Image Manipulation Manager

# Abstract

The unit controls the image manipulation process using a FSM,

**Note: X represents row indexes, Y represents column indexes.**

# Illustration

# Memory Management Generic Parameters

|  |  |  |
| --- | --- | --- |
| **Generic Parameter** | **Default Value** | **Description** |
| Reset\_polartiy\_g | '0' | Reset active in this polarity |
| img\_hor\_pixels\_g | 640 | Number of horizontal pixels(columns) |
| img\_ver\_pixels\_g | 480 | Number of horizontal pixels(rows) |
|  |  |  |
|  |  |  |

# Inputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **description** | **type** | **size** | **Recived from** |
| sys\_clk | System clock |  |  |  |
| Sys\_rst | System Reset |  |  |  |
| Req\_trig | tri |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

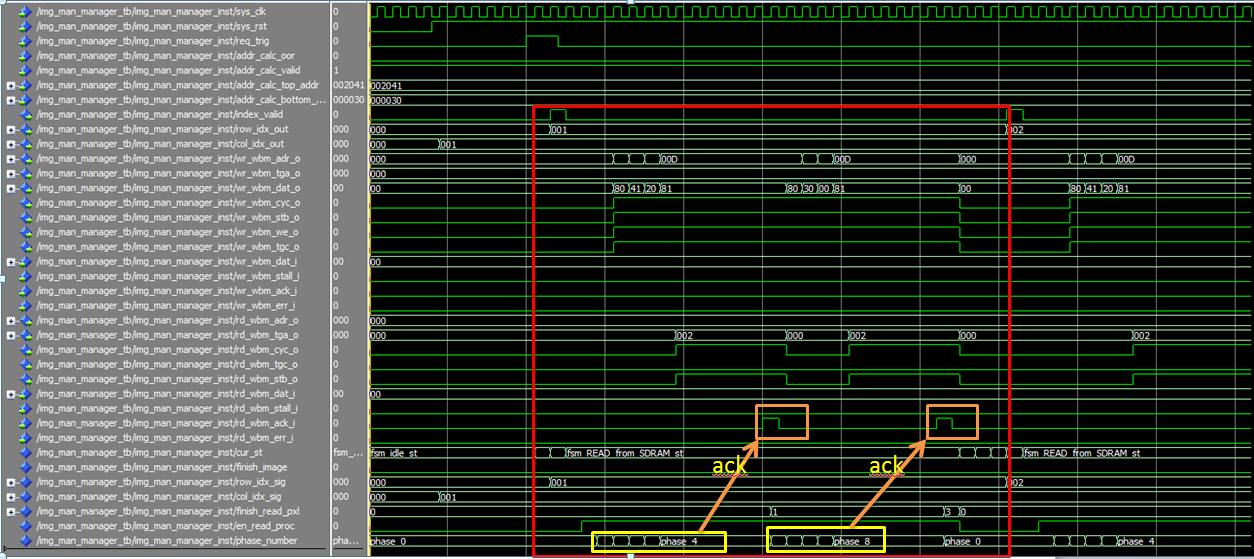
# Outputs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **description** | **type** | **size** | **Destination** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# The algorithm

The image manager consists of FSM and sub processes. A top FSM, called fsm\_proc, which controls the whole image manipulation process, and several "sub" processes (implemented with fsm or without) which execute requests from the top FSM.

* **top\_fsm**
* **read\_from\_sdram\_proc** - the process is responsible for reading 4 pixels from desired address in the image (SDRAM).  
  Read is executed by two wbm requests to the mem\_mng. Phase 1-3 works with mem\_ctrl\_wr, phase 4 is mem\_ctrl\_rd (same applies for phases 5-8).
  + phase 1 – write 0x80 to Type register
  + phase 2a - write 8 bottom bits of address to DBG\_address\_register(0x2)
  + phase 2b - write 8 top bits of address to DBG\_address\_register(0x3)
  + phase 3 - write 0x81 to Type register
  + phase 4 – read request to mem\_ctrl\_rd, and wait for rd\_wbm\_ack\_i='1'.   
    Acknowledge signal from wbs\_rd indicating data is ready.
  + phase 5 – write 0x80 to Type register
  + phase 6a - write 8 bottom bits of address to DBG\_address\_register(0x2)
  + phase 6b - write 8 top bits of address to DBG\_address\_register(0x3)
  + phase 7 - write 0x81 to Type register
  + phase 8 – read request to mem\_ctrl\_rd, and wait for rd\_wbm\_ack\_i='1'.   
    Acknowledge signal from wbs\_rd indicating data is ready.



red frame describes first pixel manipulation

* **coord\_proc** - the process is in-charge of advancing the row/col indexes until end of image.   
  When image is over a flag will rise - finish\_image.  
  Reset will set the coordinates at (0,0)

Init will set the coordinates at (1,1)

# Simulation results

# Code improvement