DIGITAL DESIGN WITH FPGA CAMP

CONTEST 1 : TEST PATTERN TO HDMI WITH DDR BUFFER

ELECTRICAL ENGINEERING
SINCE 1929
CHULALONGKORN UNIVERSITY

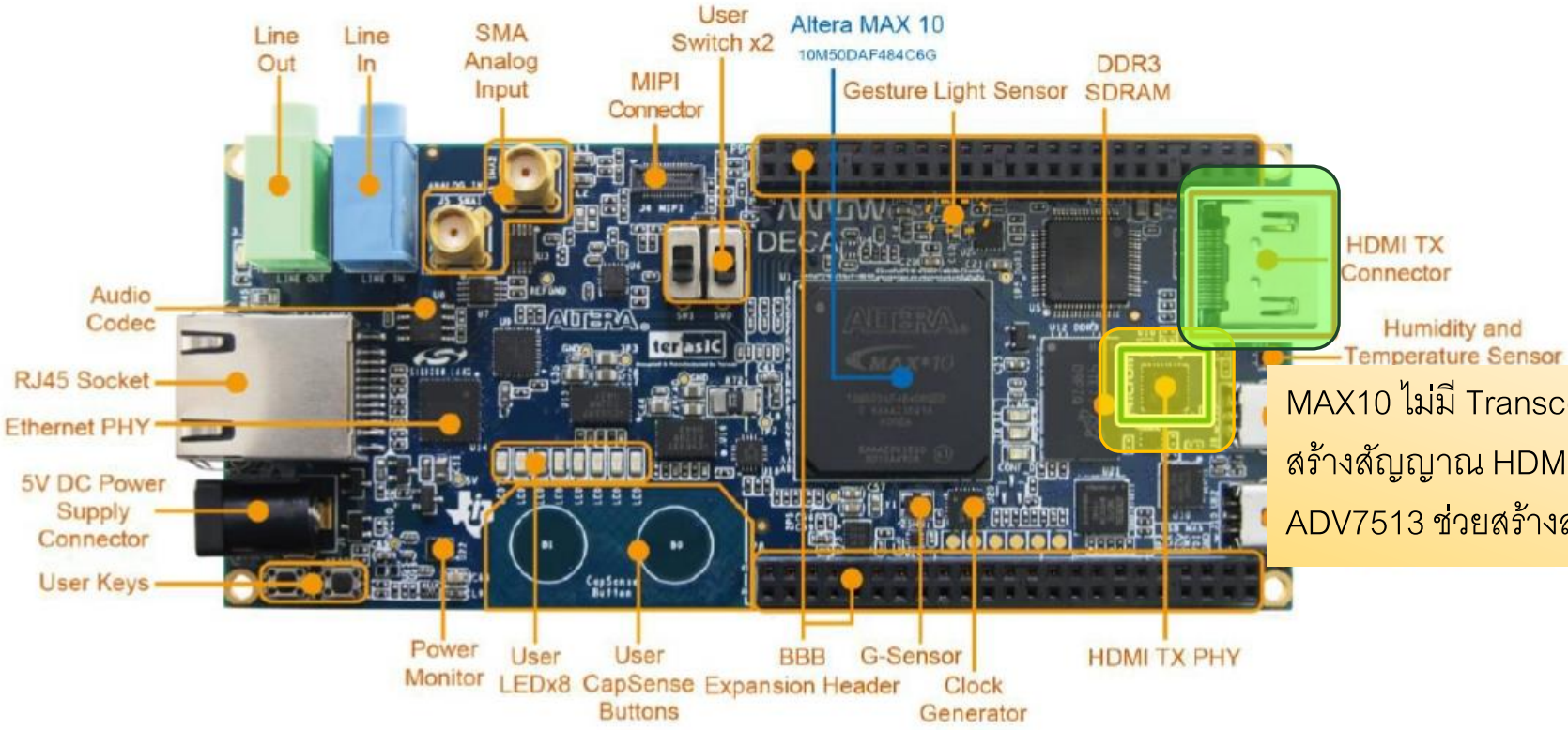
TESA

NDR Solution

DG DESIGN GATEWAY

Silicon craft

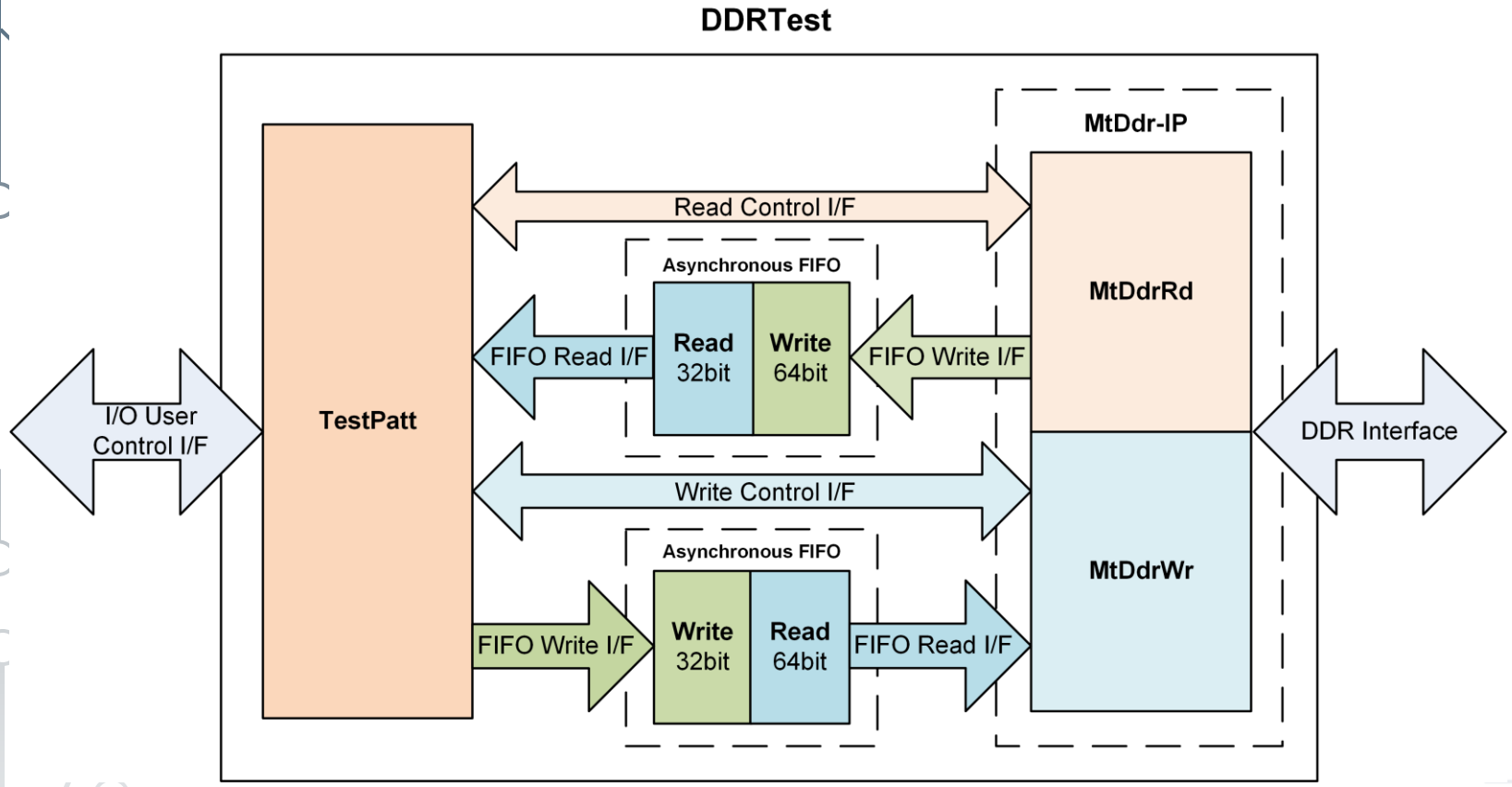
HDMI-TX ON DECA BOARD



MAX10 ไม่มี Transceiver ที่สามารถสร้างสัญญาณ HDMI TX ได้จึงต้องใช้ ADV7513 ช่วยสร้างสัญญาณ

DDR EXAMPLE

DDR EXAMPLE : BLOCK DIAGRAM

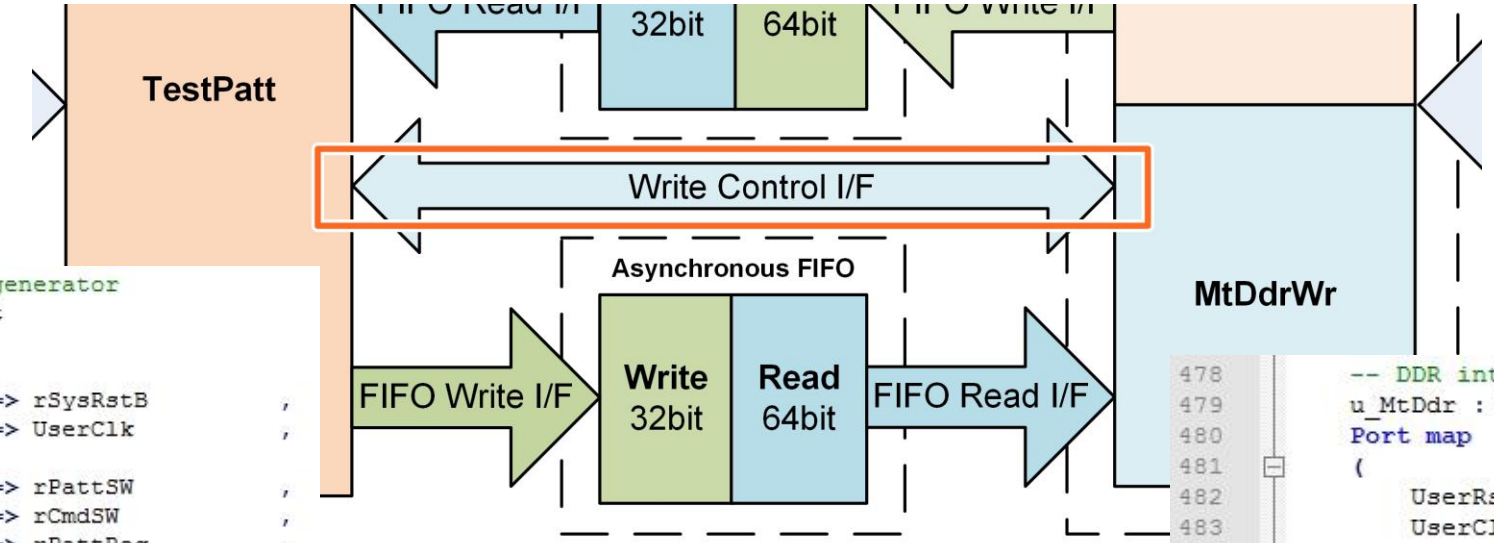


- วงจรตัวอย่างที่จะส่ง Test data เป็น increment pattern ไปเขียนลง DDR จนเต็มความจุ จากนั้นจะอ่านข้อมูลจาก DDR กลับมา เพื่อตรวจสอบความถูกต้อง

- ข้อมูลเขียนและอ่าน จะถูกส่งผ่าน FIFO ซึ่งจะมีขนาด data width ไม่เท่ากัน โดยฝั่ง TestPatt จะรับ/ส่งข้อมูลด้วยขนาด 32-bit ส่วน DDR จะรับ/ส่งข้อมูลด้วยขนาด 64-bit

WRITE OPERATION

WRITE CONTROL I/F

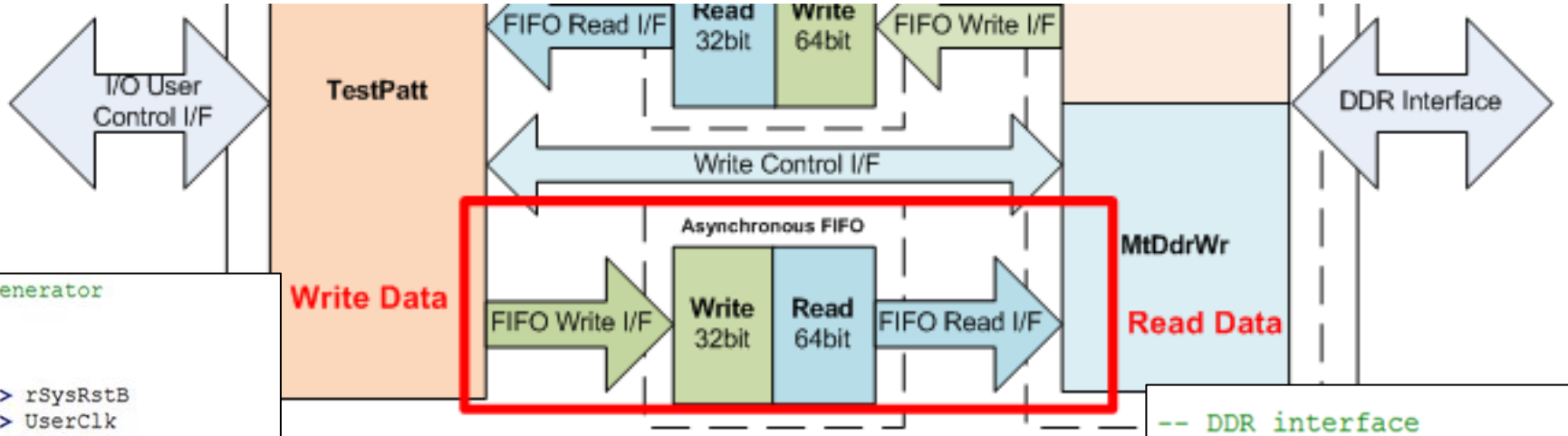



```
403  -- DDR test pattern generator
404  u_TestPatt : TestPatt
405  Port map
406  (
407      RstB            => rSysRstB         ,
408      Clk             => UserClk          ,
409
410      PattSel         => rPattSW          ,
411      PattCmd         => rCmdSW           ,
412      PattReq         => rPattReq         ,
413      PattBusy        => PattBusy         ,
414      PattFail        => PattFail         ,
415
416      MtDdrWrReq      => MtDdrWrReq       ,
417      MtDdrWrBusy     => MtDdrWrBusy      ,
418      MtDdrWrAddr     => MtDdrWrAddr      ,
419      WrFfWrCnt       => U2DdrFfWrCnt     ,
420      WrFfWrEn        => U2DdrFfWrEn      ,
421      WrFfWrData      => U2DdrFfWrData    ,
422
423      MtDdrRdReq      => MtDdrRdReq       ,
424      MtDdrRdBusy     => MtDdrRdBusy      ,
425      MtDdrRdAddr     => MtDdrRdAddr      ,
426      RdFfRdCnt       => Ddr2UFfRdCnt     ,
427      RdFfRdData      => Ddr2UFfRdData    ,
428      RdFfRdEn        => Ddr2UFfRdEn
429  );
```

```
478  -- DDR interface
479  u_MtDdr : MtDdr
480  Port map
481  (
482      UserRstB        => rSysRstB         ,
483      UserClk         => UserClk          ,
484
485      MemInitDone     => MemInitDone      ,
486
487      MtDdrWrReq      => MtDdrWrReq       ,
488      MtDdrWrBusy     => MtDdrWrBusy      ,
489      MtDdrWrAddr     => MtDdrWrAddr      ,
490      WrFfRdEn        => U2DdrFfRdEn      ,
491      WrFfRdData      => U2DdrFfRdData    ,
492      WrFfRdCnt       => U2DdrFfRdCnt     ,
493
494      MtDdrRdReq      => MtDdrRdReq       ,
495      MtDdrRdBusy     => MtDdrRdBusy      ,
496      MtDdrRdAddr     => MtDdrRdAddr      ,
497      RdFfWrEn        => Ddr2UFfWrEn      ,
498      RdFfWrData      => Ddr2UFfWrData    ,
499      RdFfWrCnt       => Ddr2UFfWrCnt     ,
```

WRITE DATA I/F

TestPatt เขียนข้อมูลไปที่ FIFO ส่วน MtDdr จะอ่านข้อมูลจาก FIFO ไปเขียนที่ DDR




```
-- DDR test pattern generator
u_TestPatt : TestPatt
Port map
(
    RstB            => rSysRstB
    Clk             => UserClk

    PattSel         => rPattSW
    PattCmd         => rCmdSW
    PattReq         => rPattReq
    PattBusy        => PattBusy
    PattFail        => PattFail

    MtDdrWrReq      => MtDdrWrReq
    MtDdrWrBusy     => MtDdrWrBusy
    MtDdrWrAddr     => MtDdrWrAddr
    WrFfWrCnt       => U2DdrFfWrCnt
    WrFfWrEn        => U2DdrFfWrEn
    WrFfWrData      => U2DdrFfWrData
```

```
-- Fifo UserData -> DDR
u_U2DdrFf : fifo256x32to64
Port map
(
    aclr            => SysRst       ,

    wrclk           => UserClk      ,
    wrreq           => U2DdrFfWrEn  ,
    data            => U2DdrFfWrData,
    wrfull          => open         ,
    wrusedw         => U2DdrFfWrCnt(7 downto 0)

    rdclk           => UserClk      ,
    rdreq           => U2DdrFfRdEn  ,
    q               => U2DdrFfRdData,
    rdempty         => open         ,
    rdusedw         => U2DdrFfRdCnt(6 downto 0)
```

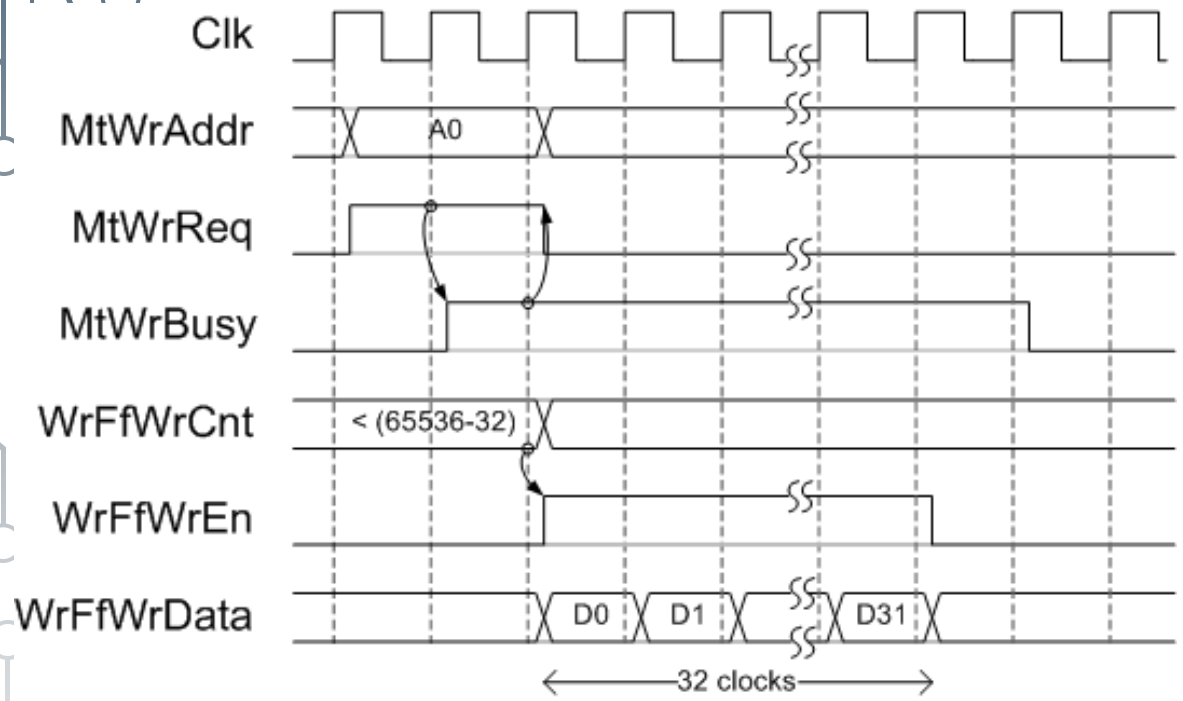
```
-- DDR interface
u_MtDdr : MtDdr
Port map
(
    UserRstB        => rSysRstB     ,
    UserClk         => UserClk      ,

    MemInitDone     => MemInitDone  ,

    MtDdrWrReq      => MtDdrWrReq   ,
    MtDdrWrBusy     => MtDdrWrBusy  ,
    MtDdrWrAddr     => MtDdrWrAddr  ,
    WrFfRdEn        => U2DdrFfRdEn  ,
    WrFfRdData      => U2DdrFfRdData,
    WrFfRdCnt       => U2DdrFfRdCnt ,
```

WRITE SIGNAL DETAILS (MTDDR-IP)

	Name	Type	Description
Control	MtDdrWrReq	Input	Request to write data to DDR (1 Req = 16x64 bit = 128 byte).
Control	MtDdrWrBusy	Output	Busy status ('0': Idle, '1': Write operating).
Control	MtDdrWrAddr[28:7]	Input	Start Address for Write Operation. (Bit[6:0] is not used because 1 request size = 128 byte).
Data	WrFfRdEn	Output	Connect to FIFO Read Enable of Write FIFO.
Data	WrFfRdData[63:0]	Input	Connect to FIFO Read Data of Write FIFO.
Data	WrFfRdCnt[15:0]	Input	Connect to FIFO Read Count of Write FIFO.

คำเตือน: ทุกสัญญาณในตารางนี้ Synchronize กับ Clock 100 MHz

TIMING DIAGRAM FOR WRITE OPERATION

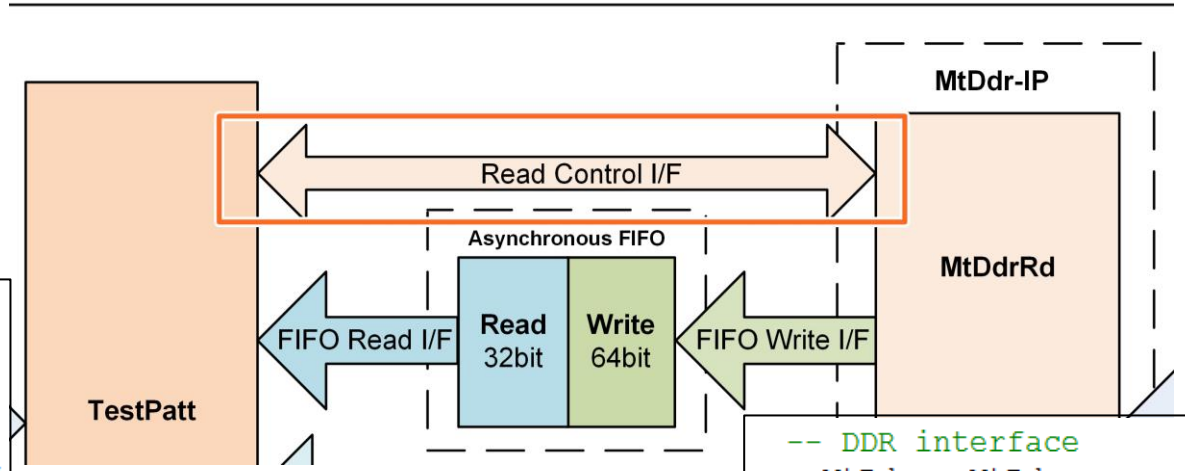


1) ส่ง MtDdrWrReq='1' พร้อมกับ MtDdrWrAddr ที่ต้องการค้างไว้
2) รอจนกระทั่ง MtDdrWrBusy='1' จึงเปลี่ยน MtDdrWrReq='0' และ MtDdrWrAddr สามารถเปลี่ยนแปลงเป็นค่าอื่นๆได้
3) เขียนข้อมูลลงใน Write FIFO ขนาด 32 bit จำนวน 32 ตัว (ตัว User logic จะ interface กับ Write FIFO ด้านเขียน (32-bit) ในขณะที่ MTDDR จะต่อกับ Write FIFO ด้านอ่าน (64-bit)
4) รอจนกระทั่ง MtDdrWrBusy='0'

การส่ง Request หนึ่งครั้งคือ การเขียนข้อมูล 32-bit จำนวน 32 ตัว เท่านั้น หากข้อมูลที่ใส่ใน FIFO มีจำนวนไม่ถึง 32 ตัว ระบบจะค้างทันที

READ OPERATION

READ CONTROL I/F

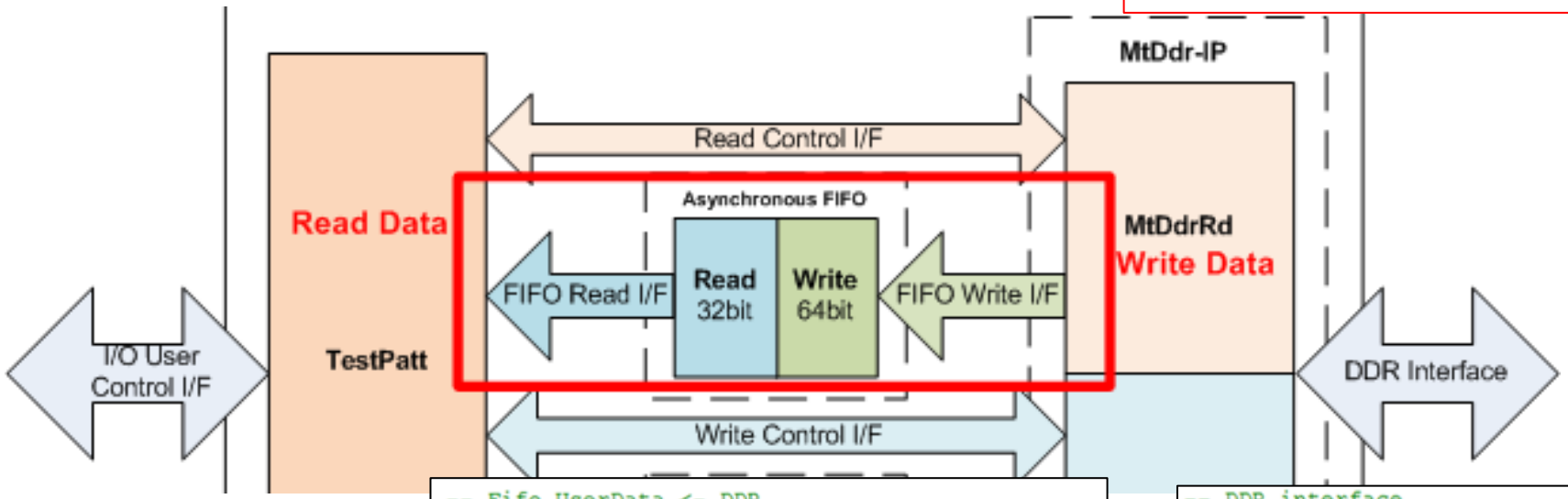



```
-- DDR test pattern generator
u_TestPatt : TestPatt
Port map
(
    RstB            => rSysRstB      ,
    Clk             => UserClk       ,

    PattSel         => rPattSW       ,
    PattCmd         => rCmdSW        ,
    PattReq         => rPattReq      ,
    PattBusy        => PattBusy      ,
    PattFail        => PattFail      ,

    MtDdrRdReq      => MtDdrRdReq    ,
    MtDdrRdBusy     => MtDdrRdBusy   ,
    MtDdrRdAddr     => MtDdrRdAddr   ,
    RdFfRdCnt       => Ddr2UFfRdCnt  ,
    RdFfRdData      => Ddr2UFfRdData ,
    RdFfRdEn        => Ddr2UFfRdEn
);
```

```
-- DDR interface
u_MtDdr : MtDdr
Port map
(
    UserRstB        => rSysRstB      ,
    UserClk         => UserClk       ,

    MemInitDone     => MemInitDone   ,

    MtDdrRdReq      => MtDdrRdReq    ,
    MtDdrRdBusy     => MtDdrRdBusy   ,
    MtDdrRdAddr     => MtDdrRdAddr   ,
    RdFfWrEn        => Ddr2UFfWrEn   ,
    RdFfWrData      => Ddr2UFfWrData ,
    RdFfWrCnt       => Ddr2UFfWrCnt  ,
```

READ DATA I/F

MtDdr เขียนข้อมูลไปที่ FIFO ส่วน TestPatt จะอ่านข้อมูลจาก FIFO ไปเขียนที่ DDR




```
-- DDR test pattern generator
u_TestPatt : TestPatt
Port map
(
    RstB            => rSysRstB         ,
    Clk             => UserClk          ,

    MtDdrRdReq      => MtDdrRdReq       ,
    MtDdrRdBusy     => MtDdrRdBusy      ,
    MtDdrRdAddr     => MtDdrRdAddr      ,
    RdFfRdCnt       => Ddr2UFfRdCnt     ,
    RdFfRdData      => Ddr2UFfRdData    ,
    RdFfRdEn        => Ddr2UFfRdEn
);
```

```
-- Fifo UserData <- DDR
u_Ddr2UFf : fifo256x64to32
Port map
(
    aclr            => SysRst           ,

    wrclk           => UserClk          ,
    wrreq           => Ddr2UFfWrEn      ,
    data            => Ddr2UFfWrData    ,
    wrfull          => open             ,
    wrusedw         => Ddr2UFfWrCnt(6 downto 0),

    rdclk           => UserClk          ,
    rdreq           => Ddr2UFfRdEn      ,
    q               => Ddr2UFfRdData    ,
    rdempty         => open             ,
    rdusedw         => Ddr2UFfRdCnt(7 downto 0)
);
```

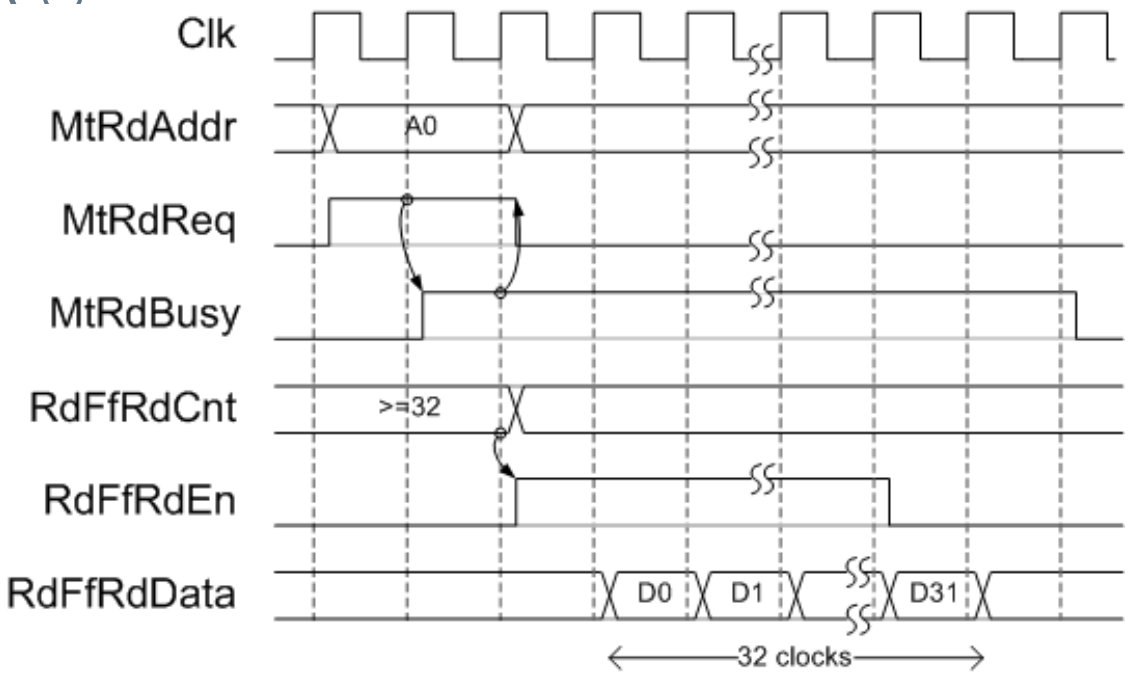
```
-- DDR interface
u_MtDdr : MtDdr
Port map
(
    UserRstB        => rSysRstB         ,
    UserClk         => UserClk          ,

    MemInitDone     => MemInitDone      ,

    MtDdrRdReq      => MtDdrRdReq       ,
    MtDdrRdBusy     => MtDdrRdBusy      ,
    MtDdrRdAddr     => MtDdrRdAddr      ,
    RdFfWrEn        => Ddr2UFfWrEn      ,
    RdFfWrData      => Ddr2UFfWrData    ,
    RdFfWrCnt       => Ddr2UFfWrCnt     ,
```

READ SIGNAL DETAILS (MTDDR-IP)

	Name	Type	Description
Control	MtDdrRdReq	Input	Request to read data from DDR (1 Req = 16x64 bit = 128 byte).
Control	MtDdrRdBusy	Output	Busy status (‘0’: Idle, ‘1’: Read operating).
Control	MtDdrRdAddr[28:7]	Input	Start Address for Read Operation in Byte unit. (Bit[6:0] is not used because 1 request size = 128 byte).
Data	RdFfWrEn	Output	Connect to FIFO Write Enable of Read FIFO.
Data	RdFfWrData[63:0]	Output	Connect to FIFO Write Data of Read FIFO.
Data	RdFfWrCnt[15:0]	Input	Connect to FIFO Write Count of Read FIFO.

คำเตือน: ทุกสัญญาณในตารางนี้ Synchronize กับ Clock 100 MHz

TIMING DIAGRAM FOR READ OPERATION



1) ส่ง MtDdrRdReq='1' พร้อมกับ MtDdrRdAddr ที่ต้องการค้างไว้
2) รอจนกระทั่ง MtDdrRdBusy='1' จึงเปลี่ยน MtDdrRdReq='0' และ MtDdrRdAddr สามารถเปลี่ยนแปลงเป็นค่าอื่น ๆ ได้
3) รอจนกระทั่ง RdFfRdCnt มีค่ามากกว่า 32
4) อ่านข้อมูลจาก RdFIFO ทั้งหมด 32 ตัว

การส่ง Request หนึ่งครั้งคือ การอ่านข้อมูล 32-bit จำนวน 32 ตัว เท่านั้น

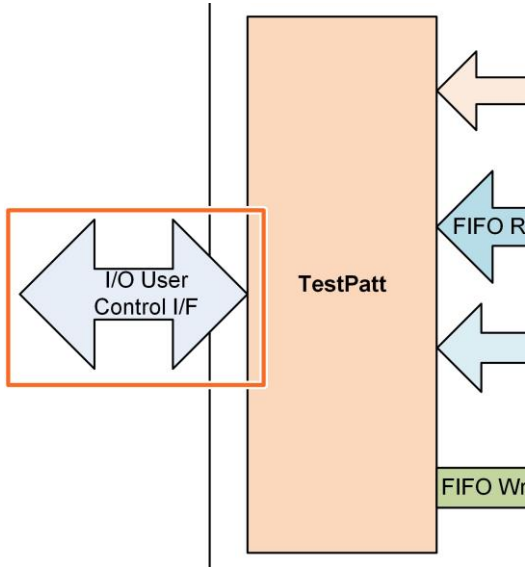
TESTPATT TO TEST DDR

TESTPATT MODULE

ทำหน้าที่เขียนข้อมูลลง DDR และอ่านกลับมาเพื่อตรวจสอบข้อมูลว่าถูกต้องหรือไม่ แบ่งสัญญาณออกเป็นกลุ่ม ๆ ดังนี้

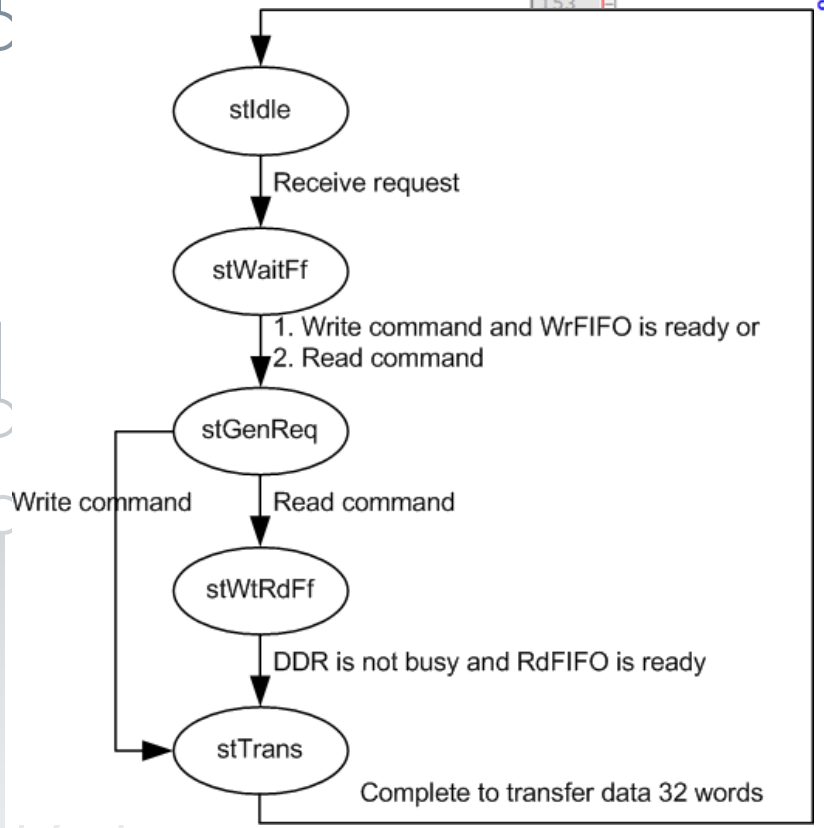
1. External User control เพื่อให้จังหวะการเริ่มทำงาน
2. MtDdrWr เป็นสัญญาณควบคุม เพื่อส่ง request ไปบอก MTDDR ว่าจะขอเขียนข้อมูล 128 byte
3. WrFf เป็นกลุ่มสัญญาณเพื่อส่งข้อมูลขนาด 32-bit ไปเขียนลง DDR ผ่าน FIFO
4. MtDdrRd เป็นสัญญาณควบคุม เพื่อส่ง request ไปบอก MTDDR ว่าจะขออ่านข้อมูล 128 byte
5. RdFf เป็นกลุ่มสัญญาณเพื่ออ่านข้อมูลขนาด 32-bit จาก DDR ผ่าน FIFO

DDRTEST : USER CONTROL I/F




```
403     -- DDR test pattern generator
404     u_TestPatt : TestPatt
405     Port map
406     (
407         RstB            => rSysRstB         ,
408         Clk             => UserClk          ,
409
410         PattSel         => rPattSW          ,
411         PattCmd         => rCmdSW           ,
412         PattReq         => rPattReq         ,
413         PattBusy        => PattBusy         ,
414         PattFail        => PattFail         ,
415
416         MtDdrWrReq      => MtDdrWrReq       ,
417         MtDdrWrBusy     => MtDdrWrBusy      ,
418         MtDdrWrAddr     => MtDdrWrAddr      ,
419         WrFfWrCnt       => U2DdrFfWrCnt     ,
420         WrFfWrEn        => U2DdrFfWrEn      ,
421         WrFfWrData      => U2DdrFfWrData    ,
422
423         MtDdrRdReq      => MtDdrRdReq       ,
424         MtDdrRdBusy     => MtDdrRdBusy      ,
425         MtDdrRdAddr     => MtDdrRdAddr      ,
426         RdFfRdCnt       => Ddr2UFfRdCnt     ,
427         RdFfRdData      => Ddr2UFfRdData    ,
428         RdFfRdEn        => Ddr2UFfRdEn
429     );
```

TESTPATT : STATE MACHINE




```
145   ------------------------------------------------------------------
146   -- State Machine
147   u_rState : Process (Clk) Is
148   Begin
149       if ( rising_edge(Clk) ) then
150           if ( RstB='0' ) then
151               rState      <= stIdle;
152           else
153               case ( rState ) is
154                   -- Wait start pulse
155                   when stIdle     =>
156                       -- Receive request from user
157                       if ( PattReq='1' ) then
158                           rState  <= stWaitFf;
159                       else
160                           rState  <= stIdle;
161                       end if;
162
163                   -- Wait until FIFO is ready
164                   when stWaitFf   =>
165                       -- Ddr is not busy
166                       -- Free space is more than 32 words
167                       if ( (rCmd='0' and MtDdrWrBusy='0' and
168                            WrFfWrCnt(15 downto 6)/=("11"&x"FF"))
169                       -- Ddr is not busy
170                         or (rCmd='1' and MtDdrRdBusy='0') ) then
171                           rState  <= stGenReq;
172                       else
173                           rState  <= stWaitFf;
174                       end if;
175
176                   -- Generate request to DDR
177                   when stGenReq   =>
178                       -- Write Command is received
179                       if ( rCmd='0' and MtDdrWrBusy='1' ) then
180                           rState  <= stTrans;
181                       -- Read Command is received
182                       elsif ( rCmd='1' and MtDdrRdBusy='1' ) then
183                           rState  <= stWtRdFf;
184                       else
185                           rState  <= stGenReq;
186                       end if;
187
188                   -- Wait 32 words available
189                   when stWtRdFf   =>
190                       if ( RdFfRdCnt(15 downto 5)/=0 ) then
191                           rState  <= stTrans;
192                       else
193                           rState  <= stWtRdFf;
194                       end if;
195
196                   -- Transfer data to/from DDR
197                   when stTrans    =>
198                       -- End of burst transfer
199                       if ( rBurstCnt(4 downto 0)=31 ) then
200                           -- End transfer when next address is end address
201                           if ( rMtDdrAddr(28 downto 7)=cEndPattReq(21 downto 0) ) then
202                               rState  <= stIdle;
203                           -- Continue for next burst transfer
204                           else
205                               rState  <= stWaitFf;
206                           end if;
207                       -- Still not complete current burst transfer
208                       else
209                           rState  <= stTrans;
210                       end if;
211               end case;
212           end if;
213       end if;
214   End Process u_rState;
```

MORE DETAILS FOR TESTPATT

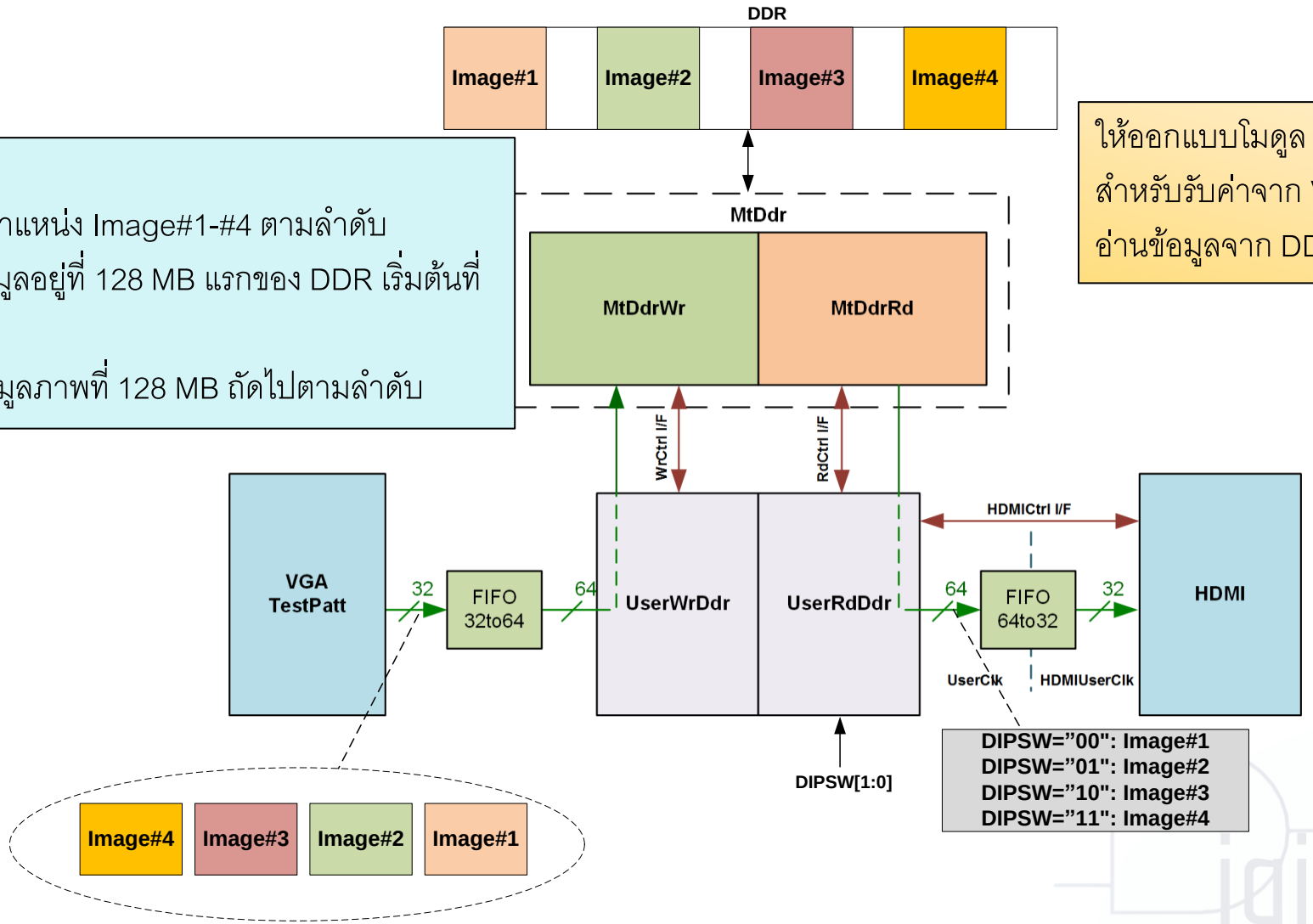
https://forfpgadesign.wordpress.com/2017/12/11/10-2-example-design2/

CONTEST1

CONTEST 1 TEST PATTERN TO HDMI WITH DDR BUFFER

UserWrDdr:
ส่งภาพ 1-4 ขึ้น DDR ที่ตำแหน่ง Image#1-#4 ตามลำดับ
โดย Image#1 จะเก็บข้อมูลอยู่ที่ 128 MB แรกของ DDR เริ่มต้นที่ตำแหน่ง 0 จนจบภาพ
Image#2 - #4 จะเก็บข้อมูลภาพที่ 128 MB ถัดไปตามลำดับ

ให้ออกแบบโมดูล UserWrDdr และ UserRdDdr สำหรับรับค่าจาก VGA TestPatt มาเขียนลง DDR และอ่านข้อมูลจาก DDR มาแสดงผลที่ HDMI



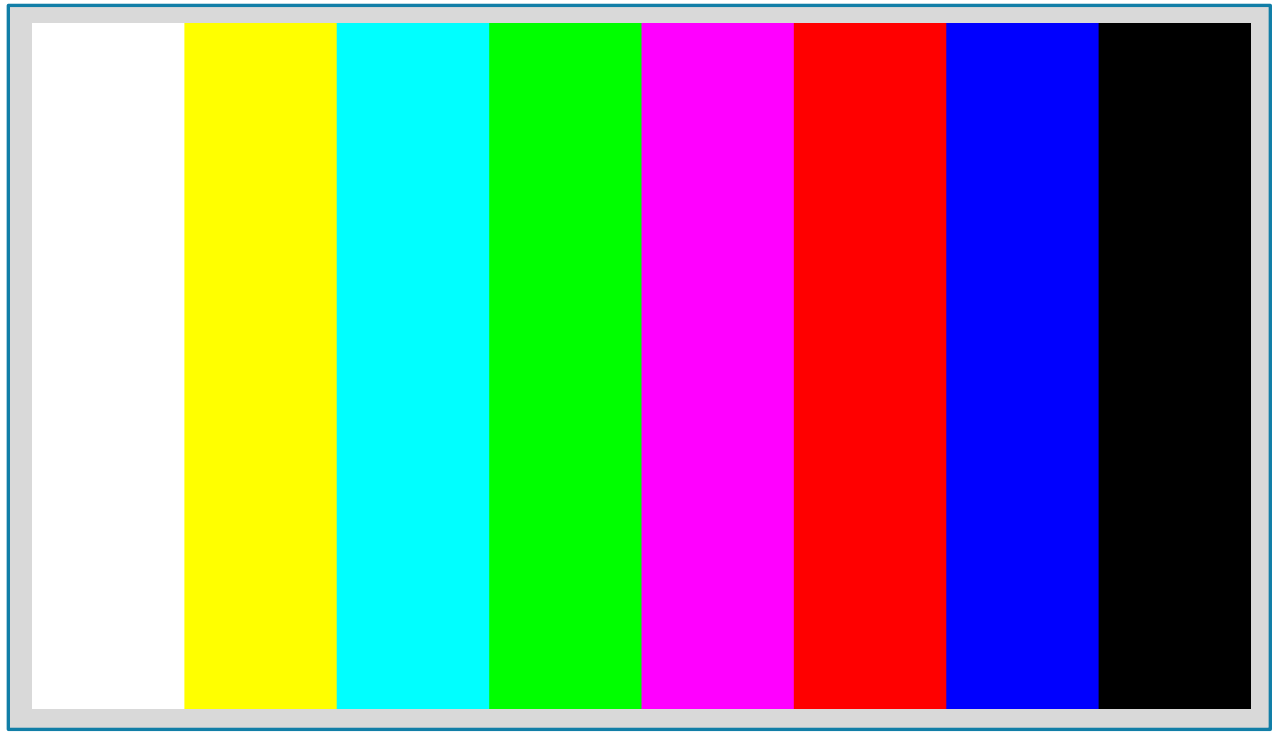
UserRdDdr:
เมื่อ DIPSW="00" จะแสดงภาพ1
เมื่อ DIPSW="01" จะแสดงภาพ2
เมื่อ DIPSW="10" จะแสดงภาพ3
เมื่อ DIPSW="11" จะแสดงภาพ4

VGA TESTPATT

จะต่างจาก TESTPATT ใน HDMI Example คือ จะเป็นวงจรสร้างข้อมูลภาพขนาด 1024 x 768 pixel ทั้งหมด 4 ภาพ ซึ่งจะขึ้นอยู่กับค่า DIPSW ว่าจะเป็นภาพไหน วนไปไม่รู้จบ
โดยมีภาพดังนี้คือ

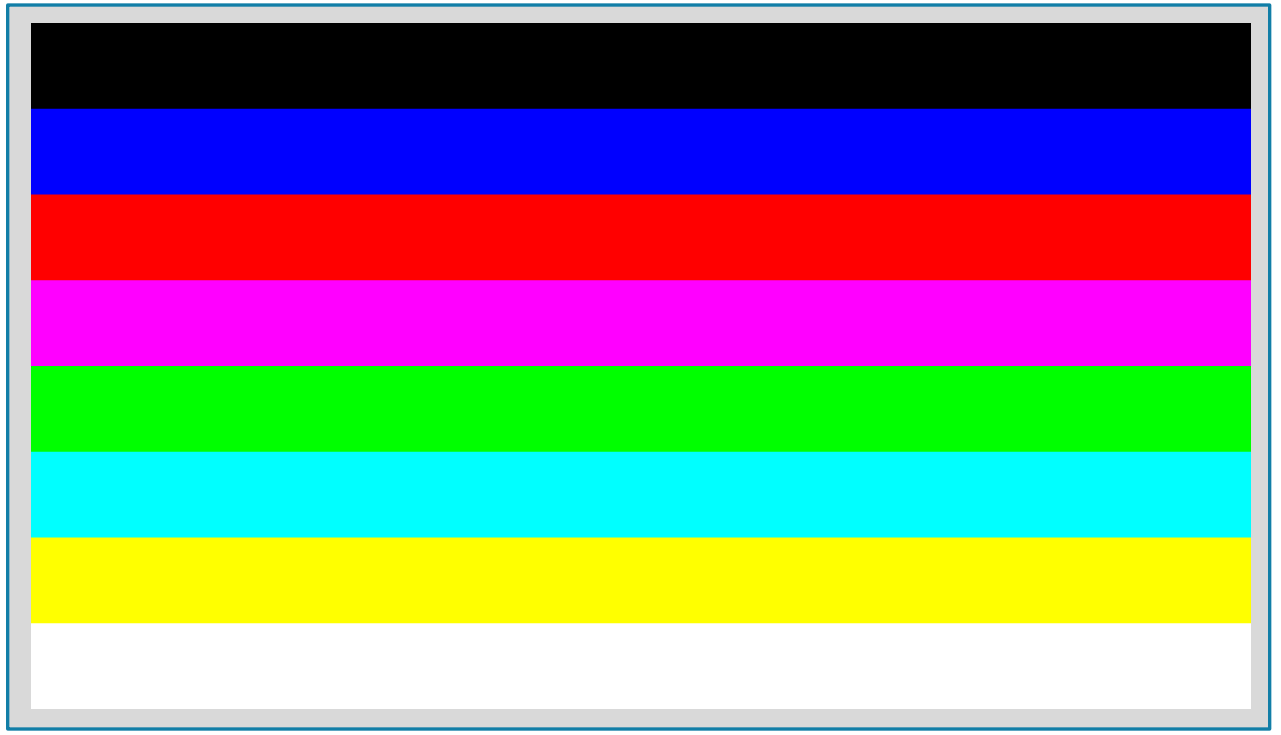
1. Vertical color bar
2. Horizontal color bar
3. Red screen
4. Blue screen

VERTICAL COLOR BAR (SW="00")



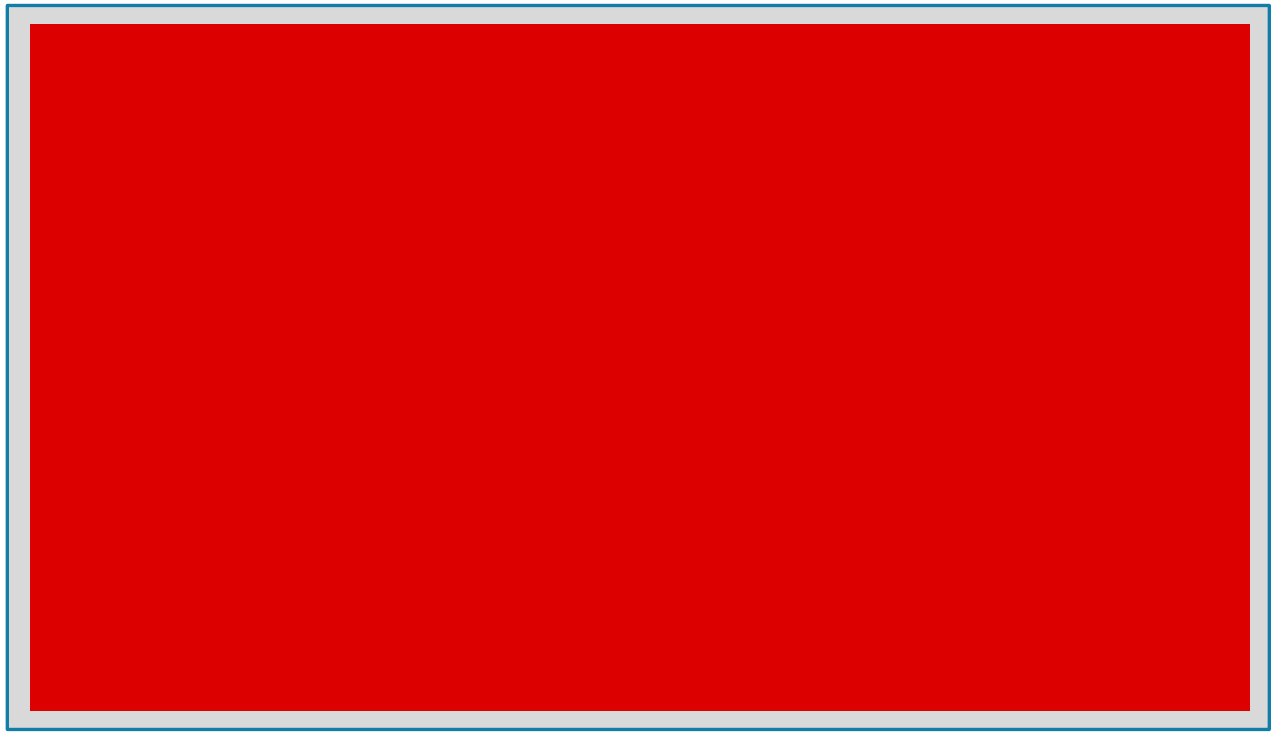
https://commons.wikimedia.org/wiki/File:EBU_Colorbars_HD.svg

HORIZONTAL COLOR BAR (SW="01")



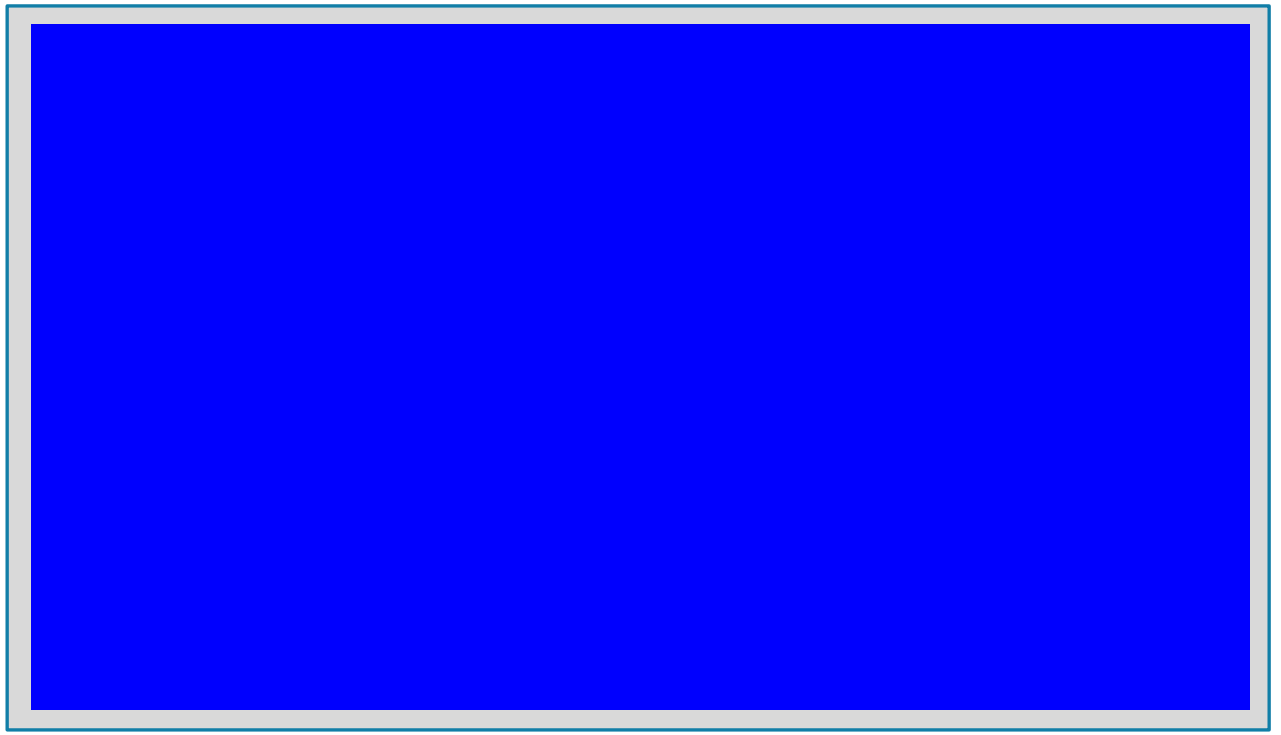
https://commons.wikimedia.org/wiki/File:EBU_Colorbars_HD.svg

RED SCREEN (SW="10")



https://commons.wikimedia.org/wiki/File:EBU_Colorbars_HD.svg

BLUE SCREEN (SW="11")



https://commons.wikimedia.org/wiki/File:EBU_Colorbars_HD.svg

TESTPATT : SELECT PATTERN

```
u_HDMIFfWrData : Process (Clk) Is
Begin
    if ( rising_edge(Clk) ) then
        if ( RstB='0' ) then
            rHDMIRed    <= x"FF";   -- 255
            rHDMIGreen  <= x"FF";   -- 255
            rHDMIBlue   <= x"FF";   -- 255
        else
            case ( DipSwitch(1 downto 0) ) is
                -- Vertical Color Bar
                when "00"   =>
                    rHDMIRed    <= rVCBRed;
                    rHDMIGreen  <= rVCBGreen;
                    rHDMIBlue   <= rVCBBlue;

                -- Horizontal Color Bar
                when "01"   =>
                    rHDMIRed    <= rHCBRed;
                    rHDMIGreen  <= rHCBGreen;
                    rHDMIBlue   <= rHCBBlue;

                -- Red Screen
                when "10"   =>
                    rHDMIRed    <= x"FF";           -- 255
                    rHDMIGreen  <= (others=>'0');   -- 0
                    rHDMIBlue   <= (others=>'0');   -- 0

                -- Blue Screen
                -- when "11"
                when others =>
                    rHDMIRed    <= (others=>'0');   -- 0
                    rHDMIGreen  <= (others=>'0');   -- 0
                    rHDMIBlue   <= x"FF";           -- 255
            end case;
        end if;
    end if;
End Process u_HDMIFfWrData;
```

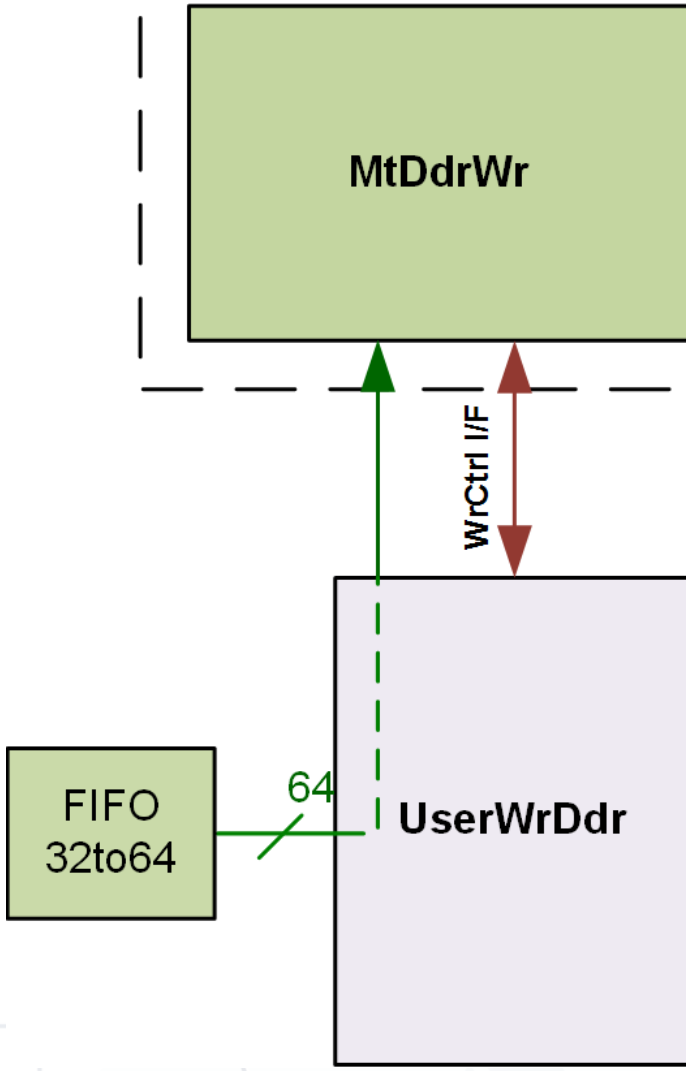
DIPSWITCH จะนำไปใช้ในการเลือกภาพที่จะส่งออกว่าเป็นภาพใด

(“00”: Vertical color bar, “01”: Horizontal color bar,

“10”: Red, “11”: Blue)

USERWRDDR

- ส่งคำสั่ง WrReq ไปที่ MtDdrWr เพื่อเขียนข้อมูล 128-byte ต่อ 1 คำสั่ง
โดย WrReq=‘1’ พร้อมระบุ address ที่ต้องการเขียน
Note: สัญญาณ Address จะมีขนาด [28:7] เพราะต้องการบอกหน่วยเป็นระดับ 128-byte
เช่น Address[28:7]=0 สำหรับคำสั่งเขียนข้อมูล byte ที่ 0 – 127
Address[28:7]=1 สำหรับคำสั่งเขียนข้อมูล byte ที่ 127- 255 เป็นต้น
- แบ่งหน่วยความจำออกเป็น 4 ส่วน เพื่อเก็บภาพตัวอย่าง 4 ภาพ โดยเลือกจาก Address bit ที่ [28:27] ว่าถ้ามีค่าเป็น “00” จะใช้เก็บข้อมูลภาพแรก ”01” ภาพสอง “10” ภาพสาม และ “11” ภาพที่สี่ (แต่ละส่วนมีขนาด 128 Mbyte ซึ่งใหญ่กว่าขนาดภาพที่ใช้จริง คือ 1024 x 768 x4 byte มาก ทั้งนี้เพื่อให้สามารถออกแบบวงจรควบคุมได้ง่าย)
- เมื่อส่งข้อมูลไปที่ DDR ครบ 1 ภาพ (ขนาด 1024 x 768 x 4 byte) ก็จะปรับค่า Address ไปเริ่มต้นชี้ที่ตำแหน่งแรกของภาพใหม่ (ขยับ bit[28:27] ไปที่ 128M ถัดไป และเคลียร์ค่า[26:0]=0)
- ส่วนที่เป็นสัญญาณ data และ valid ของ data ทั้งหมด จะใช้การเชื่อมโยงสัญญาณโดยตรง (bypass) ระหว่าง FIFO กับ MtDdrWr ไม่มีการสร้าง logic เพิ่มเติม



USERRDDDR

- ส่งคำสั่ง RdReq ไปที่ MtDdrRd เพื่ออ่านข้อมูล 128-byte ต่อ 1 คำสั่ง พร้อมระบุ address ที่ต้องการอ่าน

Note: สัญญาณ Address จะมีขนาด [28:7] เพราะต้องการบอกหน่วยเป็นระดับ 128-byte เช่น Address[28:7]=0 สำหรับคำสั่งเขียนข้อมูล byte ที่ 0 – 127

Address[28:7]=1 สำหรับคำสั่งเขียนข้อมูล byte ที่ 127- 255 เป็นต้น

- ตำแหน่ง Address ที่จะอ่าน ขึ้นกับค่า DIPSW ที่ได้รับ กล่าวคือ

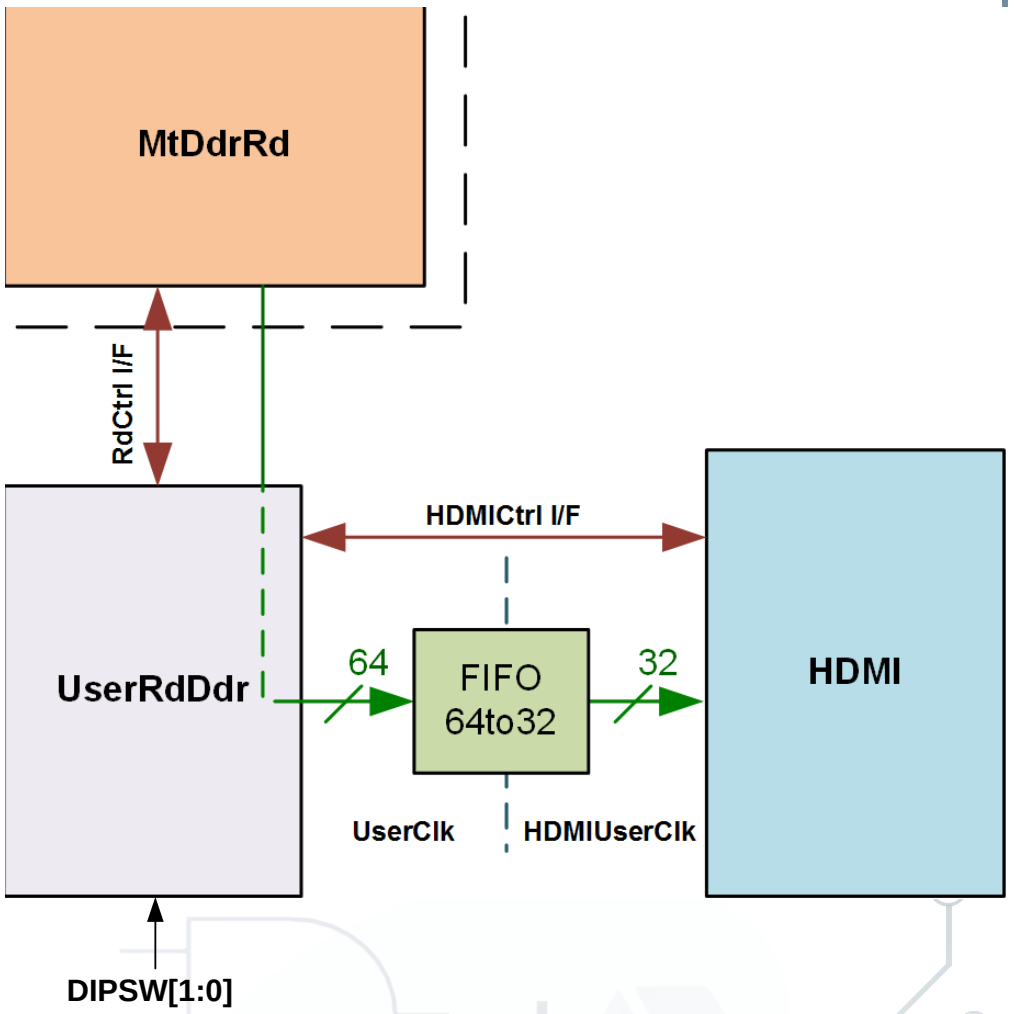
DIPSW="00" ให้อ่านข้อมูลภาพเริ่มต้นจากตำแหน่ง Address[28:27]="00", [26:0]=0

DIPSW="01" ให้อ่านข้อมูลภาพเริ่มต้นจากตำแหน่ง Address[28:27]="01", [26:0]=0

DIPSW="10" ให้อ่านข้อมูลภาพเริ่มต้นจากตำแหน่ง Address[28:27]="10", [26:0]=0

DIPSW="11" ให้อ่านข้อมูลภาพเริ่มต้นจากตำแหน่ง Address[28:27]="11", [26:0]=0

- ตำแหน่ง Address จะเปลี่ยนค่าเริ่มต้นใหม่ (อ่านค่าจาก DIPSW ใหม่เมื่อจบการส่งภาพแต่ละภาพแล้วเท่านั้น ไม่เปลี่ยนค่าระหว่างที่ยังส่งข้อมูลภาพเดิมไม่เสร็จ)
- ส่วนที่เป็นสัญญาณ data และ valid ของ data ทั้งหมด จะใช้การเชื่อมโยงสัญญาณโดยตรง(bypass) ระหว่าง MtDdrRd กับ FIFO ไม่มีการสร้าง logic เพิ่มเติม



TEST ON REAL BOARD

- TestPatt จะเริ่มส่งข้อมูลภาพ 1 ภาพเมื่อมีการกดปุ่ม KEY1 1 ครั้ง โดยภาพที่ส่งจะขึ้นกับ DIPSW
- ถ้าจะส่งข้อมูล 4 ภาพไปเก็บไว้ที่ DDR ต้องทำตามขั้นตอนดังนี้

(1) เลือก DIPSW=“00” แล้วกดปุ่ม KEY1 1 ครั้งเพื่อเก็บภาพที่ 1 ไปที่ตำแหน่ง 128 MB แรกของ DDR

(2) เลือก DIPSW=“01” แล้วกดปุ่ม KEY1 1 ครั้งเพื่อเก็บภาพที่ 2 ไปที่ตำแหน่ง 128 MB ถัดไปของ DDR

(3) เลือก DIPSW=“10” แล้วกดปุ่ม KEY1 1 ครั้งเพื่อเก็บภาพที่ 3 ไปที่ตำแหน่ง 128 MB ถัดไปของ DDR

(4) เลือก DIPSW=“11” แล้วกดปุ่ม KEY1 1 ครั้งเพื่อเก็บภาพที่ 4 ไปที่ตำแหน่ง 128 MB สุดท้ายของ DDR

- หลังจากนั้น เราสามารถเปลี่ยนภาพที่แสดงบนจอไปมาได้เลย ผ่าน DIPSW เพื่อให้ฝั่งอ่าน DDR เลือกอ่านภาพที่ 1-4 ตามที่ต้องการแสดง โดยที่วงจรฝั่งเขียนภาพจะไม่ได้ทำงานเลย
- หากเลือก DIPSW แล้วกด KEY1 ไม่เรียงตามลำดับ “00” “01” “10” “11” ภาพที่จะเขียนขึ้น DDR จะสลับกัน เพราะข้อมูลภาพที่จะเขียน DDR จะถูกเลือกจาก DIPSW ใน TestPatt แต่ตำแหน่ง Address ที่จะเขียนภาพถูกกำหนดที่ UserWrDdr ซึ่งจะกำหนดตำแหน่งตั้งแต่ 128 MB แรกจน 128 MB สุดท้าย แล้ววนกลับไปตำแหน่งแรกใหม่