Video_gen_1

Generates continous image pattern data feed to HIBI-bus

USED IN FUNBASE FIRST APPLICATION

General Info

- 640 x 480 size, 8bit wide gray image
- 3 different test pattern images
- After reset, one image is been generated and written through HIBI-bus to DDR2 block
- After that waiting for permission to start over and producing the next image frame

Generics

- image_data_width: Bits, tells how wide is the image data.
- image_fps : Not implemented.. For the future…
  Desired frames per second…

- image_pattern_id: Which kind of image pattern is been produced;
  1 = Constant Squares, 2 = T-letter, 3 = Moving T-letter
  (Default is 1.)

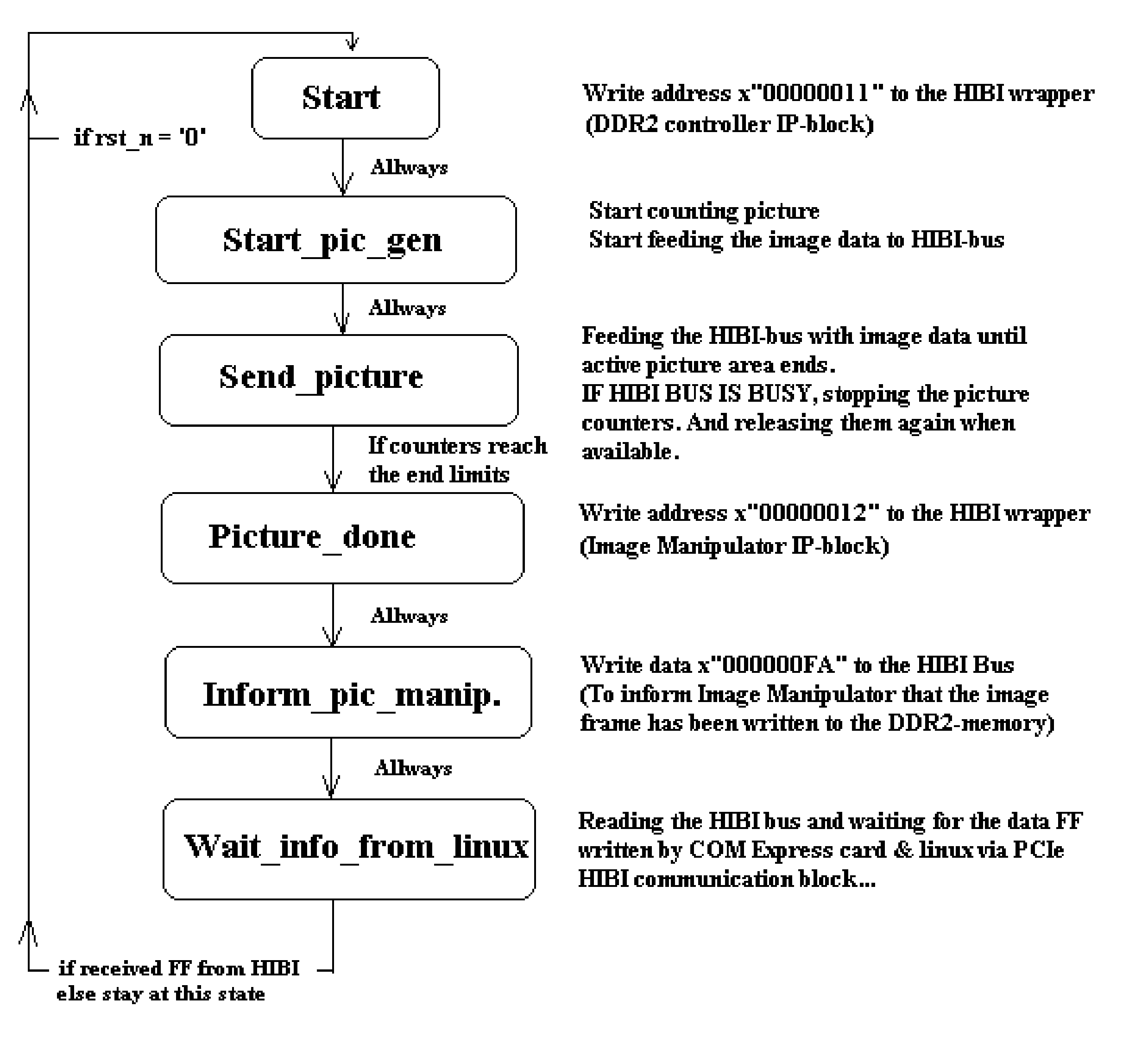
- H_pixels_across: Image size in Horizontal direction (Default 640)
- V_pixels_down: Image size in Vertical direction (Default 480)

- video_gen_1_addr_g: Own HIBI address: x"00000010"
- ddr2_controller_addr_g: DDR2 controller IP block HIBI address: x"00000011"
- Picture_manipulator_addr_g: Picture Manipulator HIBI address: x"00000012"

HIBI-BUS;

- data_width_g : integer := 32;
- comm_width_g : integer := 3;

State Diagram



Write address x"00000011" to the HIBI wrapper (DDR2 controller IP-block)

Start counting picture
Start feeding the image data to HIBI-bus

Feeding the HIBI-bus with image data until active picture area ends.
IF HIBI BUS IS BUSY, stopping the picture counters. And releasing them again when available.

Write address x"00000012" to the HIBI wrapper (Image Manipulator IP-block)

Write data x"000000FA" to the HIBI Bus (To inform Image Manipulator that the image frame has been written to the DDR2-memory)

Reading the HIBI bus and waiting for the data FF written by COM Express card & linux via PCIe HIBI communication block...

Image patterns

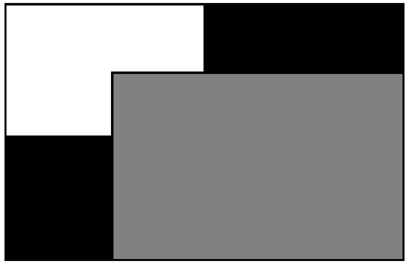


Image pattern type 1;
Constant Squares

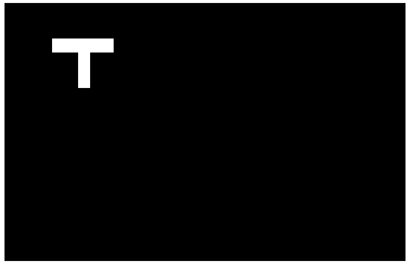


Image pattern type 2;
T-letter

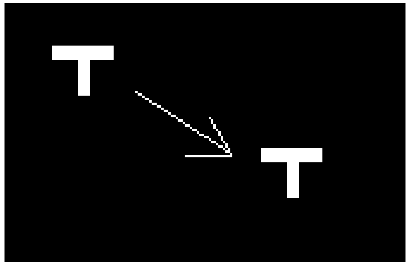


Image pattern type 3;
Moving T-letter

By changing the generic;

Image pattern type,

Several test pattern images can be produced.

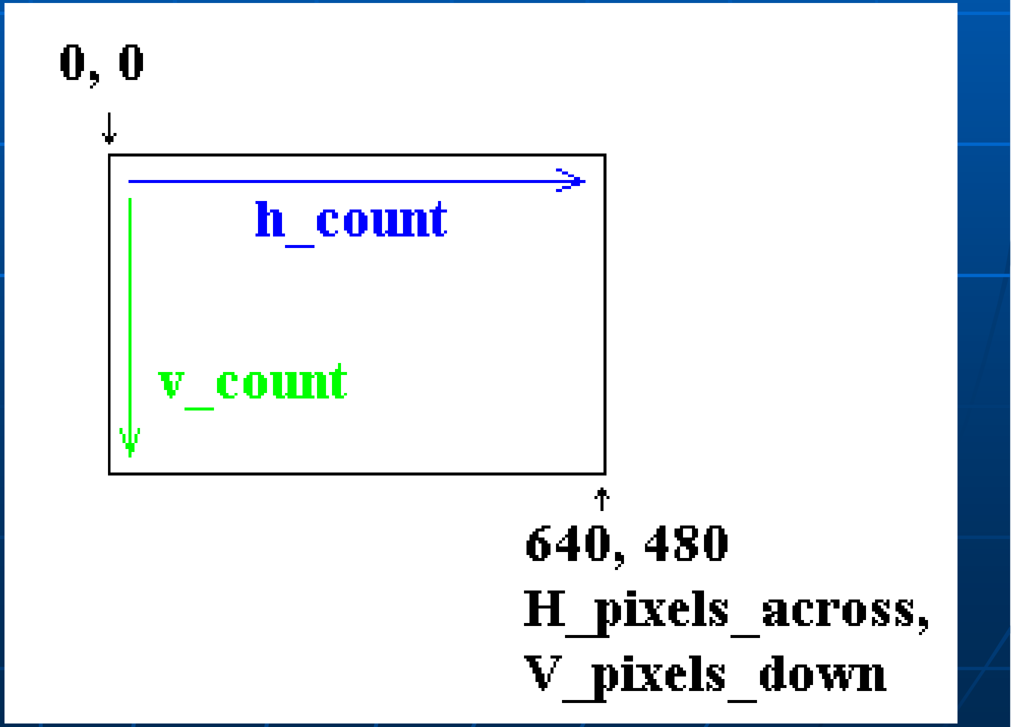
New ones can be easily made, if needed.

Process frame

```
----------------------------------------------------------------
-- PROCESS FOR CALCULATING THE PICTURE SIZE
----------------------------------------------------------------
frame : process(rst_n, pix_clk)
begin
if rst_n = '0' then
     counting_picture <= '0';
     h_count <= 0;
     v_count <= 0;
elsif (pix_clk'EVENT and pix_clk = '1')then
 if (hibi_full = '0') then
  if (start_counting_picture = '1' or counting_picture = '1') then
     if (h_count = H_pixels_across) then
                        h_count <= 0;
      else
                        h_count <= h_count + 1;
                        counting_picture <= '1';
      end if;
     if (v_count >= V_pixels_down) and (h_count >= H_pixels_across) then
                        v_count <= 0;
     elsif (h_count = H_pixels_across) then
                        v_count <= v_count + 1;
     end if;
 end if;    -- if hibi_full = '1', we will stop the counters.
end if;
end process frame;
----------------------------------------------------------------
```

If after reset or when got info to start over, then start counting the image.

If HIBI bus gets busy somehow, stopping the counters until the bus is again available.



Future versions?

- Stopping the image counters is not so good idea after all...
- What if, next time we need to plug in a camera, which clock we cant't stop?
- Need to figure out a better way ”to do the thing”, maybe use some linememory; catch up the incoming feed if HIBI is not available, or direct writing to the memory.

(Ringbuffer for the incoming video data)