Freescale Semiconductor
Application Note

Document Number:AN4635
Rev. 0, 11/2012

Using the Vybrid TCON Module

by: Melissa Hunter

Contents

1 Purpose and overview

The timing controller (TCON) module on the Vybrid processors provides an alternative to the DCU for generating control signals to a TFT panel. The module was originally designed to generate timing signals for raw TFT panels that do not include an embedded timing controller. These panels are few and far between, but the TCON can be useful for generating control signals to traditional TFT panels that do include a timing controller.

This application note walks through an example of generating an inverted data enable (DE) signal for a TFT panel in order to demonstrate how the TCON module might be used to avoid adding external logic to interface to a TFT panel.

2 TCON Features

The TCON module includes up to 12 external timing channel signals that can be generated from a variety of sources. TCON features include:

- Bypass mode where DCU signals are passed through the TCON
- 4 comparators that can compare in the horizontal or vertical directions
- 6 pulse generators with independent set and reset points
- 4 toggle generator signals

- 14 signal mixer channels that can be used to combine signals within the TCON
- Option to swap color channels and relocate the pixel clock

3 Using the TCON in Bypass Mode

All applications that use the DCU are required to enable the TCON module. This is because the DCU signals are not output directly to the processor pins. Instead all of the DCU signals go through the TCON module, and the TCON module drives the external signals. Therefore at minimum if you want to use the DCU to interface to a TFT panel with no intervention by the TCON module, then the TCON module clock gate should be enabled and the TCON_CTRL1[TCON_BYPASS] bit set to configure the TCON in bypass mode:

```
TCON1->CTRL1 |= TCON_CTRL1_TCON_BYPASS_MASK;
```

The code above is required in order to get the DCU signals to the pins.

Let's start by taking a look at the control signals that are output by the DCU where the TCON is in bypass mode. For purposes of this document, the panel has the following parameters:

- Panel height = 272 pixels
- Panel width = 480 pixels
- Pixel clock frequency = 4.04 MHz
- Vsync and Hsync both active low (inverted). **DE will also need to be active low!**
- Vsync front porch = 10 lines, Vsync width = 10 lines, Vsync back porch = 2 lines
- Hsync front porch = 0x29 clocks, Hsync width = 41 clocks, Hsync back porch = 2 clocks

The figures below show the control signals output from the DCU with the TCON in bypass mode. The DCU includes bits that control the polarity of the HSYNC and VSYNC signals, but the DCU doesn't include a similar bit to control the polarity of the DE signal. In the figures below, the DCU has been configured to use active low polarity for both HSYNC and VSYNC signals, but the DE signal is active high.

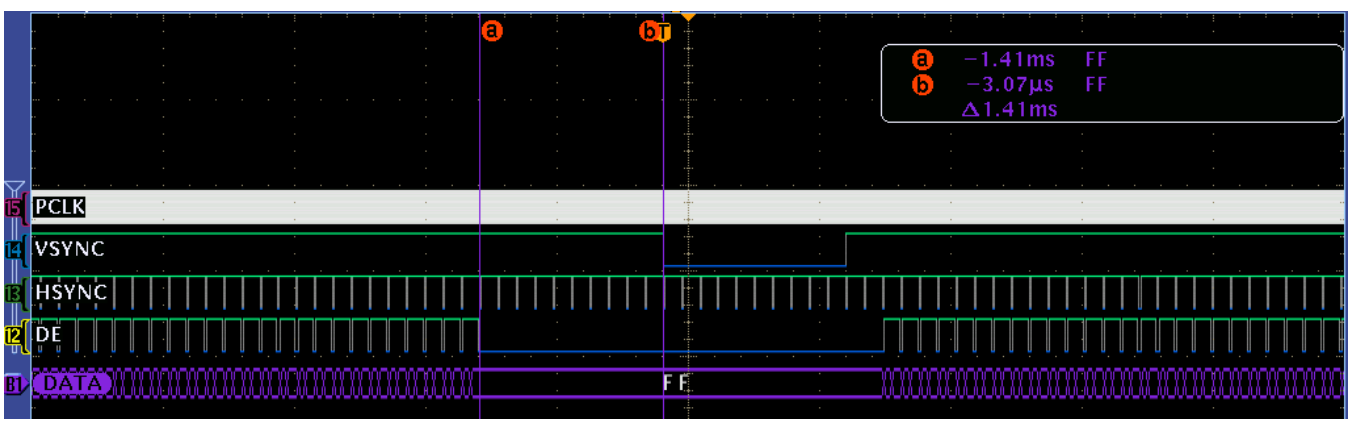

Figure 1. Bypass mode vertical timing

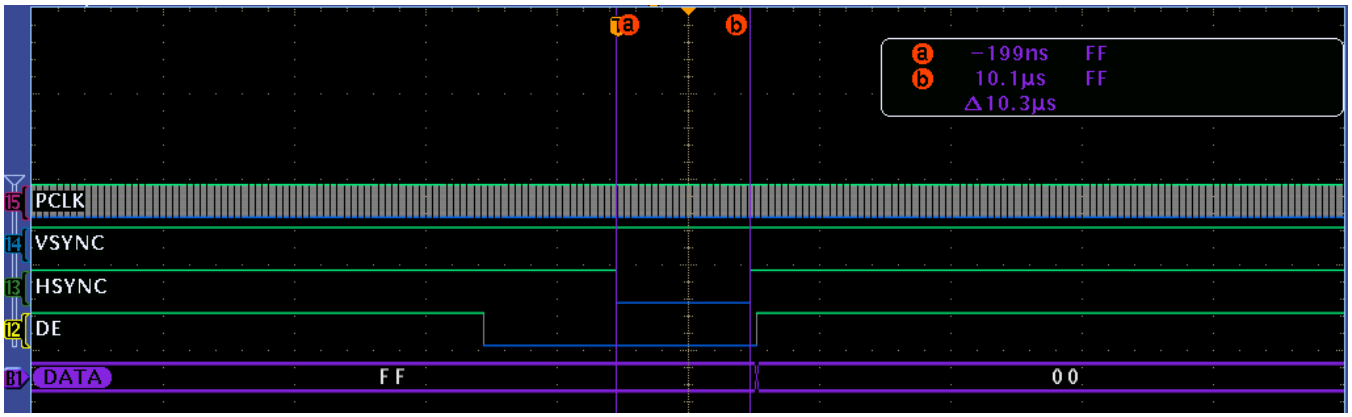

Figure 2. Bypass mode horizontal timing

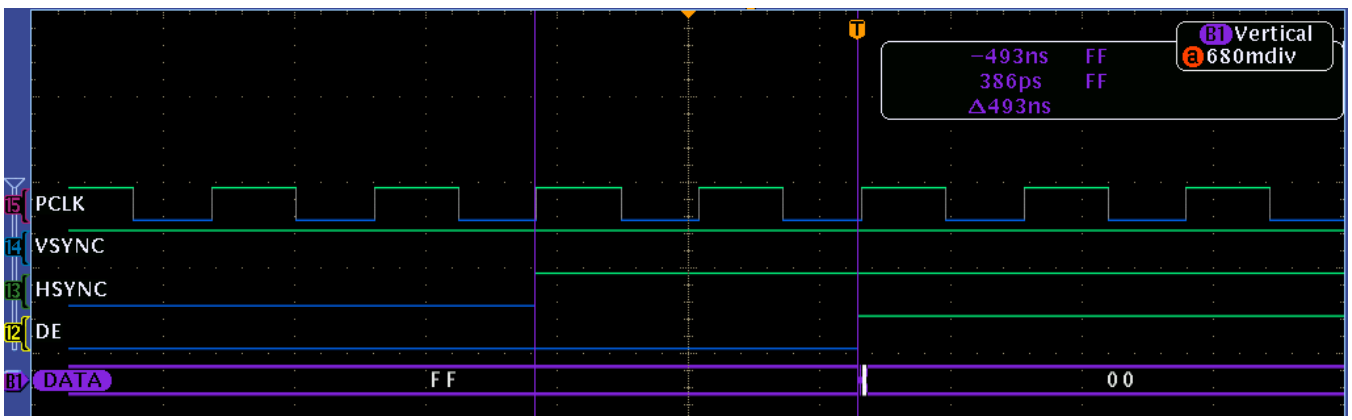

Figure 3. Bypass mode DE assertion

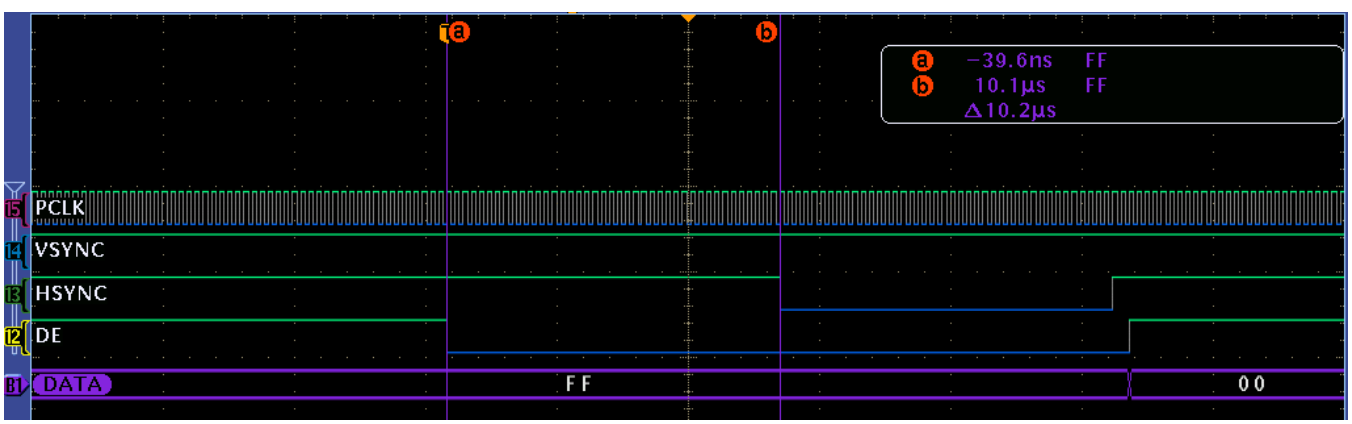

Figure 4. Bypass mode DE negation

4 Using TCON Mode

From the bypass mode figures above, you can see that the DE signal is active high. Our goal is to have an active low version of the DE signal. The TCON module can be used to solve this problem without requiring any external logic.

When the TCON is enabled, we will lose all of the DCU signals. The TCON will need to be configured to generate all of the control signals at this point. The following sections will show how the TCON module can be used to recreate the HSYNC and VSYNC signals so that they match up to the signals from the DCU. Then the final step will be configuring the TCON to generate the active low DE signal.

4.1 Regenerating HSYNC

The HSYNC signal can be recreated using one of the TCON's pulse generators. Each of the pulse generator channels has programmable set and reset points that can be compared to an hcount or a vcount value. In order to generate the HSYNC signal the hcount option is used.

For every line the hcount counts the current pixel number, from one to the number of pixels per line (the full line, including the HSYNC's front porch, pulse width, and back porch). The counter resets on the rising edge of HSYNC and increments on each pixel clock.

4.1.1 Configuring the pulse generator

Because the HSYNC signal we are generating is active low, the set point will determine when the signal negates, and the reset point will determine when the signal asserts. In most cases the following code will correctly configure pulse generator 1 to create the HSYNC signal:

```
TCON1->PULSE1 = TCON_PULSE0_SET(0)
                    | TCON_PULSE0_RESET(HSYNC_BP + PANEL_X + HSYNC_FP);
```

This code causes the HSYNC signal to go high at the same point as the HSYNC from the DCU. Then the signal will remain high through the HSYNC back porch period, the active line time (PANEL_X), and the HSYNC front porch period. The difference between the set and reset points is equal to the HSYNC pulse width.

4.1.2 Pulse generator configuration when HSYNC pulse width is one clock

Internal to the TCON module there is a two pixel clock delay between the pulse generator and the output of the TCON signal mux. The pulse generator and pixel data both incur a two clock delay as they pass through the TCON, so in cases where the HSYNC pulse width is at least two pixel clocks the code above applies. If there is a case where the HSYNC pulse width is only one pixel clock, then internal delay creates a special case that uses a different configuration. If the HSYNC pulse width is one clock, then the configuration below is used instead:

```
TCON1->PULSE1 = TCON_PULSE0_SET(HSYNC_BP + PANEL_X + HSYNC_FP)
                    | TCON_PULSE0_RESET(HSYNC_BP + PANEL_X + HSYNC_FP - 1);
```

4.2 Regenerating VSYNC

A different pulse generator output will be used to create the VSYNC. In order to get the correct signal timing for VSYNC, the pulse generator will be configured for function 1 mode. In this mode the vcount value is compared to the set and reset points, but the placement of the signal edge within the line is determined by one of the TCON's comparators.

4.2.1 Configuring the VSYNC comparator

Because of the delays within the TCON module, the VSYNC and HSYNC edges will occur in different places if the regular vertical compare mode for the pulse generator is used (PULSEn[FUNC_SEL] = 2). Instead the FUNC_SEL = 1 mode is used. This mode allows a comparator to be configured to place the VSYNC edge at the same point as the HSYNC edge, as shown below:

```
#if (HSYNC_PW == 1)
      TCON1->COMP0 = HSYNC_BP + PANEL_X + HSYNC_FP - 1;
#else
      TCON1->COMP0 = HSYNC_BP + PANEL_X + HSYNC_FP;
#endif
```

The comparator is configured to trigger at the same point that was used as the reset for the HSYNC signal. This will keep the HSYNC and VSYNC assertions lined up to the same pixel clock edge. Note that if the HSYNC pulse width is one clock it is still a special case that is treated slightly differently.

4.2.2 Configuring the VSYNC pulse generator

The pulse generator configuration is similar to what was done to generate the HSYNC pulse. Because the comparator is being used to set the transition point for the signal near the end of the horizontal line, the set and reset points for the VSYNC are configured to take place one line before where you might expect.

Here is the pulse generator configuration for VSYNC:

```
TCON1->PULSE0 = TCON_PULSE0_FUNC_SEL(1)
                  | TCON_PULSE0_SET(VSYNC_PW - 1)
                  | TCON_PULSE0_COMPARATOR_SEL(0)
                  | TCON_PULSE0_RESET(VSYNC_PW -1 + VSYNC_BP + PANEL_Y + VSYNC_FP);
```

The pulse generator is configured in function 1 mode where comparator 0 determines the signal transition point within a horizontal line. Notice that both the set and reset point are decremented by one. This is because the signal edge needs to move back a few pixel clock cycles to line up with VSYNC. To do this the set and reset points are decremented, then the comparator moves the signal transition near the end of the horizontal line.

4.3 Creating an active low data enable signal

Now that HSYNC and VSYNC are running, it is time to generate the active low data enable signal.

4.3.1 Generate the basic DE pulse

Generating the DE pulse is very similar to generating the HSYNC pulse. Again, we will use a pulse generator channel configured for hcount compares:

```
#if (HSYNC_PW == 1)
      TCON1->PULSE2 = TCON_PULSE0_SET(HSYNC_BP - 1)
                    | TCON_PULSE0_RESET(HSYNC_BP - 1 + PANEL_X);
#else
      TCON1->PULSE2 = TCON_PULSE0_SET(HSYNC_BP)
                    | TCON_PULSE0_RESET(HSYNC_BP + PANEL_X);
```

Looking at the typical case where the HSYNC pulse width is greater than one, the signal will go high when the HSYNC back porch expires. Then the signal goes low before the HSYNC front porch starts.

At this point we have a signal very similar to the original active high DE signal that was generated by the DCU. There are two issues that need to be resolved. First, the signal we have now is asserting during the vertical blanking period for the panel. Secondly, the signal is active high when we want an active low signal.

4.3.2 Generate a keep-out pulse

In order to prevent our new DE signal from asserting during the vertical blanking period, we'll setup yet another pulse generator signal. This signal will be configured to assert at the start of the VSYNC front porch and negate at the end of the VSYNC back porch. Later, this keep-out pulse will be combined with the basic DE pulse to generate the active low DE signal.

To generate the keep-out signal a pulse generator will be configured to compare the set and reset points to the vcount. This time around we don't need to use a comparator to place the edge at an exact point within the line, so pulse generator functional mode 2 is used:

```
TCON1->PULSE3 = TCON_PULSE0_FUNC_SEL(2)
                  | TCON_PULSE0_SET(VSYNC_PW + VSYNC_BP)
                  | TCON_PULSE0_RESET(VSYNC_PW + VSYNC_BP + PANEL_Y);
```

This will configure the signal to go high at the end of the VSYNC back porch and go low when the VSYNC front porch starts. The end result is a low pulse during the vertical blanking time.

4.3.3 Combine the two signals

To get the final active low DE signal, we will combine the basic DE and keep-out pulses using one of the TCON's signal mixer channels. The signal mixer channels can be used to perform logic operations between two inputs, where the inputs can be almost any of the TCON generated signals.

Signal mixer channel six is the first of the signal mixer channels that can be output to an external pin, so channel six is used:

```
TCON1->SMX6 = TCON_SMX0_X_SEL(0x4) /* Use PULSE2 as the X input */
                  | TCON_SMX0_Y_SEL(0x5) /* Use PULSE3 as the Y input */
                  | TCON_SMX0_INDEX0_SEL(0x6) /* output the !(X&Y) value */
                  | TCON_SMX0_INDEX1_SEL(0x6) /* output the !(X&Y) value */
                  | TCON_SMX0_INDEX2_SEL(0x6) /* output the !(X&Y) value */
                  | TCON_SMX0_INDEX3_SEL(0x6); /* output the !(X&Y) value */
```

This code configures the signal mixer to use PULSE2 (the basic active high DE pulse) as input X and PULSE3 (the keep-out pulse) as input Y. The signal mixer is capable of cycling through different logic combinations of the signals. In this case we always want to output !(X&Y), so each of the indexes is set to the same value to keep the logical operation constant.

The figure below shows all of the control signals as generated by the TCON module. The basic DE pulse and the keep-out pulse would not actually be connected to a panel, but they have been added to the diagram to show the components used to generate the active low DE signal. The timings for the TCON generated signals match up to the timings when the TCON is in bypass mode, but now the polarity of the DE signal is active low.

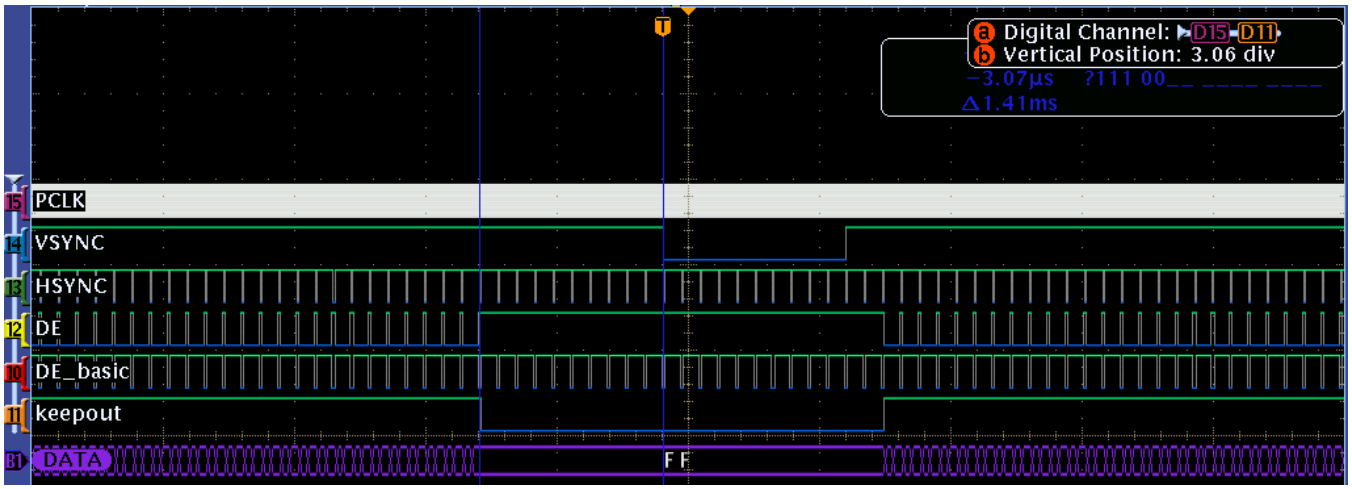

Figure 5. Control signals generated by TCON module

5 Conclusion

This application note has walked through how the TCON module can be used to generate a LCD panel control signal that might not be available directly from the DCU module. This is just one example of how the TCON could be used. In almost any situation where an LCD panel is being used that needs a control signal that cannot be generated using the DCU, the TCON could be used to generate the required signal or signals helping to eliminate the need for external logic for the LCD panel digital interface.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: AN4635
Rev. 0, 11/2012