FRST

S0

S1

FDMA\_WR\_REQ == 1

state = S1;

vi\_fifo\_reset\_cnt = 0;

pkg\_wr\_areq =1;

IDLE

p\_fs\_f == 0

WR\_burst\_cnt = 0;

vi\_fifo\_reset\_cnt = 0;

WR\_addr = 0;

WR\_rdy\_s = 1;

state = IDLE;

p\_fs\_f == 1

state = FRST;

vi\_fifo\_reset\_cnt <= 30

WR\_rdy\_s = 0;

vi\_fifo\_reset\_cnt= vi\_fifo\_reset\_cnt+1;

vi\_fifo\_reset <=

(vi\_fifo\_reset\_cnt < 20);

state = FRST;

vi\_fifo\_reset\_cnt > 30

state = S0;

WR\_burst\_cnt==BURST\_TIMES

if(WR\_Fbuf == FBUF\_SIZE)

WR\_Fbuf = 0;

else WR\_Fbuf +1;

state = IDLE;

pkg\_wr\_last ==0

state = S1;

pkg\_wr\_areq =0;

pkg\_wr\_last ==1

state = S0;

pkg\_wr\_areq =0;

WR\_burst\_cnt= WR\_burst\_cnt + 1;

WR\_addr=WR\_addr + BURST\_SIZE;

FDMA\_WR\_REQ == 0

state = S0;

vi\_fifo\_reset\_cnt = 0;

RESET

WR\_rdy\_s = 0;

WR\_burst\_cnt = 0;

vi\_fifo\_reset\_cnt = 0;

vi\_fifo\_reset =0;

WR\_addr = 0;

WR\_rdy\_s = 1;

WR\_Fbuf =0;

pkg\_wr\_areq =0;

state = IDLE;

FDMA\_WR\_REQ<= (vi\_fifo\_rcnt>= PKG\_SIZE);

FDMA\_RD\_REQ<=(vo\_fifo\_wcnt<= PKG\_SIZE);

pkg\_wr\_addr = {W0\_Fbuf,W0\_addr}+ ADDR\_OFFSET;

pkg\_rd\_addr = {R0\_Fbuf,R0\_addr}+ ADDR\_OFFSET;

FBUF\_SIZE = BUF\_SIZE -1'b1;

BURST\_SIZE = 1024\*4; // one time 4KB

BURST\_TIMES = H\_CNT\*V\_CNT/1024;//one frame burst times

parameter PKG\_SIZE = 256; //256\*128= 4KB

pkg\_wr\_size = PKG\_SIZE;

pkg\_rd\_size = PKG\_SIZE;

FRST

S0

S1

FDMA\_RD\_REQ == 1

state = S1;

vo\_fifo\_reset\_cnt = 0;

pkg\_rd\_areq =1;

IDLE

vga\_fs\_f == 0

RD\_burst\_cnt = 0;

vo\_fifo\_reset\_cnt = 0;

RD\_addr = 0;

RD\_rdy\_s = 1;

state = IDLE;

vga\_fs\_f == 1

state = FRST;

vo\_fifo\_reset\_cnt <= 30

RD\_rdy\_s = 0;

vo\_fifo\_reset\_cnt= vo\_fifo\_reset\_cnt+1;

vo\_fifo\_reset <=

(vo\_fifo\_reset\_cnt < 20);

state =FRST;

vo\_fifo\_reset\_cnt > 30

state = S0;

RD\_burst\_cnt==BURST\_TIMES

if(WR\_Fbuf == 0 )

RD\_Fbuf = FBUF\_SIZE;

else RD\_Fbuf = WR\_Fbuf - 1;

state = IDLE;

pkg\_rd\_last ==0

state = S1;

pkg\_rd\_areq =0;

pkg\_rd\_last ==1

state = S0;

pkg\_rd\_areq =0;

RD\_burst\_cnt= RD\_burst\_cnt + 1;

RD\_addr=RD\_addr + BURST\_SIZE;

FDMA\_RD\_REQ == 0

state = S0;

vo\_fifo\_reset\_cnt = 0;

RESET

RD\_rdy\_s = 0;

RD\_burst\_cnt = 0;

vo\_fifo\_reset\_cnt = 0;

vo\_fifo\_reset =0;

RD\_addr = 0;

RD\_rdy\_s = 1;

RD\_Fbuf =0;

pkg\_rd\_areq =0;

state = IDLE;

p\_clk

p\_de

p\_data[31:0]

vi\_fifo\_reset

ui\_clk

(WR) VI\_FIFO (RD)

read latency 0

pkg\_wr\_en

**FDMA**

pkg\_wr\_data[127:0]

vi\_fifo\_wr\_cnt[10:0]

pkg\_wr\_last

FSM

BUF\_SIZE = 3;

FBUF\_SIZE= BUF\_SIZE -1;

BURST\_SIZE= 1024\*4;

BURST\_TIMES= H\_CNT\*V\_CNT/1024;

PKG\_SIZE = 256;

ui\_clk

pkg\_wr\_req

ui\_clk

ui\_rst\_n

ui\_rst\_n

pkg\_wr\_addr[31:0]

pkg\_wr\_size[31:0]

vi\_fifo\_wr\_cnt[10:0]

p\_fs\_f

vi\_fs\_cap

ui\_clk

ui\_rst\_n

p\_fs

vo\_fifo\_reset

vi\_fifo\_reset

vo\_fifo\_rd\_cnt[10:0]

pkg\_rd\_last

pkg\_rd\_req

vga\_fs\_f

vo\_fs\_cap

ui\_clk

ui\_rst\_n

vga\_fs

pkg\_rd\_addr[31:0]

pkg\_rd\_size[31:0]

vo\_fifo\_reset

vga\_clk

vga\_de

vga\_RGB[31:0]

ui\_clk

pkg\_rd\_en

(RD) VI\_FIFO (WR)

read latency 0

pkg\_rd\_data[127:0]

vo\_fifo\_rd\_cnt[10:0]

**cache\_data\_controller**

AXI4

**DDR3\_**

**Controller**

**(PL/PS)**

cahe\_data

\_controller

FDMA

(user ip)

tx\_hdmi

image\_

sensor

\_cap

PLL

(IP)

VGA\_

Timing\_gen

p\_clk

p\_fs

p\_de

p\_data[31:0]

vga\_clk

vga\_fs

vga\_de

vga\_rgb[31:0]

vga\_clk\_x5

hdmi\_clk\_p

hdmi\_clk\_n

hdmi\_d0\_p

hdmi\_d0\_n

hdmi\_d1\_p

hdmi\_d1\_n

hdmi\_d2\_p

hdmi\_d2\_n

pkg\_wr\_data[127:0]

pkg\_wr\_en

pkg\_rd\_data[127:0]

pkg\_rd\_en

pkg\_wr\_size[31:0]

pkg\_wr\_addr[31:0]

pkg\_wr\_req

pkg\_rd\_size[31:0]

pkg\_rd\_addr[31:0]

pkg\_rd\_req

pkg\_wr\_last

pkg\_rd\_last

ui\_clk

ui\_clk

rst\_n

rst\_n

locked

locked

vga\_clk

vga\_clk\_x5

locked

ui\_clk

rst\_n

DDR3

\_controller

(PS/PL)

AXI4

image\_

sensor

DDR3

**Video\_Processing\_Framework**

vga\_hs