Application Note

Applying eZSelect Program Blocking to PIP Circuits

AN005000-TVC1299

ZiLOG Worldwide Headquarters • 910 E. Hamilton Avenue • Campbell, CA 95008
Telephone: 408.558.8500 • Fax: 408.558.8300 • http://www.ZiLOG.com

This publication is subject to replacement by a later edition. To determine whether a later edition exists, or to request copies of publications, contact

ZiLOG Worldwide Headquarters
910 E. Hamilton Avenue
Campbell, CA 95008
Telephone: 408.558.8500
Fax: 408.558.8300
www.ZiLOG.com

Information Integrity

The information contained within this document has been verified according to the general principles of electrical and mechanical engineering. Any applicable source code illustrated in the document was either written by an authorized ZiLOG employee or licensed consultant. Permission to use these codes in any form, besides the intended application, must be approved through a license agreement between both parties. ZiLOG will not be responsible for any code(s) used beyond the intended application. Contact the local ZiLOG Sales Office to obtain necessary license agreements.

Document Disclaimer

Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZiLOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZiLOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. Except with the express written approval ZiLOG, use of information, devices, or technology as critical components of life support systems is not authorized. No licenses or other rights are conveyed, implicitly or otherwise, by this document under any intellectual property rights.

Table of Contents

Acknowledgement: Allen Chang, Principal Engineer

List of Figures

Applying eZSelect Program Blocking to PIP Circuits

Overview

Effective January 1, 2000, all TV receivers with a picture screen larger than 13 inches that are shipped through interstate commerce or manufactured in the United States must be equipped with V-Chip Program blocking technology based on FCC requirements. The same Program blocking requirement applies to TV sets with the Picture-in-Picture (PIP) function. This means that the PIP circuit must be able to decode program rating information for the PIP video signal.

ZiLOG's eZSelect Z86130/Z86230 program blocking devices are stand-alone, off-the-shelf, low-cost devices and are the best chips on the market today to connect to the PIP circuit to decode and block program rating information intelligently. Figure 1 is a system application diagram that illustrates how the eZSelect Z86130/Z86230 devices can integrate with PIP circuit.

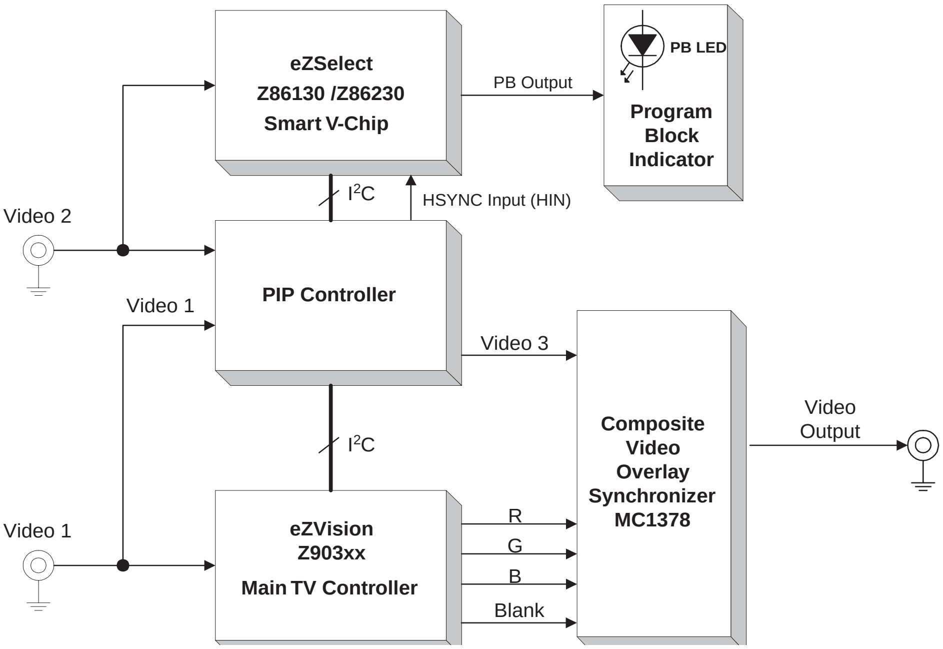

Figure 1 System Application Diagram

This application note discusses how well ZiLOG's eZSelect Z86130 and Z86230 data decoders work with two popular PIP chips

- Motorola MC44461
- Siemens SDA9288

All communication between the TV main controller, Z86130, Z86230, and PIP devices uses I^{2}C (Inter-Integrated Circuit) bus protocol. The I^{2}C driver routines for the Z86130 and Z86230 are included in this application note.

Basic Hardware Requirements

The basic hardware requirements to connect the Z86130/Z86230 in a PIP circuit are listed below.

1. PIP Video signal. The same video signal input to the PIP circuit.
2. I^{2}C Signals (SDA and SCL). These signals are the clock and data signals on the I^{2}C bus of the PIP circuit.
3. Timing reference supporting to pin 5 of the Z86130 or Z86230. The timing reference can use either the Z86130/230 crystal (32.768 KHz) or be directly connected to the H_{SNYC} signal as the timing reference source.
4. Proper I^{2}C address selection (`28h (W)` or `2Ah(W)`) on pin 1 for the Z86230.
5. A +5V power suppy to Z86130 or Z86230.

Hardware Design Tips for the Z86130 and Z86230

Before designing the Z86130/Z86230 into any circuits, some design issues must be established to ensure design reliability.

1. **Video input on pin 7.** The video signal amplitude must meet Z86130/Z86230 specifications. The noise filtering circuit in front of pin 7 must follow the reference circuit in the Z86230 Product Specification or circuits that follow the schematic diagrams presented below. Good noise filtering can prevent the chip from locking up and improve the data decoding reliability.
2. **H_{SNYC} input on pin 5.** If H_{IN} is used for pin 5, the input signal must be H_{SNYC} . Do NOT use a Composite sync (C_{SNYC}) signal. Using a Composite sync signal causes decoding to be unreliable.
3. **LPF and C_{SYNC} on pin 9 and pin 8.** Use the resistor and capacitor values on the reference circuit in the Z86230 Product Specification, or values that follow the schematic diagrams presented below, for better reliability.

PIP Hardware Module with the Motorola MC44461 and ZiLOG's Z86130/Z86230

Figures 2 through 4 are schematic diagrams of the PIPcircuit using a Motorola MC44461 and ZiLOG's eZSelect Z86130/Z86230. The Motorola MC44461 accepts two video base-band signals as input. Either of them can be used for the Main video picture or the Sub video picture (PIP). Therefore, video switching must be established to determine the proper video signal for the Z86130/Z86230. LM1881 is used to separate H_{SNYC} and V_{SNYC} and provide the timing reference for the Z86130/Z86230. The LM6181, the video amplifier, increases the current gain of the video signal.

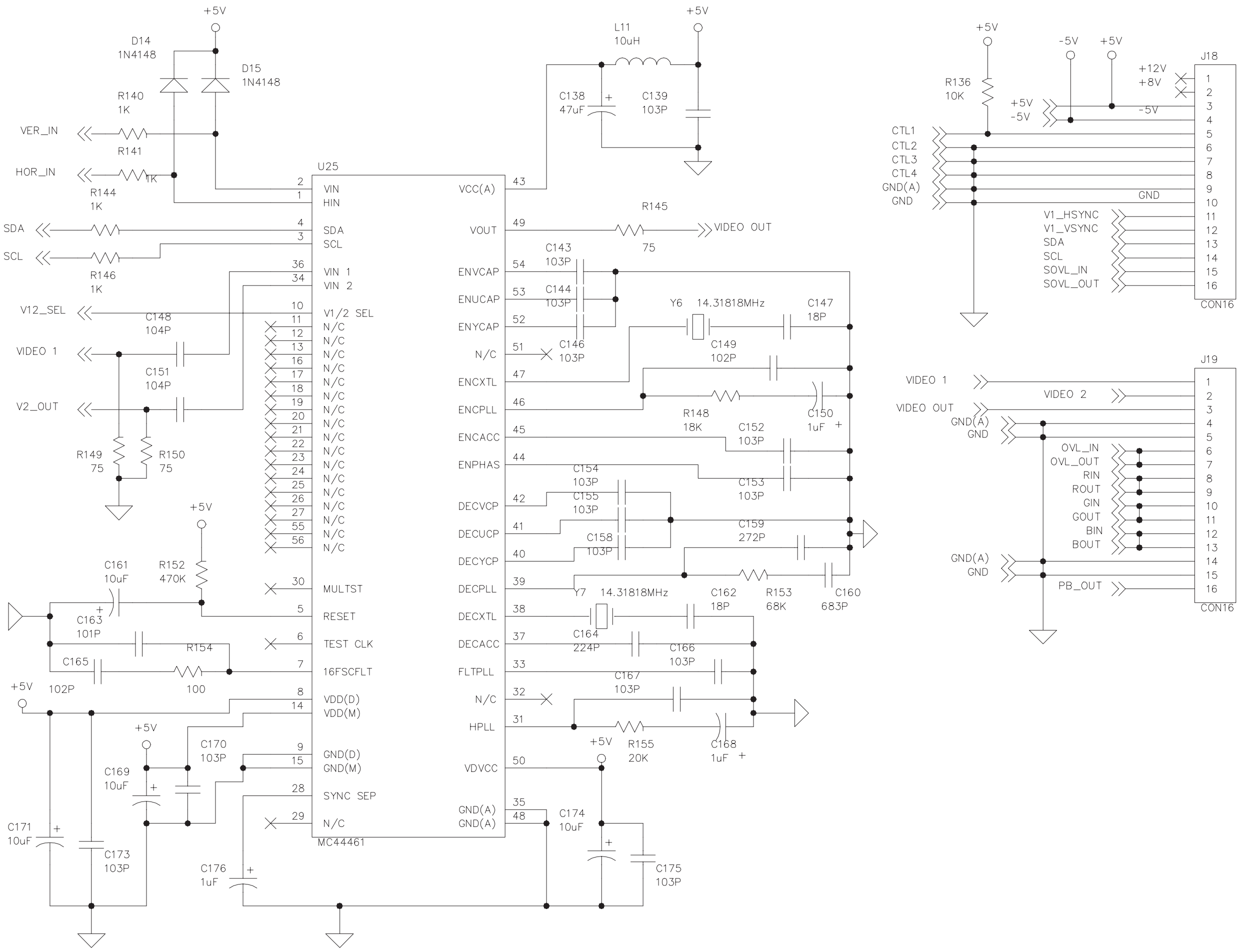

Figure 2 PIP Module with the Motorola MC44461 and Z86130/230 Part 1

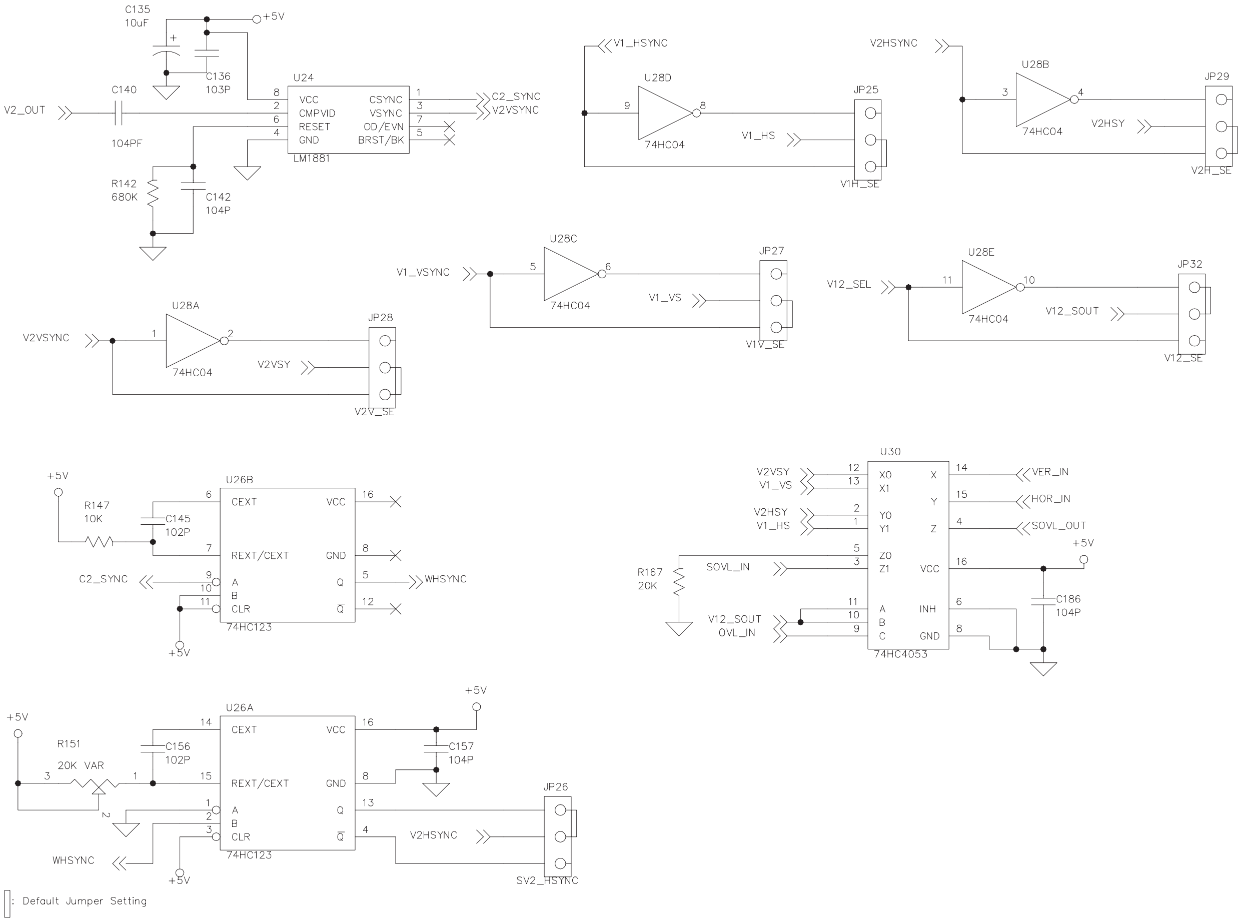

Figure 3 PIP Module with the Motorola MC44461 and Z86130/230 Part 2

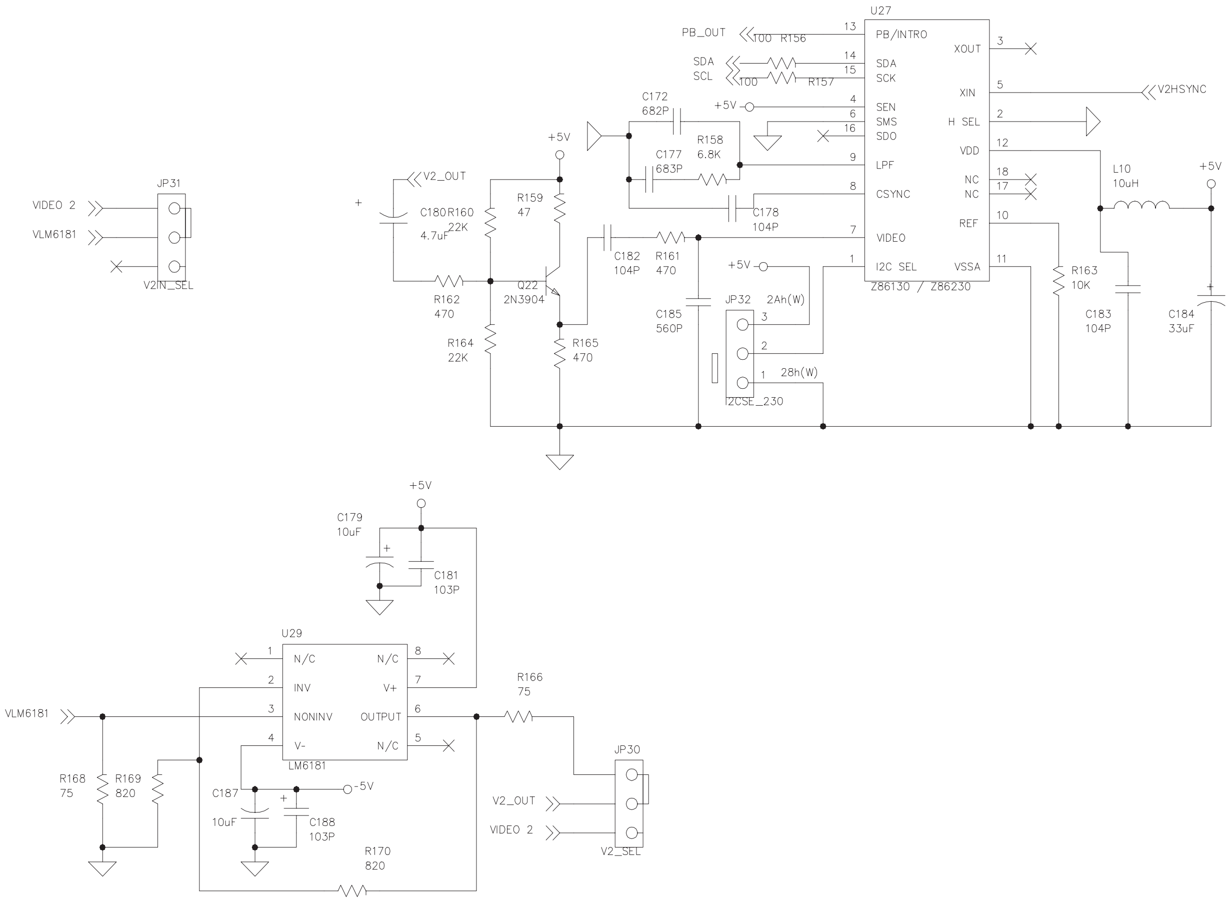

Figure 4 PIP Module with the Motorola MC44461 and Z86130/230 Part 3

PIP Hardware Module with the Siemens SDA9288 and ZiLOG's Z86130/Z86230

Figures 5 through 7 are the schematic diagrams of the PIP circuit using Siemens SDA9288 and ZiLOG's eZSelect Z86130/Z86230. The SDA9288 (U22) accepts Y, U, V signals instead of a composite video signal as input. A TDA8315 (U18) (video decoder) is used to convert video signal to Y, U, V. The H_{SNYC} signal can be generated directly from the TDA8315 device to be used as the timing reference for pin 5 (HIN) of the Z86130/Z86230, or it can be adjusted for the correct pulse width by the 74HC123 (U23A) device and used as the timing reference.

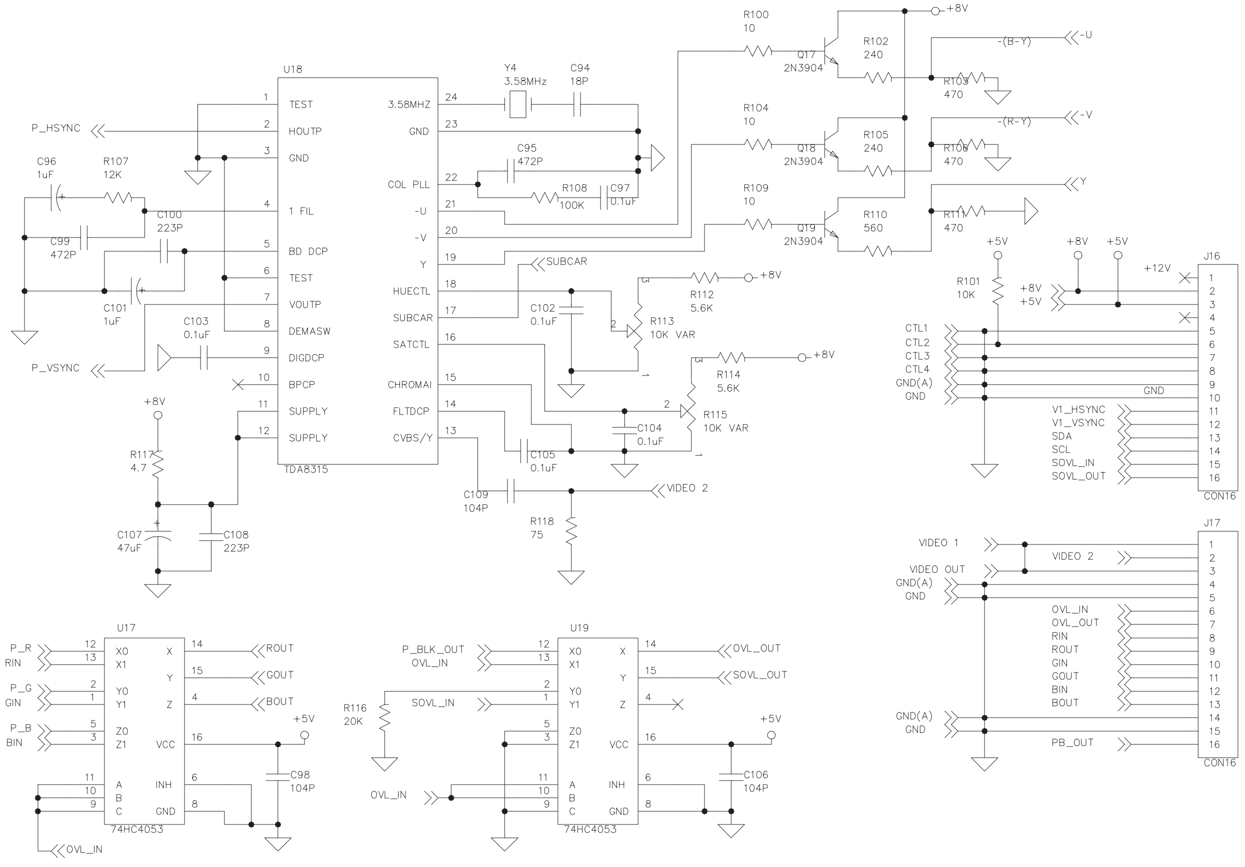

Figure 5 PIP Module with the Siemens SDA9288 and ZiLOG's Z86130/230 Part 1

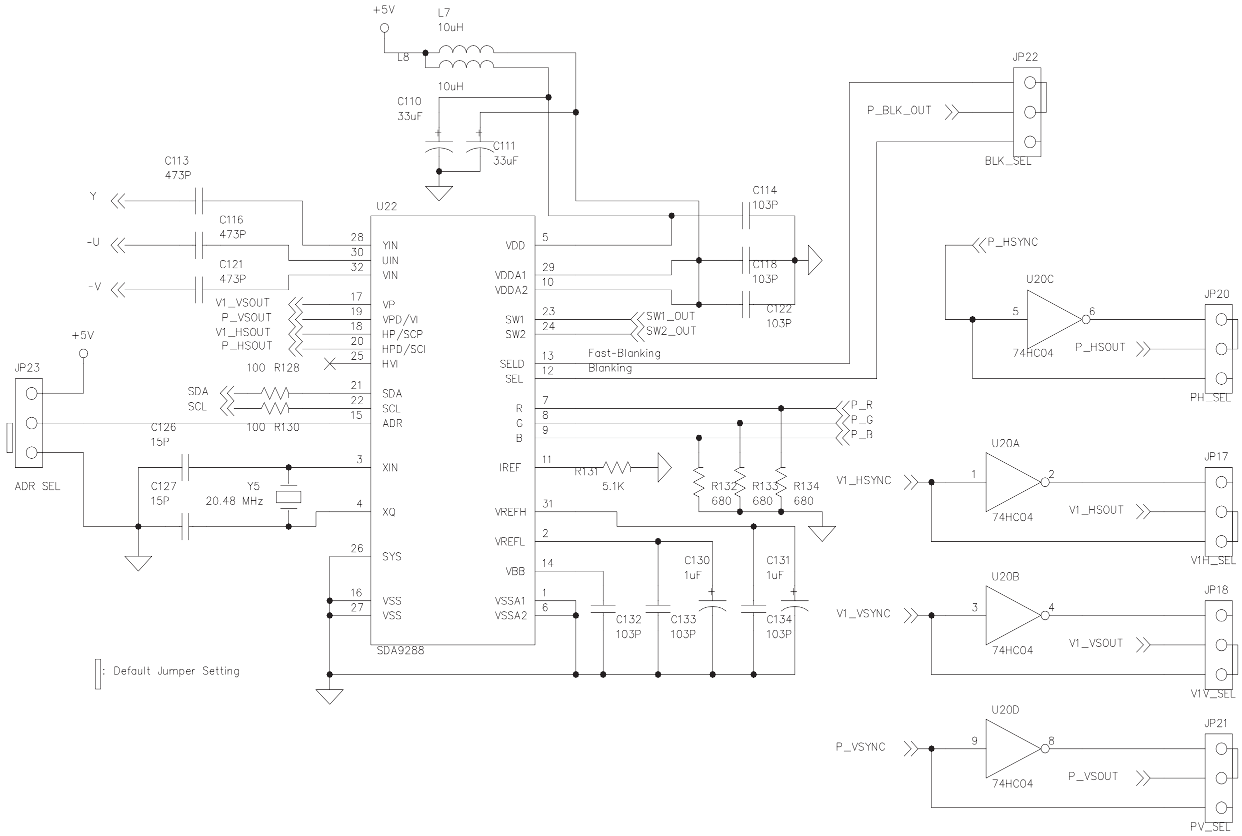

Figure 6 PIP Module with the Siemens SDA9288 and ZiLOG's Z86130/230 Part 2

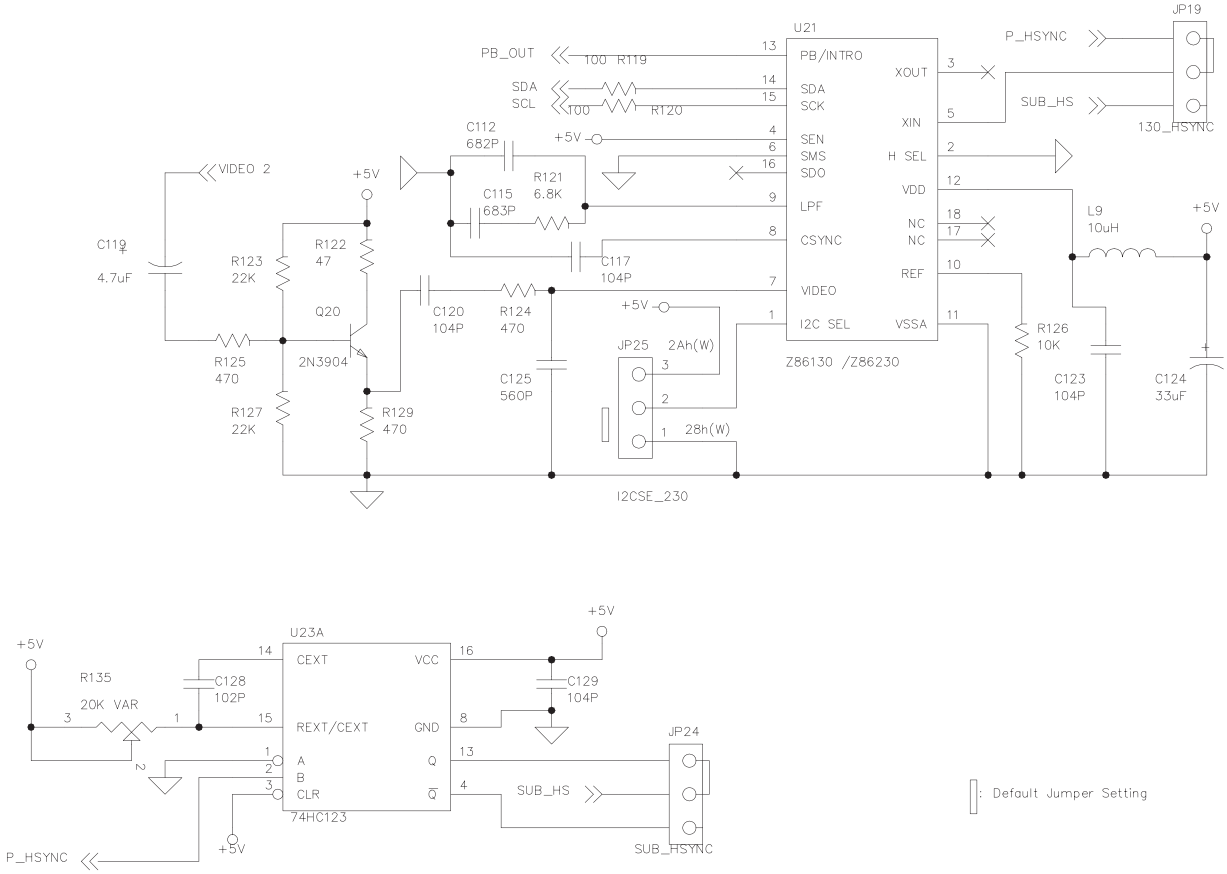

Figure 7 PIP Module with the Siemens SDA9288 and ZiLOG's Z86130/230 Part 3

Example Software Driver in C

The following example routine called, GetVchip( ), is written in C code to demonstrate how to read 2-byte data from internal registers `0Ch` and `0Dh` of the eZSelect Z86130 or Z86230 data decoder. After successfully reading the 2 bytes from the registers, the routine processes the data and displays the rating information on the PC. This is a routine in the C Language used on the eZSelect line 21 decoder reference board. With some modifications, this routine can be ported to the main microcontroller in the embedded system design.

```
void GetVchip()
{
int out1=1;
int byte1, byte2, vstat, temp;
unsigned char sbuf[10], srbuf[10], tst;
int ratesys, vclp=1;

while(out1)
{
     ccd2add = i2c230;           // now using I2C address
                                 // selected for Z86130 or
                                 // Z86230
      wrserial(0x6C);            // "read 2 bytes" command
                                 // to read data in
RegisterC&D
      i2c_stop_pc();

      srbuf[0] = byte1 = rdserial(); // Read status and 1st
                                      // byte. Return the 1st
                                     // byte to byte1
      sendack();                 // Send ACK
      srbuf[1] = byte2 = i2c_rdata_pc();
      i2c_stop_pc();
```

```
vstat = byte1;
vstat >>= 3;
vstat &= 0x07;

temp = byte2;
temp &= 0x08;
vstat = vstat | temp;      // get rating system in
                           // place

if(byte1 && 0x80)
   vstat |= 0x40;          // set block bit
else
   vstat &= 0xBF;          // reset block bit

if(byte2 && 0x80)
   vstat |= 0x20;          // set recovery bit
else
   vstat &= 0xDF;          // reset recovery bit

srbuf[2] = vstat;

if(((srbuf[2] & 0x20) == 0x20) || (vclp ==1))
{
   printf("%02X,  %02X,  %02X  \n", srbuf[0], srbuf[1],
   srbuf[2]);
   vclp = 0;

   switch(srbuf[2] & 0x0f)
```

```
{
   case 0x00:
   case 0x04:
   case 0x08:
   case 0x0c:
      printf("Rating System: MPAA \n");
      ratesys = 0;
      break;
   case 0x02:
   case 0x06:
   case 0x0a:
   case 0x0e:
      printf("Rating System: MPAA \n");
      ratesys = 2;
      break;
   case 0x01:
   case 0x05:
   case 0x09:
   case 0x0d:
      printf("Rating System: U.S. TV Parental
      Guidelines \n");
      ratesys = 1;
      break;
   case 0x03:
      printf("Rating System: Canadian English
      Language Rating \n");
      ratesys = 3;
      break;
   case 0x07:
      printf("Rating System: Canadian French
      Language Rating \n");
      ratesys = 4;
      break;
   case 0x0b:
      printf("Rating System: Reserved for non-U.S. &
      non-Canadian system \n");
      ratesys = 5;
      break;
   case 0x0f:
      printf("Rating System: Reserved for non-U.S. &
      non-Canadian system \n");
      ratesys = 6;
      break;
}

if( ratesys==0 || ratesys==2)  // MPAA system
{
   switch(srbuf[0] & 0x07)
   {
      case 0:
         printf("Rating: N/A \n");
         break;
      case 1:
         printf("Rating: G \n");
         break;
      case 2:
         printf("Rating: PG \n");
         break;
      case 3:
```

```
      printf("Rating: PG-13 \n");
      break;
    case 4:
      printf("Rating: R \n");
      break;
    case 5:
      printf("Rating: NC-17 \n");
      break;
    case 6:
      printf("Rating: X \n");
      break;
    case 7:
      printf("Rating: Not Rated \n");
      break;
  }
}
else
{
  if( ratesys==1)       // TVPG system
  {
    switch(srbuf[1] & 0x07)
    {
      case 0:
      printf("Rating: None ");
        DispContent(srbuf[0], srbuf[1]);
        break;
      case 1:
        printf("Rating: TV-Y ");
        DispContent(srbuf[0], srbuf[1]);
```

```
            break;
         case 2:
            printf("Rating: TV-Y7 ");
            DispContent(srbuf[0], srbuf[1]);
            break;
         case 3:
            printf("Rating: TV-G ");
            DispContent(srbuf[0], srbuf[1]);
            break;
         case 4:
            printf("Rating: TV-PG ");
            DispContent(srbuf[0], srbuf[1]);
            break;
         case 5:
            printf("Rating: TV-14 ");
            DispContent(srbuf[0], srbuf[1]);
            break;
         case 6:
            printf("Rating: TV-MA ");
            DispContent(srbuf[0], srbuf[1]);
            break;
         case 7:
            printf("Rating: None ");
            DispContent(srbuf[0], srbuf[1]);
            break;
      }
   }
   printf("\n");
}
```

```
         if((srbuf[2] & 0x40) == 0x40)
            printf("Vchip Block: Yes \n");
         else
            printf("Vchip Block: No \n");

         if((srbuf[2] & 0x20) == 0x20)
            printf("New Updated Vchip Data: Yes \n");
         else
            printf("New Updated Vchip Data: No \n");

      }
      if( kbhit() )
         if(getch() == 27)
            out1 = 0;
   }
}
```

Example Software Driver in Assembly

The following driver routines were written in 89C00 DSP assembly to support ZiLOG’s eZVision 300 series TV controllers. Those routines can be used individually to send commands to the Z86130/Z86230 and read V-Chip data from the Z86130/Z86230. These routines can also be used as part of the I^{2}C driver routines in the I^{2}C scheduler to send continuous data and refresh those I^{2}C devices. For details regarding the I^{2}C scheduler and drivers, please refer to “i2c_act.asm” in the Z903xx demo project files (apic_36a.zip). This file can be requested from the local ZiLOG Sales Office or e-mail directly to achang@zilog.com.

```
;***********************************************************
; VCHIP Z86130/Z86230 I2C routines

;

; Z86130W_ REFRESH: to write preset rating values to
; internal registers, 08h, 09h, 0Ah, 0Bh, 0Eh
;

;***********************************************************

Z86130W_REFRESH:

; Execute this subroutine once every field.

   LD      A, I2C_DEVICE

   AND     A, #I2C_SCH_ON

   JP      Z, DEVICE_SKIP       ; Not a new field skip
                                ; Z86130_REFRESH.


   LD      A, I2CM_STATUS       ; Check if I2C busy.

   AND     A, #I2C_BUSY         ; MSB = "busy bit"

   JP      NZ, DEVICE_EXIT      ; Yes, exit but do not inc.
                                ; I2C device.


   CALL    I2C_M2_ACTIVATE      ; Switch to i2c master 2
   LD       A, ZVCHIP_BCTL
   AND      A, #Z86130_CYL_MASK
   ADD      A, #Z86130WR_TABLE
   LD       D0:1, A
   LD       A, ZVCHIP_BCTL
   ADD      A, #1
   LD       ZVCHIP_BCTL, A
   LD       A, @D0:1
   LD       PC, A                   ; jump to those routines
                                    ; based on the following
                                    ; table

;=========================================
; Table:
Z86130WR_TABLE:
   DW       ZWR_C8
   DW       ZWR_C9
   DW       ZWR_CA
   DW       ZWR_CB
   DW       ZWR_CE
;=========================================

ZWR_C8:
;  LD       A, #%C83C
   LD       A, TV_RATING1
   CALL     SHIFT_RIGHT_8
   AND      A, #%00FF
   OR       A, #%C800
   JP       Z86130_SD_2BYTE
```

```
ZWR_C9:
;  LD      A, #%C93E
   LD      A, TV_RATING1
   AND     A, #%00FF
   OR      A, #%C900
   JP      Z86130_SD_2BYTE

ZWR_CA:
;  LD      A, #%CAF7
   LD      A, TV_RATING2
   CALL    SHIFT_RIGHT_8
   AND     A, #%00FF
   OR      A, #%CA00
   JP      Z86130_SD_2BYTE

ZWR_CB:
;  LD      A, #%CB73
   LD      A, TV_RATING2
   AND     A, #%00FF
   OR      A, #%CB00
   JP      Z86130_SD_2BYTE

ZWR_CE:
   LD      A, ZVCHIP_BCTL
   AND     A, #~(Z86130_CYL_MASK); reset counter
   LD      ZVCHIP_BCTL, A
;  LD      A, #%CEC0
   LD      A, ZVCHIP_BCTL
   CALL    SHIFT_RIGHT_8
   AND     A, #%00FF
   OR      A, #%CE00
Z86130_SD_2BYTE:
   LD      I2C_DATA_1 ,A
   CALL    I2C_M2_ACTIVATE         ; Switch to I2C master 2
   LD      A, #(2*BYTE_NUMBER_M) | Z86130_ADDR
   CALL    SEND_I2C
   JP      DEVICE_SKIP

;**********************************************************
; VCHIP Z86130 I2C read Serial Status Register routines here
;**********************************************************
Z86130RSSR_REFRESH:
; Execute this subroutine once every field.
   LD      A, I2C_DEVICE
   AND     A, #I2C_SCH_ON
   JP      Z, DEVICE_SKIP

   LD      A, I2CM_STATUS        ; Check if I2C busy.
   AND     A, #I2C_BUSY          ; MSB = "busy bit"
   JP      NZ, DEVICE_EXIT       ; Yes, exit but do not
                                 ; inc. I2C device.

   CALL    I2C_M2_ACTIVATE

   LD      A, #(1*BYTE_NUMBER_M | Z86130_ADDR)
   CALL    RECEIVE_I2C           ; Read data from eeprom
   LD      A, I2C_DATA_1          ; first byte is the upper
                                  ; byte
   AND     A, #%FF00
   AND     A, #Z130_RDY           ; check RYD bit from Z86130
                                  ; SS Register
   JP      Z, Z86130W_NOTREADY    ; Not a ready yet, skip
                                  ; Z86130_REFRESH.
   LD      A, I2C_DATA_1          ; first byte is upper byte
   AND     A, #%FF00
   AND     A, #Z130_VLOCK         ; check video lock
   JP      Z, Z86130W_NOTREADY    ; Not a ready yet, skip
                                  ; Z86130_REFRESH.

   LD      A, BG_flags
   OR      A, #ZVC_RDY_W          ; Now it is ready to write
   LD      BG_flags, A
   JP      DEVICE_SKIP

Z86130W_NOTREADY:
   LD      A, BG_flags
   AND     A, #~(ZVC_RDY_W)       ; It is not ready to write
   LD      BG_flags, A
   JP      DEVICE_SKIP
```

Summary

Applying the eZSelect Z86130/Z86230 to PIP circuits is just one of many applications for ZiLOG's eZSelect VBI decoder devices. The eZSelect Z86130/ Z86230 has the intelligence to decode the Program Rating information automatically and block the screen directly using one output pin (PB). Therefore, the Z86130/Z86230 can be used in PIP circuits and can also be designed into the main TV chassis, Stand-alone Set-Top Box (STB), PC monitor displaying TV signals, or any other applications that need program rating data decoding. ZiLOG's eZSelect Z86130/Z86230 is proven to be an easy-to-use, low-cost data decoding device.