCC เพื่อให้ทำงานได้ตลอด เมื่อเคาน์เตอร์ตัวแรกนับจาก 0 ถึง 9 แล้ว ขา RCO ของไอซี 1 จะมีค่าเป็น high ส่งไปที่เคาน์เตอร์ตัวที่ 2 (ไอซี 2) ทำงาน ส่วนไอซี 3 ก็เช่นเดียวกัน ต้องรอสัญญาณ high ที่มาจากขา RCO ด้วย จากนั้นจึงต่อขาของเอาท์พุทของเคาน์เตอร์เข้ากับคอมพาราเตอร์ (comparator) ขนาด 4 บิต โดยด้าน A ต่อเข้ากับเอาท์พุทของเคาน์เตอร์ ส่วนด้าน B ต่อเข้ากับดิพสวิทช์ ดังรูป เมื่อเราตั้งค่าของสวิทช์ไว้มีค่าเท่าใด ถ้า A กับ B มีค่าเท่ากันหรือตรงกัน จะมีสัญญาณ A = B ออกมาจากขา 6 แต่ถ้าไม่ตรงกันก็จะไม่มีเอาท์พุทออกมา ซึ่งเราต่อสัญญาณ A = B ไปยังตัวต่อไปด้วย ซึ่งสัญญาณ A = B ของคอมพาราเตอร์ตัวสุดท้าย จะถูกนำไปเปรียบเทียบกับสัญญาณ RGB โดยสัญญาณ RGB จะถูกขยายให้เป็นระดับดิจิตอลก่อน เนื่องจากสัญญาณ RGB นั้น เป็นสัญญาณอนาลอก มีระดับความสูงของ

สัญญาณประมาณ 0.7 โวลท์ ซึ่งไม่สามารถนำมาใช้งานได้ จึงต้องขยายสัญญาณจาก 0.7 ให้เป็น 5 โวลท์ ด้วยทรานซิสเตอร์ Q1 เบอร์ 2SC945 เอาท์พุทที่ได้จาก Q1 จะกลับเฟสจากสัญญาณ RGB แต่มีความสูงเท่ากับ 5 โวลท์ เราจึงต้องต่อกับไอซี 8/6 เพื่อกลับเฟสของสัญญาณให้ตรงกัน จากนั้นก็ป้อนเข้าขาของไอซี 7/1 เพื่อเปรียบเทียบกับสัญญาณที่มาจากคอมพาราเตอร์ ถ้าสัญญาณที่ได้จากเคาน์เตอร์มีตำแหน่งตรงกับสัญญาณที่เกิดเส้นขาวเส้นสุดท้าย ก็จะมีเอาท์พุทออกมาจากแอนด์เกท จากเอาท์พุทของแอนด์เกทส่วนหนึ่งจะถูกส่งไปยังโมโนสเตเบิล (monostable) แบบ retriggable ซึ่งถ้าไม่มีสัญญาณสีขาวเส้นสุดท้ายมาแล้วเอาท์พุทจากไอซี 7/1 ก็จะไม่มีออกมา วงจรโมโนก็จะไม่ทำงานด้วย เมื่อเราได้สัญญาณจากการเปรียบเทียบกับเส้นขาวเส้นสุดท้ายแล้ว คอมพาราเตอร์จะถูกกลับเฟสด้วยไอซี 8/3 เพื่อเข้าไปเคลียร์ไอซี 9 ซึ่งเป็นทีฟลิปฟลอป ขา CLK ของไอซี 9 จะใช้สัญญาณหักเหทางแนวตั้งเป็นตัวกำเนิดสัญญาณออกมาที่ Q ต่อไป AND กับโมโน การทำงานของไอซี 9 จะเริ่มจาก เมื่อมีสัญญาณสีขาวเส้นสุดท้าย และมีเอาท์พุทออกมาจากไอซี 7/1 เป็น high และอินเวอร์ทเป็นตรงกันข้าม เพื่อเคลียร์ไอซี 9 ก่อน เมื่อสัญญาณเคลียร์เป็น high แล้ว มีสัญญาณหักเหทางแนวตั้งมาที่ขา CLK ก็จะมีเอาท์พุทออกที่ขา Q โดยที่ Q จะมีค่าเท่ากับ D จากวงจรไอซี 9 นั้น เราต่อวงจรลักษณะทีฟลิปฟลอป คือเราเอา D ต่อกับคิวบาร์ ซึ่งเมื่อไอซี 9 ถูกเคลียร์นั้น สถานะของ Q จะเป็น low ส่วนคิวบาร์จะเป็น high เมื่อเริ่มทำงาน D จะได้สถานะเป็น high ทำให้ Q เป็น high ด้วย เมื่อสัญญาณหักเหทางแนวตั้งลูกต่อไปเข้ามา ก็จะทำให้ Q เป็น low เนื่องจาก D ต่อกับคิวบาร์ ซึ่งเอาท์พุทที่ได้จากไอซี 9 นั้น จะถูกหารลงครึ่งหนึ่งของสัญญาณหักเหทางแนวตั้ง จากนั้นสัญญาณที่ออกจากไอซี 9 จะนำไป AND กับสัญญาณที่มาจาก monoretriggable ซึ่งจะเป็น high ตลอด เอาท์พุทที่ได้จากไอซี 7/2 จะเป็นรูปคลื่นเดียวกับเอาท์พุทที่ออกมาจากไอซี 9 จากเอาท์พุทของไอซี 7/2 จะถูกส่งไปยังทรานซิสเตอร์ Q2 เพื่อขยายสัญญาณให้เป็นระดับ 12 โวลท์ เพื่อใช้ป้อนให้กับแว่นอีกที แต่เมื่อเรายกเลิกการทำงานของโปรแกรมก็จะไม่มีเอาท์พุทออกจากไอซี 7/1 เป็นผลให้ไอซี 9 และไอซี 10 ไม่ทำงาน และ Q2 ก็ไม่ทำงานด้วย