PVR350 Video decoder registers 0x02002800 -> 0x02002B00
==========

This list has been worked out through trial and error. There will be mistakes and omissions. Some registers have no obvious effect so it's hard to say what they do, while others interact with each other, or require a certain load sequence. Horizontal filter setup is one example, with six registers working in unison and requiring a certain load sequence to correctly configure. The indexed colour palette is much easier to set at just two registers, but again it requires a certain load sequence.

Some registers are fussy about what they are set to. Load in a bad value & the decoder will fail. A firmware reload will often recover, but sometimes a reset is required. For registers containing size information, setting them to 0 is generally a bad idea. For other control registers i.e. 2878, you'll only find out what values are bad when it hangs.

2800

bit 0
Decoder enable
0 = disable
1 = enable

2804

bits 0:31
Decoder horizontal Y alias register 1

2808

bits 0:31
Decoder horizontal Y alias register 2

280C

bits 0:31
Decoder horizontal Y alias register 3

2810

bits 0:31
Decoder horizontal Y alias register 4

2814

bits 0:31
Decoder horizontal Y alias register 5

2818

bits 0:31
Decoder horizontal Y alias trigger

These six registers control the horizontal aliasing filter for the Y plane. The first five registers must all be loaded before accessing the trigger (2818), as this register actually clocks the data through for the first five.

To correctly program set the filter, this whole procedure must be done 16 times. The actual register contents are copied from a lookup-table in the firmware which contains 4 different filter settings.

281C
      bits 0:31
        Decoder horizontal UV alias register 1

2820
      bits 0:31
        Decoder horizontal UV alias register 2

2824
      bits 0:31
        Decoder horizontal UV alias register 3

2828
      bits 0:31
        Decoder horizontal UV alias register 4

282C
      bits 0:31
        Decoder horizontal UV alias register 5

2830
      bits 0:31
        Decoder horizontal UV alias trigger

     These six registers control the horizontal aliasing for the UV plane.
     Operation is the same as the Y filter, with 2830 being the trigger
     register.

2834
      bits 0:15
        Decoder Y source width in pixels

      bits 16:31
        Decoder Y destination width in pixels

2838
      bits 0:15
        Decoder UV source width in pixels

      bits 16:31
        Decoder UV destination width in pixels

     NOTE: For both registers, the resulting image must be fully visible on
     screen. If the image exceeds the right edge both the source and destination
     size must be adjusted to reflect the visible portion. For the source width,
     you must take into account the scaling when calculating the new value.

283C
      bits 0:31
        Decoder Y horizontal scaling
                    Normally = Reg 2854 >> 2

2840
bits 0:31
Decoder ?? unknown - horizontal scaling
Usually 0x00080514

2844
bits 0:31
Decoder UV horizontal scaling
Normally = Reg 2854 >> 2

2848
bits 0:31
Decoder ?? unknown - horizontal scaling
Usually 0x00100514

284C
bits 0:31
Decoder ?? unknown - Y plane
Usually 0x00200020

2850
bits 0:31
Decoder ?? unknown - UV plane
Usually 0x00200020

2854
bits 0:31
Decoder 'master' value for horizontal scaling

2858
bits 0:31
Decoder ?? unknown
Usually 0

285C
bits 0:31
Decoder ?? unknown
Normally = Reg 2854 >> 1

2860
bits 0:31
Decoder ?? unknown
Usually 0

2864
bits 0:31
Decoder ?? unknown
Normally = Reg 2854 >> 1

2868
bits 0:31
Decoder ?? unknown
Usually 0

Most of these registers either control horizontal scaling, or appear linked to it in some way. Register 2854 contains the 'master' value & the other

registers can be calculated from that one. You must also remember to correctly set the divider in Reg 2874.

To enlarge:
Reg 2854 = (source_width * 0x00200000) / destination_width
Reg 2874 = No divide

To reduce from full size down to half size:
Reg 2854 = (source_width/2 * 0x00200000) / destination width
Reg 2874 = Divide by 2

To reduce from half size down to quarter size:
Reg 2854 = (source_width/4 * 0x00200000) / destination width
Reg 2874 = Divide by 4

The result is always rounded up.

286C

bits 0:15
Decoder horizontal Y buffer offset

bits 15:31
Decoder horizontal UV buffer offset

Offset into the video image buffer. If the offset is gradually incremented, the on screen image will move left & wrap around higher up on the right.

2870

bits 0:15
Decoder horizontal Y output offset

bits 16:31
Decoder horizontal UV output offset

Offsets the actual video output. Controls output alignment of the Y & UV planes. The higher the value, the greater the shift to the left. Use reg 2890 to move the image right.

2874

bits 0:1
Decoder horizontal Y output size divider
00 = No divide
01 = Divide by 2
10 = Divide by 3

bits 4:5
Decoder horizontal UV output size divider
00 = No divide
01 = Divide by 2
10 = Divide by 3

bit 8
Decoder ?? unknown

```
          0 = Normal
          1 = Affects video output levels

      bit 16
        Decoder ?? unknown
          0 = Normal
          1 = Disable horizontal filter

--------------------------------------------------------------------------------
2878
      bit 0
        ?? unknown

      bit 1
        osd on/off
          0 = osd off
          1 = osd on

      bit 2
        Decoder + osd video timing
          0 = NTSC
          1 = PAL

      bits 3:4
        ?? unknown

      bit 5
        Decoder + osd
          Swaps upper & lower fields

--------------------------------------------------------------------------------
287C
      bits 0:10
        Decoder & osd ?? unknown
          Moves entire screen horizontally. Starts at 0x005 with the screen
          shifted heavily to the right. Incrementing in steps of 0x004 will
          gradually shift the screen to the left.

      bits 11:31
        ?? unknown

     Normally contents are 0x00101111 (NTSC) or 0x1010111d (PAL)

--------------------------------------------------------------------------------
2880  --------    ?? unknown
2884  --------    ?? unknown
--------------------------------------------------------------------------------
2888
      bit 0
        Decoder + osd ?? unknown
          0 = Normal
          1 = Misaligned fields (Correctable through 289C & 28A4)

      bit 4
        ?? unknown
```

bit 8
?? unknown

Warning: Bad values will require a firmware reload to recover.
Known to be bad are 0x000,0x011,0x100,0x111

288C

bits 0:15
osd ?? unknown
Appears to affect the osd position stability. The higher the value the more unstable it becomes. Decoder output remains stable.

bits 16:31
osd ?? unknown
Same as bits 0:15

2890

bits 0:11
Decoder output horizontal offset.

Horizontal offset moves the video image right. A small left shift is possible, but it's better to use reg 2870 for that due to its greater range.

NOTE: Video corruption will occur if video window is shifted off the right edge. To avoid this read the notes for 2834 & 2838.

2894

bits 0:23
Decoder output video surround colour.

Contains the colour (in yuv) used to fill the screen when the video is running in a window.

2898

bits 0:23
Decoder video window colour
Contains the colour (in yuv) used to fill the video window when the video is turned off.

bit 24
Decoder video output
0 = Video on
1 = Video off

bit 28
Decoder plane order
0 = Y,UV
1 = UV,Y

bit 29
Decoder second plane byte order
0 = Normal (UV)
1 = Swapped (VU)

In normal usage, the first plane is Y & the second plane is UV. Though the order of the planes can be swapped, only the byte order of the second plane can be swapped. This isn't much use for the Y plane, but can be useful for the UV plane.

289C

bits 0:15
  Decoder vertical field offset 1

bits 16:31
  Decoder vertical field offset 2

Controls field output vertical alignment. The higher the number, the lower the image on screen. Known starting values are 0x011E0017 (NTSC) & 0x01500017 (PAL)

28A0

bits 0:15
  Decoder & osd width in pixels

bits 16:31
  Decoder & osd height in pixels

All output from the decoder & osd are disabled beyond this area. Decoder output will simply go black outside of this region. If the osd tries to exceed this area it will become corrupt.

28A4

bits 0:11
  osd left shift.

Has a range of 0x770->0x7FF. With the exception of 0, any value outside of this range corrupts the osd.

28A8

bits 0:15
  osd vertical field offset 1

bits 16:31
  osd vertical field offset 2

Controls field output vertical alignment. The higher the number, the lower the image on screen. Known starting values are 0x011E0017 (NTSC) & 0x01500017 (PAL)

28AC -------- ?? unknown
 |
 V
28BC -------- ?? unknown

28C0

bit 0
  Current output field
    0 = first field
    1 = second field

```
      bits 16:31
        Current scanline
          The scanline counts from the top line of the first field
          through to the last line of the second field.
-------------------------------------------------------------------------------
28C4  --------    ?? unknown
 |
 V
28F8  --------    ?? unknown
-------------------------------------------------------------------------------
28FC
      bit 0
        ?? unknown
          0 = Normal
          1 = Breaks decoder & osd output
-------------------------------------------------------------------------------
2900
      bits 0:31
        Decoder vertical Y alias register 1
---------------
2904
      bits 0:31
        Decoder vertical Y alias register 2
---------------
2908
      bits 0:31
        Decoder vertical Y alias trigger

     These three registers control the vertical aliasing filter for the Y plane.
     Operation is similar to the horizontal Y filter (2804). The only real
     difference is that there are only two registers to set before accessing
     the trigger register (2908). As for the horizontal filter, the values are
     taken from a lookup table in the firmware, and the procedure must be
     repeated 16 times to fully program the filter.
-------------------------------------------------------------------------------
290C
      bits 0:31
        Decoder vertical UV alias register 1
---------------
2910
      bits 0:31
        Decoder vertical UV alias register 2
---------------
2914
      bits 0:31
        Decoder vertical UV alias trigger

     These three registers control the vertical aliasing filter for the UV
     plane. Operation is the same as the Y filter, with 2914 being the trigger.
-------------------------------------------------------------------------------
2918
      bits 0:15
        Decoder Y source height in pixels

      bits 16:31
```

Decoder Y destination height in pixels

291C

bits 0:15
Decoder UV source height in pixels divided by 2

bits 16:31
Decoder UV destination height in pixels

NOTE: For both registers, the resulting image must be fully visible on screen. If the image exceeds the bottom edge both the source and destination size must be adjusted to reflect the visible portion. For the source height, you must take into account the scaling when calculating the new value.

2920

bits 0:31
Decoder Y vertical scaling
Normally = Reg 2930 >> 2

2924

bits 0:31
Decoder Y vertical scaling
Normally = Reg 2920 + 0x514

2928

bits 0:31
Decoder UV vertical scaling
When enlarging = Reg 2930 >> 2
When reducing = Reg 2930 >> 3

292C

bits 0:31
Decoder UV vertical scaling
Normally = Reg 2928 + 0x514

2930

bits 0:31
Decoder 'master' value for vertical scaling

2934

bits 0:31
Decoder ?? unknown - Y vertical scaling

2938

bits 0:31
Decoder Y vertical scaling
Normally = Reg 2930

293C

bits 0:31
Decoder ?? unknown - Y vertical scaling

2940

bits 0:31
Decoder UV vertical scaling

When enlarging = Reg 2930 >> 1
When reducing = Reg 2930

2944

bits 0:31
Decoder ?? unknown - UV vertical scaling

2948

bits 0:31
Decoder UV vertical scaling
Normally = Reg 2940

294C

bits 0:31
Decoder ?? unknown - UV vertical scaling

Most of these registers either control vertical scaling, or appear linked to it in some way. Register 2930 contains the 'master' value & all other registers can be calculated from that one. You must also remember to correctly set the divider in Reg 296C

To enlarge:
Reg 2930 = (source_height * 0x00200000) / destination_height
Reg 296C = No divide

To reduce from full size down to half size:
Reg 2930 = (source_height/2 * 0x00200000) / destination height
Reg 296C = Divide by 2

To reduce from half down to quarter.
Reg 2930 = (source_height/4 * 0x00200000) / destination height
Reg 296C = Divide by 4

2950

bits 0:15
Decoder Y line index into display buffer, first field

bits 16:31
Decoder Y vertical line skip, first field

2954

bits 0:15
Decoder Y line index into display buffer, second field

bits 16:31
Decoder Y vertical line skip, second field

2958

bits 0:15
Decoder UV line index into display buffer, first field

bits 16:31
Decoder UV vertical line skip, first field

295C

bits 0:15
  Decoder UV line index into display buffer, second field

bits 16:31
  Decoder UV vertical line skip, second field

2960

bits 0:15
  Decoder destination height minus 1

bits 16:31
  Decoder destination height divided by 2

2964

bits 0:15
  Decoder Y vertical offset, second field

bits 16:31
  Decoder Y vertical offset, first field

These two registers shift the Y plane up. The higher the number, the greater the shift.

2968

bits 0:15
  Decoder UV vertical offset, second field

bits 16:31
  Decoder UV vertical offset, first field

These two registers shift the UV plane up. The higher the number, the greater the shift.

296C

bits 0:1
  Decoder vertical Y output size divider
    00 = No divide
    01 = Divide by 2
    10 = Divide by 4

bits 8:9
  Decoder vertical UV output size divider
    00 = No divide
    01 = Divide by 2
    10 = Divide by 4

2970

bit 0
  Decoder ?? unknown
    0 = Normal
    1 = Affect video output levels

bit 16
  Decoder ?? unknown
    0 = Normal
    1 = Disable vertical filter

```
--------------------------------------------------------------------------------
2974  --------   ?? unknown
 |
 V
29EF  --------   ?? unknown
--------------------------------------------------------------------------------
2A00
      bits 0:2
        osd colour mode
          000 = 8 bit indexed
          001 = 16 bit (565)
          010 = 15 bit (555)
          011 = 12 bit (444)
          100 = 32 bit (8888)

      bits 4:5
        osd display bpp
          01 = 8 bit
          10 = 16 bit
          11 = 32 bit

      bit 8
        osd global alpha
          0 = Off
          1 = On

      bit 9
        osd local alpha
          0 = Off
          1 = On

      bit 10
        osd colour key
          0 = Off
          1 = On

      bit 11
        osd ?? unknown
          Must be 1

      bit 13
        osd colour space
          0 = ARGB
          1 = AYVU

      bits 16:31
        osd ?? unknown
          Must be 0x001B (some kind of buffer pointer ?)

    When the bits-per-pixel is set to 8, the colour mode is ignored and
    assumed to be 8 bit indexed. For 16 & 32 bits-per-pixel the colour depth
    is honoured, and when using a colour depth that requires fewer bytes than
    allocated the extra bytes are used as padding. So for a 32 bpp with 8 bit
    index colour, there are 3 padding bytes per pixel. It's also possible to
    select 16bpp with a 32 bit colour mode. This results in the pixel width
```

being doubled, but the color key will not work as expected in this mode.

Colour key is as it suggests. You designate a colour which will become completely transparent. When using 565, 555 or 444 colour modes, the colour key is always 16 bits wide. The colour to key on is set in Reg 2A18.

Local alpha works differently depending on the colour mode. For 32bpp & 8 bit indexed, local alpha is a per-pixel 256 step transparency, with 0 being transparent and 255 being solid. For the 16bpp modes 555 & 444, the unused bit(s) act as a simple transparency switch, with 0 being solid & 1 being fully transparent. There is no local alpha support for 16bit 565.

Global alpha is a 256 step transparency that applies to the entire osd, with 0 being transparent & 255 being solid.

It's possible to combine colour key, local alpha & global alpha.

2A04

bits 0:15
osd x coord for left edge

bits 16:31
osd y coord for top edge

2A08

bits 0:15
osd x coord for right edge

bits 16:31
osd y coord for bottom edge

For both registers, (0,0) = top left corner of the display area. These registers do not control the osd size, only where it's positioned & how much is visible. The visible osd area cannot exceed the right edge of the display, otherwise the osd will become corrupt. See reg 2A10 for setting osd width.

2A0C

bits 0:31
osd buffer index

An index into the osd buffer. Slowly incrementing this moves the osd left, wrapping around onto the right edge

---

2A10

bits 0:11
osd buffer 32 bit word width

Contains the width of the osd measured in 32 bit words. This means that all colour modes are restricted to a byte width which is divisible by 4.

2A14

bits 0:15
osd height in pixels

bits 16:32

```
        osd line index into buffer
          osd will start displaying from this line.
--------------------------------------------------------------------------------
2A18
      bits 0:31
        osd colour key

     Contains the colour value which will be transparent.
--------------------------------------------------------------------------------
2A1C
      bits 0:7
        osd global alpha

     Contains the global alpha value (equiv ivtvfbctl --alpha XX)
--------------------------------------------------------------------------------
2A20  --------    ?? unknown
  |
  V
2A2C  --------    ?? unknown
--------------------------------------------------------------------------------
2A30
      bits 0:7
        osd colour to change in indexed palette
---------------
2A34
      bits 0:31
        osd colour for indexed palette

     To set the new palette, first load the index of the colour to change into
     2A30, then load the new colour into 2A34. The full palette is 256 colours,
     so the index range is 0x00-0xFF
--------------------------------------------------------------------------------
2A38  --------    ?? unknown
2A3C  --------    ?? unknown
--------------------------------------------------------------------------------
2A40
      bits 0:31
        osd ?? unknown

     Affects overall brightness, wrapping around to black
--------------------------------------------------------------------------------
2A44
      bits 0:31
        osd ?? unknown

     Green tint
--------------------------------------------------------------------------------
2A48
      bits 0:31
        osd ?? unknown

     Red tint
--------------------------------------------------------------------------------
2A4C
      bits 0:31
        osd ?? unknown
```

```
     Affects overall brightness, wrapping around to black
--------------------------------------------------------------------------------
2A50
      bits 0:31
        osd ?? unknown

     Colour shift
--------------------------------------------------------------------------------
2A54
      bits 0:31
        osd ?? unknown

     Colour shift
--------------------------------------------------------------------------------
2A58  --------    ?? unknown
  |
  V
2AFC  --------    ?? unknown
--------------------------------------------------------------------------------
2B00
      bit 0
        osd filter control
          0 = filter off
          1 = filter on

      bits 1:4
        osd ?? unknown

--------------------------------------------------------------------------------
```

v0.4 - 12 March 2007 - Ian Armstrong (ian@iarmst.demon.co.uk)