Atmark Techno

XPS SIV00 (v1.00c)

Introduction

XPS-SIV00 は SUZAKU-V(SZ410-U00)と SUZAKU I/O シリーズ AV ボード(SIV00-U00)[1]を接続した状態で使用できる Video 用 IP コアのサンプルです。

AV ボードは映像と音データの取得・出力が可能なボードです。XPS-SIV00 は取得した ITU656 フォーマットのデータをデジタルデータに変換して SDRAM に展開すること、SDRAM のデジタルデータを ITU656 フォーマットのデータに変換して出力することができます。

Features

XPS-SIV00 は以下のような特徴を持っています。

- SUZAKU-I/O シリーズ AV ボード専用
- SZ410-U00 のみ対応
- NTSC 入出力
- エフェクト機能(2 値化、グレースケール、エッジ検出)

Caution

本 IP コア(付属のドキュメント等も含みます)は、現状のまま(AS IS)提供されるものであり、特定の目的に適合することや、その信頼性、正確性を保証するものではありません。

また、本製品の使用による結果についてもなんら保証するものではありません。

Block Diagram

XPS-SIV00 のブロック図を示します。

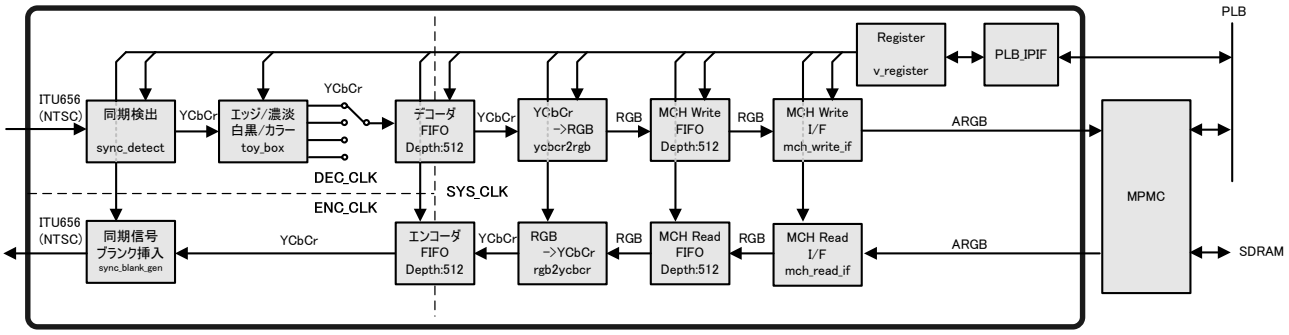

Figure 1 : XPS-SIV00 Block Diagram

[1] AV ボード(SIV00-U00)の詳細につきましては AV ボードハードウェアマニュアルをご参照ください。

Parameters

XPS-SIV00 のパラメータを示します。

Table 1 : Design Parameters

Feature/Description	Parameter Name	Allowable Values	Default Values	VHDL Type
XPS SIV00 Base Address	C_MEM0_BASEADDR	Valid Address	0xF0FF 4200	std_logic_vector
XPS SIV00 High Address	C_MEM0_HIGHADDR	Valid Address	0xF0FF 43FF	std_logic_vector
PLB Address Width	C_PLB_AWIDTH	32	32	integer
PLB Data Width	C_PLB_DWIDTH	32, 64, 128	128	integer
Number of PLB Masters	C_SPLB_NUM_MASTERS	1:16	8	integer
PLB Master ID Bus Width	C_SPLB_MID_WIDTH	1:4	3	integer
Width of the Slave Data Bus	C_SPLB_NATIVE_DWIDTH	32, 64, 128	32	integer
Selects point-to-point or shared PLB topology	C_SPLB_P2P	0,1	0	integer
Selects the transactions as being single beat or burst	C_SPLB_SUPPORT_BURSTS	0,1	0	integer
Data Width of Smallest Master	C_SPLB_SMALLEST_MASTER	32, 64, 128	32	integer
Period of the PLB Clock	C_SPLB_CLK_PERIOD_PS	Valid Period	11428	integer
Target FPGA Family	C_FAMILY	virtex4	virtex4	string
Capture Address	C_CAP_ADDR	Valid Address	0x0000 0000	std_logic_vector
Display Address	C_DISP_ADDR	Valid Address	0x0000 0000	std_logic_vector

I/O Signals

XPS-SIV00 の I/O Signals を示します。

Table 2 : I/O Signals

Signal Name	I/O	Initial State	Description
SPLB_Clk	I		PLB Clock
SPLB_Rst	I		PLB RESET
PLB_ABus[0:31]	I		PLB Address Bus
PLB_UABus[0:31]	I		PLB Upper Address Bits
PLB_PAValid	I		PLB Primary Address Valid
PLB_SAValid	I		PLB Secondary Address Valid
PLB_rdPrim	I		PLB Read Primary
PLB_wrPrim	I		PLB Write Primary
PLB_masterID[0:C_SPLB_MID_WIDTH-1]	I		PLB Master Identification
PLB_abort	I		PLB Transaction Abort
PLB_busLock	I		PLB Bus Lock
PLB_RNW	I		PLB Read Not Write
PLB_BE[0:C_SPLB_DWIDTH/8-1]	I		PLB Byte Enable
PLB_MSize[0:1]	I		PLB Master Size
PLB_size[0:3]	I		PLB Size
PLB_type[0:2]	I		PLB Transaction Type
PLB_lockErr	I		PLB Bus Lock Error
PLB_wrDBus[0:C_SPLB_DWIDTH-1]	I		PLB Write Data Bus
PLB_wrBurst	I		PLB Write Burst
PLB_rdBurst	I		PLB Read Burst
PLB_wrPendReq	I		PLB Write Pending Request
PLB_rdPendReq	I		PLB Read Pending Request

PLB_wrPendPri[0:1]	I		PLB Write Pending Priority
PLB_rdPendPri[0:1]	I		PLB Read Pending Priority
PLB_reqPri[0:1]	I		PLB Request Primary
PLB_TAttribute[0:15]	I		PLB Transaction Attribute
Sl_addrAck	O	0	PLB Slave Address Acknowledge
Sl_SSize[0:1]	O	0	PLB Slave Size
Sl_wait	O	0	PLB Slave Wait
Sl_rearbitrate	O	0	PLB Rearbitrate
Sl_wrDAck	O	0	PLB Write Data Acknowledge
Sl_wrComp	O	0	PLB Write Complete
Sl_wrBTerm	O	0	PLB Burst Terminate
Sl_rdDBus[0:C_SPLB_DWIDTH-1]	O	0	PLB Read Data Bus
Sl_rdWdAddr[0:3]	O	0	PLB Read Word Address
Sl_rdDAck	O	0	PLB Read Data Acknowledge
Sl_rdComp	O	0	PLB Read Complete
Sl_rdBTerm	O	0	PLB Read Burst Terminate
Sl_MBusy[0:C_SPLB_NUM_MASTERS-1]	O	0	PLB Slave Master Busy
Sl_MWrErr[0:C_SPLB_NUM_MASTERS-1]	O	0	PLB Master Write Error
Sl_MRdErr[0:C_SPLB_NUM_MASTERS-1]	O	0	PLB Read Error
Sl_MIRQ[0:C_SPLB_NUM_MASTERS-1]	O	0	PLB Master Interrupt Request
nRESET	O	0	Encoder/Decoder Reset Signal
ENC_P[0:7]	O	0	Encoder Data
ENC_SCRSTnRTC	O	DEC_SFL	Encoder Real Time Control Signal
ENC_CLK_27M	I		Encoder Clock Input
ENC_CLOCK	O	not ENC_CLK_27M	Encoder Clock Output
DEC_P[0:7]	I		Decoder Data
DEC_LLC	I		Decoder Line Locked Clock Input
DEC_SFL	I		Decoder Subcarrier Frequency Lock
SIV_Intr	O	0	Active Level High Interrupt Signal
R_MCH_Access_Control	O	0	MCH Read I/F Control Signal to the Access Buffer
R_MCH_Access_Data[0:31]	O	0	MCH Read I/F Write Data to the Access Buffer
R_MCH_Access_Write	O	0	MCH Read I/F Write Signal to the Access Buffer
R_MCH_Access_Full	I		MCH Read I/F Indicator that the Access Buffer
R_MCH_ReadData_Control	I		MCH Read I/F Control Signal for the ReadData Buffer
R_MCH_ReadData_Data[0:31]	I		MCH Read I/F Read Data from the ReadData Buffer
R_MCH_ReadData_Read	O	0	MCH Read I/F Read Signal to the ReadData Buffer
R_MCH_ReadData_Exists	I		MCH Read I/F Indicator that the ReadData Buffer
W_MCH_Access_Control	O	0	MCH Write I/F Control Signal to the Access Buffer
W_MCH_Access_Data[0:31]	O	0	MCH Write I/F Write Data to the Access Buffer
W_MCH_Access_Write	O	0	MCH Write I/F Write Signal to the Access Buffer
W_MCH_Access_Full	I		MCH Write I/F Indicator that the Access Buffer
W_MCH_ReadData_Control	I		MCH Write I/F Control Signal for the ReadData Buffer
W_MCH_ReadData_Data[0:31]	I		MCH Write I/F Read Data from the ReadData Buffer
W_MCH_ReadData_Read	O	0	MCH Write I/F Read Signal to the ReadData Buffer
W_MCH_ReadData_Exists	I		MCH Write I/F Indicator that the ReadData Buffer

Register

XPS-SIV00 のレジスタを示します。

Table 3 : Register Overview

Register Name	Description	PLB Address	Reset Value	Access
Control	Control Register	C_MEM0_BASEADDR+0x00	0x0000 0000	Read/Write
Status	Status Register	C_MEM0_BASEADDR+0x10	0x0000 0000	Read
Status Clear	Status Clear Register	C_MEM0_BASEADDR+0x14	0x0000 0000	Write
Interrupt Enable	Interrupt Enable Register	C_MEM0_BASEADDR+0x40	0x0000 0000	Read/Write
Effect Selection	Effect Selection Register	C_MEM0_BASEADDR+0x60	0x0000 0000	Read/Write
Capture Mode Selection	Capture Mode Selection Register	C_MEM0_BASEADDR+0x64	0x0000 0001	Read/Write
Capture Address	Capture data DMA Address Register	C_MEM0_BASEADDR+0x80	0x81E0 0000	Read/Write
Display Address	Display data DMA Address Register	C_MEM0_BASEADDR+0x84	0x81E0 0000	Read/Write
Core ID	Core ID Register	C_MEM0_BASEADDR+0xFC	0x5007100A	Read

※ 32bit アクセスのみ対応。8bit 及び 16bit アクセスはできません。

• Control (Control Register) Offset:0x00

XPS-SIV00 の動作を制御します。

0 27	28	29	30	31
Unused	CORE_RST	BOARD_RST	CAPTURE	DISPLAY

MSB LSB

Figure 2 : Control Register

Table 4 :Control Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 27	Unused	Read	Unused	zeroes
28	CORE_RST	Read/Write	1 : IP コアへ Reset 信号を出力します。 0 ; IP コアへの Reset 信号の出力を停止します。	0
29	BOARD_RST	Read/Write	1 : エンコーダ、デコーダに Reset 信号を出力します。 0 : エンコーダ、デコーダへの Reset 信号の出力を停止します。	0
30	CAPTURE	Read/Write	1 : 画像取得を開始します。取得したデータは Capture Address Register に設定したアドレスに転送します。エンコーダ IC、デコーダ IC への初期化を行ってから 1 にしてください。 0 : 画像取得を停止します。	0
31	DISPLAY	Read/Write	1 : 画像出力を開始します。Display Address Register に設定したアドレスからデータを読み出し、エンコーダ IC への出力を開始します。エンコーダ IC、デコーダ IC への初期化を行ってから 1 にしてください。 0 : 画像出力を停止します。	0

• Status (Status Register) Offset:0x10

XPS_SIV00 の状態を表示します。

0	30	31
Unused		CAP_DONE

MSB LSB

Figure 3 : Status Register

Table 5 : Status Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 30	Unused	Read	Unused	zeroes
31	CAP_DONE	Read	キャプチャ状態を表示します。 1 : 1 フレームキャプチャ終了。 0 : キャプチャ中	0

• Status Clear (Status Clear Register) Offset:0x14

Status Register のクリアを行います。

0	30	31
Unused		CAP_DONE_CLEAR

MSB LSB

Figure 4 : Status Clear Register

Table 6 : Status Clear Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 30	Unused	Read	Unused	zeroes
31	CAP_DONE_CLEAR	Write	1 : Status Register の CAP_DONE ビットを 0 にする信号を出力します。1 にした後は 0 に戻してください。 0: 信号の出力を停止します。	0

• Interrupt Enable (Interrupt Enable Register) Offset:0x40

割込みの有効無効を制御します。

0	30	31
Unused		CAP_DONE_EN

MSB LSB

Figure 5 : Status Register

Table 7 :Status Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 30	Unused	Read	Unused	zeroes
31	CAP_DONE_EN	Read/Write	Status Register の CAP_DONE が 1 になった時の割り込み 1： 有効 0： 無効	0

• Effect Selection (Effect Selection Register) Offset:0x60

データにエフェクトを追加します。

0	29	30	31
Unused		TYPE_SEL	

MSB LSB

Figure 6 : Display Select Register

Table 8 : Display Select Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 29	Unused	Read	Unused	zeroes
30:31	TYPE_SEL	Read/Write	取得する画像データのエフェクトを選択します。 00： カラー(スルー) 01： 白黒(2 値化) 10： 濃淡(グレースケール) 11： エッジ(エッジ検出)	00

- **Capture Mode Selection (Capture Mode Selection Register) Offset:0x64**

表示するデータのタイプを選択します。

0	30	31
Unused		CAP_MODE

MSB LSB

Figure 7 : Display Select Register

Table 9 : Display Select Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 30	Unused	Read	Unused	zeroes
31	CAP_MODE	Read/Write	取得モードを選択します。 1： 連続で画像データを取得します。 0 : Status Register の CAP_DONE ビットが 1 の時に画像データ取得を停止します。	1

- **Capture Address Register (Capture data DMA Address Register) Offset:0x80**

キャプチャしたデータを DMA 転送するアドレスをセットします。64Byte 境界でアラインしてください。

0	31
CAP_ADDR	

MSB LSB

Figure 8 : Capture Address Register

Table 10 : Capture Address Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 31	CAP_ADDR	Read/Write	キャプチャしたデータの DMA 転送先アドレスをセットします。	0x81E0 0000

- **Display Address Register (Display data DMA Address Register) Offset:0x84**

表示するデータを DMA 転送するアドレスをセットします。64Byte 境界でアラインしてください。

0	31
DISP_ADDR	

MSB LSB

Figure 9 : Display Address Register

Table 11 : Display Address Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 31	DISP_ADDR	Read/Write	表示するデータの DMA 転送元アドレスをセットします。	0x81E0 0000

- **Core ID (Core ID Register) Offset:0xFC**

本 IP コアの ID です。

0	31
CORE_ID	

MSB LSB

Figure 10 : Core ID Register

Table 12 : Core ID Register Description

Bits	Name	Core Access	Description	Reset Value
0 : 31	CORE_ID	Read	本 IP コアの ID です。	0x5007100A

Functional Description

大きく分けて以下の 2 種類の機能を持ちます。

- 画像取得
- 画像表示

● 画像取得

画像取得は同期検出、エフェクト、デコーダ FIFO、YCbCr->RGB 変換、MCH Write FIFO、MCH Write I/F ブロックから成り立ちます。

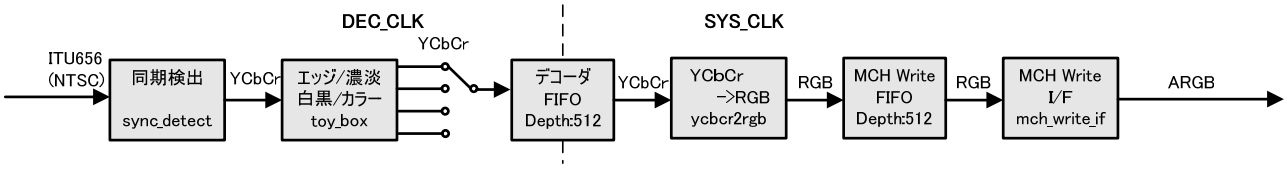

Figure 11 : XPS-SIV00 Block Diagram (Write Side)

● 同期検出ブロック

同期検出ブロックでは DEC_CLK 同期の ITU-656 信号に重畳されている同期信号を抽出し、ブランク信号の除去を行い、Y,Cb,Cr 各 8bit で順次送られてくるピクセルデータを 24bit に連接し、抽出した同期信号を付加してデータを送り出します。デコーダ FIFO への書き込みタイミングチャートを以下に示します。

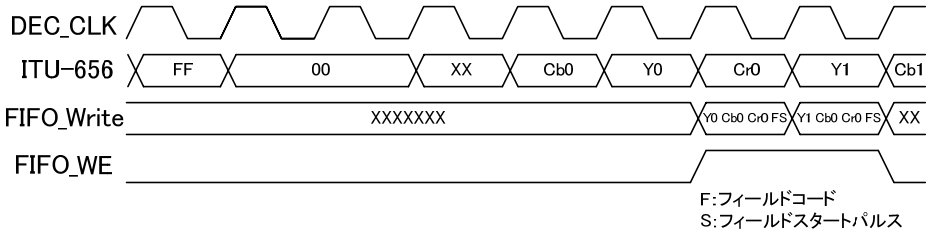

Figure 12 : Decoder FIFO Write Timing chart

同期信号として、トップおよびボトムフィールドを示すフィールドコードと、フィールドの先頭ピクセルを示すフィールドスタートビットの 2bit を生成します。フィールドコードはトップフィールドを 0、ボトムフィールドを 1 とします。フィールドスタートビットは各フィールドの有効ピクセルの先頭を書き込む時 1 を、それ以外は 0 を書き込みます。書き込みフォーマットを以下に示します。

Table 13 : Decoer FIFO Writing Format

0 7	8 15	16 23	24	25
Cb	Y	Cr	フィールドコード	フィールドスタート

- エフェクトブロック

エッジ検出/濃淡/白黒/カラー画像生成ブロックでは送られてきた YCbCr ピクセルデータをスルー(カラー画像)もしくはエッジ検出、濃淡、白黒画像データに変換してデコーダ FIFO に送り込みます。

- デコーダ FIFO ブロック

デコーダ FIFO は入出力が 26bit 幅のデュアルクロック FIFO で、DEC_CLK と SYS_CLK の乗せ換えを行います。

- YCbCr→RGB 変換ブロック

YCbCr→RGB 変換ブロックは FIFO からリードした YCbCr ピクセルデータを 24bit RGB データに変換し、変換に必要なレイテンシを加えた同期信号をあわせて 26bit のデータを MCH Write FIFO ブロックへ書き込みます。

Table 14 : YCrCb->RGB Conversion Block Output format

0 7	8 15	16 23	24	25
R	G	B	フィールドコード	フィールドスタート

- MCH Write FIFO ブロック

MCH Write FIFO は入出力が 26bit 幅の FIFO で、MCH Write I/F ブロックがバースト転送するためのバッファです。

- MCH Write I/F ブロック

MCH Write I/F ブロックは画像取得レジスタに 1 がセットされると、フィールドスタートパルスを待ちます。スタートパルスを検出すると、YCbCr→RGB 変換ブロックから MCH Write FIFO ブロックへ書き込まれた RGB データを ARGB 形式に拡張し(A=0x00)、MCH OPB SDRAM Controller へ出力します。このとき RGB データはインターレースで送られてくるので、SDRAM 上でノンインターレース画像になるよう、1 ライン毎にアドレスを指定しなければなりません。

1 フレーム分のデータを出力し終えたら、CAP_DONE レジスタに 1 をセットします。

Table 15 : MCH Write I/F Block Output format

0 7	8 15	16 23	24 31
A	R	G	B

● 画像表示

画像表示は MCH Read I/F、MCH Read FIFO、RGB->YCbCr 変換、エンコーダ FIFO、同期信号・ブランク信号挿入ブロックから成り立ちます。

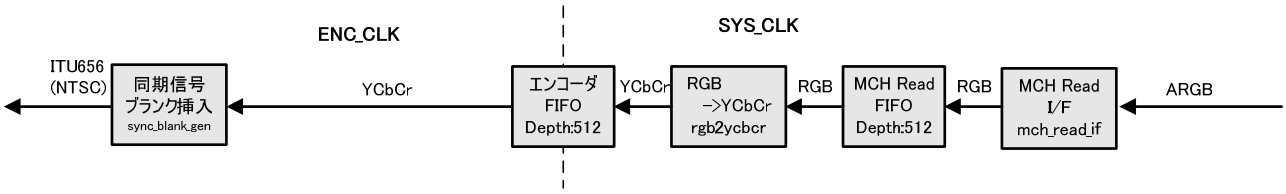

Figure 13 : XPS-SIV00 Block Diagram(Read Side)

● MCH Read I/F ブロック

MCH Read I/F ブロックは、DISPLAY レジスタに 1 がセットされると、DISP_ADDR レジスタにセットされたアドレスから、出力画像の 32bit ARGB データをバーストリードし、24bit RGB データに変換します。フィールドコード及びフィールドスタートビットを生成し、24bit RGB データと合わせた 26bit を MCH Read FIFO ブロックに書き込みます。入力フォーマットおよび出力フォーマットを以下に示します。

Table 16 : MCH Write I/F Block Input format

0 7	8 15	16 23	24 31
A	R	G	B

Table 17 : MCH Read I/F Block Output format

0 7	8 15	16 23	24	25
R	G	B	フィールドコード	フィールドスタート

● RGB→YCbCr 変換ブロック

このブロックは MCH Read FIFO にバッファされた 24bit RGB データを YCbCr ピクセルデータに変換し、変換に必要なレイテンシを加えた同期信号をあわせて 26bit YCbCr データをエンコード FIFO へ書き込みます。書き込みフォーマットを以下に示します。

Table 18 : RGB->YCbCr Conversion Block Output format

0 7	8 15	16 23	24	25
Cb	Y	Cr	フィールドコード	フィールドスタート

● エンコーダ FIFO ブロック

エンコーダ FIFO ブロックは、入出力が 26bit 幅のデュアルクロック FIFO で、SYS_CLK と ENC_CLK の乗せ換えを行います。

- 同期信号･ブランク挿入ブロック

このブロックはエンコード FIFO から YCbCr データと同期信号を読み出し、ITU-656 に準じた SAV 及び EAV とブランク信号を付加して出力します。ITU-656 の 1 ラインのデータは SAV、ピクセルデータ、EAV、ブランクデータで構成されます。

SAV と EAV は 4 バイトで構成され､NTSC の場合は以下に示すようにラインによって組み合わせが定義されています。

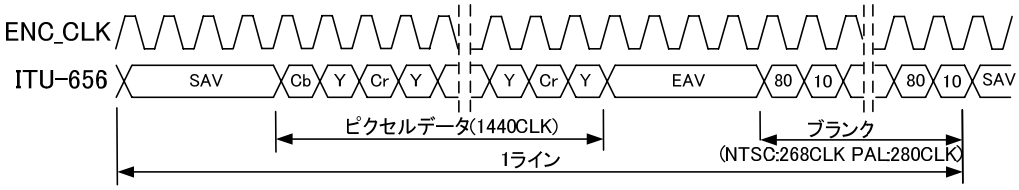

Figure 14 : ITU-656 1 line Data Format

Table 19 : Relation among line, SAV, and EAV(NTSC)

ライン	有効/無効	フィールド	SAV	EAV
1~3	無効	ボトム	FF0000EC	FF0000F1
4~20	無効	トップ	FF0000AB	FF0000B6
21~263	有効	トップ	FF000080	FF00009D
264~265	無効	トップ	FF0000AB	FF0000B6
266~283	無効	ボトム	FF0000EC	FF0000F1
284~525	有効	ボトム	FF0000C7	FF0000DA

ピクセルデータは SAV を出力後、エンコード FIFO からリードした 24bit YCbCr データを奇数ピクセルでは Cb、Y、Cr の順に出力し、偶数ピクセルでは Y のみを出力し、Cb,Cr は使用しません。

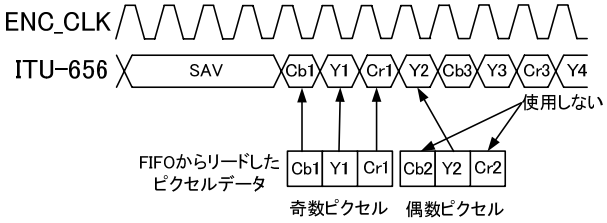

Figure 15 : Pixel Data Output

1 ライン分の有効ピクセル（720 ピクセル）を出力したあと EAV を出力し、ブランクデータとして 0x80、0x10 を交互に 268clk 繰り返し出力し､次のラインの SAV を出力します｡無効ラインのピクセルデータも 0x80、0x10 の繰り返しとします。エンコーダ FIFO が Empty 状態の場合は、0x00 を出力します。

Implementation

SUZAKU-I/O シリーズ AV ボードの IP コア(XPS-SIV00 および XPS-SIA00[2])を SUZAKU のデフォルトプロジェクトに組み込む際の手順を以下に示します。

1. IP Core のコピー

SUZAKU のデフォルトプロジェクト[3]の pcores フォルダ以下に xps_siv00_vx_xx_x、xps_sia00_vx_xx_x をコピーしてください。

sz410-yyyymmdd¥pcores¥xps_siv00_vx_xx_x

sz410-yyyymmdd¥pcores¥xps_sia00_vx_xx_x

(yyyymmdd : 日付　vx_xx_x : バージョン)

2. MHS File の編集

sz410-yyyymmdd¥xps_proj.mhs ファイルを開き以下のように編集してください。太字部分が変更もしくは追加された部分です。

```
# External Port 宣言の追加
 PORT IIC_DATA = IIC_DATA, VEC = [0:1], DIR = IO
 PORT nRESET = nRESET, DIR = O
 PORT ENC_P = ENC_P, DIR = O, VEC = [0:7]
 PORT ENC_SCRSTnRTC = ENC_SCRSTnRTC, DIR = O
 PORT ENC_CLK_27M = ENC_CLK_27M, DIR = I, SIGIS = Clk
 PORT ENC_CLOCK = ENC_CLOCK, DIR = O, SIGIS = Clk
 PORT DEC_P = DEC_P, DIR = I, VEC = [0:7]
 PORT DEC_LLC = DEC_LLC, DIR = I, SIGIS = Clk
 PORT DEC_SFL = DEC_SFL, DIR = I
 PORT COD_DIN = COD_DIN, DIR = O
 PORT COD_LRCIN = COD_LRCIN, DIR = I
 PORT COD_DOUT = COD_DOUT, DIR = I
 PORT COD_BCLK = COD_BCLK, DIR = I

# IP コア XPS_GPIO の追加[4]
BEGIN xps_gpio
 PARAMETER INSTANCE = i2c
 PARAMETER HW_VER = x.xx.x          # x.xx.x : ご使用になるコアのバージョン
 PARAMETER C_BASEADDR = 0xF0FF4600
 PARAMETER C_HIGHADDR = 0xF0FF47FF
 PARAMETER C_GPIO_WIDTH = 2
 PARAMETER C_IS_BIDIR = 1
 PARAMETER C_TRI_DEFAULT = 0x00000000
 BUS_INTERFACE SPLB = plb_peripheral
 PORT GPIO_IO = IIC_DATA
END
```

[2] XPS-SIA00 は SUZAKU I/O シリーズ AV ボードの Audio 用 IP コアです。詳細につきましては、XPS-SIV00 のデータシートをご参照ください。
[3] SUZAKU CD-ROM 20080118 以降の ISE/EDK9.2i プロジェクトをご使用ください。
[4] エンコーダ IC、デコーダ IC、コーデック IC の初期化を I²C で行うため XPS_GPIO が必要となります。

```
# IP コア XPS-SIV00 の追加
BEGIN xps_siv00
 PARAMETER INSTANCE = siv_cntlr
 PARAMETER HW_VER = x.xx.x          # x.xx.x : ご使用になるコアのバージョン
 PARAMETER C_MEM0_BASEADDR = 0xF0FF4200
 PARAMETER C_MEM0_HIGHADDR = 0xF0FF43FF
 PARAMETER C_CAP_ADDR = 0x00000000
 PARAMETER C_DISP_ADDR = 0x00000000
 BUS_INTERFACE SPLB = plb_peripheral
 BUS_INTERFACE W_XCL = siv_cntlr_w_xcl
 BUS_INTERFACE R_XCL = siv_cntlr_r_xcl
 PORT nRESET = nRESET
 PORT ENC_P = ENC_P
 PORT ENC_SCRSTnRTC = ENC_SCRSTnRTC
 PORT ENC_CLK_27M = ENC_CLK_27M
 PORT ENC_CLOCK = ENC_CLOCK
 PORT DEC_P = DEC_P
 PORT DEC_LLC = DEC_LLC
 PORT DEC_SFL = DEC_SFL
 PORT SIV_Intr = siv_intr
END

# IP コア XPS-SIA00 の追加
BEGIN xps_sia00
 PARAMETER INSTANCE = sia_cntlr
 PARAMETER HW_VER = x.xx.x          # x.xx.x : ご使用になるコアのバージョン
 PARAMETER C_MEM0_BASEADDR = 0xF0FF4400
 PARAMETER C_MEM0_HIGHADDR = 0xF0FF45FF
 BUS_INTERFACE SPLB = plb_peripheral
 PORT COD_DIN = COD_DIN
 PORT COD_LRCIN = COD_LRCIN
 PORT COD_DOUT = COD_DOUT
 PORT COD_BCLK = COD_BCLK
 PORT SIA_Intr = sia_intr
END

# IP コア MPMC_SZ410 の変更
BEGIN mpmc_sz410
 PARAMETER INSTANCE = mpmc_ddr2
 PARAMETER HW_VER = x.xx.x          # x.xx.x : ご使用になるコアのバージョン
 PARAMETER C_MEM_PARTNO = MT47H16M16-37E
 PARAMETER C_MPMC_CLK0_PERIOD_PS = 5714
 PARAMETER C_MPMC_BASEADDR = 0x00000000
 PARAMETER C_MPMC_HIGHADDR = 0x03FFFFFF
 PARAMETER C_MEM_DATA_WIDTH = 32
 PARAMETER C_DDR2_DQSN_ENABLE = 1
 PARAMETER C_MEM_CLK_WIDTH = 2
 PARAMETER C_PIM1_BASETYPE = 1
 PARAMETER C_PIM2_BASETYPE = 1
 PARAMETER C_XCL1_LINESIZE = 8
 PARAMETER C_XCL1_WRITEXFER = 2
 PARAMETER C_XCL2_LINESIZE = 16
 PARAMETER C_NUM_PORTS = 3
 PARAMETER C_PI1_RD_FIFO_TYPE = SRL
 PARAMETER C_PI1_WR_FIFO_TYPE = SRL
```

```
 PARAMETER C_PI2_RD_FIFO_TYPE = SRL
 PARAMETER C_PI2_WR_FIFO_TYPE = SRL
 BUS_INTERFACE SPLB0 = plb_memory
 BUS_INTERFACE XCL1 = siv_cntlr_w_xcl
 BUS_INTERFACE XCL2 = siv_cntlr_r_xcl
 PORT FSL1_M_Clk = sys_clk_s
 PORT FSL2_M_Clk = sys_clk_s
 PORT MPMC_Clk90 = DDR_SDRAM_64Mx32_mpmc_clk_90_s
 PORT MPMC_Clk0 = DDR_SDRAM_64Mx32_mpmc_clk_0_s
 PORT MPMC_Clk_200MHz = clk_200mhz_s
 PORT DDR2_DQS_n = DDR_DQSn
 PORT DDR2_DQS = DDR_DQS
 PORT DDR2_DM = DDR_DM
 PORT DDR2_DQ = DDR_DQ
 PORT DDR2_Addr = DDR_Addr
 PORT DDR2_BankAddr = DDR_BankAddr
 PORT DDR2_WE_n = DDR_WEn
 PORT DDR2_CAS_n = DDR_CASn
 PORT DDR2_RAS_n = DDR_RASn
 PORT DDR2_ODT = DDR_ODT
 PORT DDR2_CS_n = DDR_CSn
 PORT DDR2_CE = DDR_CKE
 PORT DDR2_Addr_2 = DDR_Addr_2
 PORT DDR2_BankAddr_2 = DDR_BankAddr_2
 PORT DDR2_WE_n_2 = DDR_WEn_2
 PORT DDR2_CAS_n_2 = DDR_CASn_2
 PORT DDR2_RAS_n_2 = DDR_RASn_2
 PORT DDR2_ODT_2 = DDR_ODT_2
 PORT DDR2_CS_n_2 = DDR_CSn_2
 PORT DDR2_CE_2 = DDR_CKE_2
 PORT DDR2_Clk_n = DDR_CLK_OUTn
 PORT DDR2_Clk = DDR_CLK_OUT
 PORT MPMC_Rst = sys_bus_reset
END

# IP コア XPS_INTC の変更
BEGIN xps_intc
 PARAMETER INSTANCE = intc_system
 PARAMETER HW_VER = x.xx.x          # x.xx.x : ご使用になるコアのバージョン
 PARAMETER C_BASEADDR = 0xF0FF3000
 PARAMETER C_HIGHADDR = 0xF0FF30FF
 BUS_INTERFACE SPLB = plb_peripheral
 PORT Irq = EICC405EXTINPUTIRQ
 PORT Intr = phy_mii_int&console_uart_interrupt&fifo_int&siv_intr&sia_intr
END
```

3. MSS File の編集

sz410-yyyymmdd¥xps_proj.mss ファイルを開き以下のように編集してください。太字部分が変更もしくは追加された部分です。

```
# XPS_GPIO のドライバの宣言を追加
BEGIN DRIVER
 PARAMETER DRIVER_NAME = gpio
 PARAMETER DRIVER_VER = x.xx.x          # x.xx.x : ご使用になるドライバのバージョン
 PARAMETER HW_INSTANCE = i2c
END

# XPS_SIV00 のドライバの宣言を追加
BEGIN DRIVER
 PARAMETER DRIVER_NAME = generic
 PARAMETER DRIVER_VER = 1.00.a
 PARAMETER HW_INSTANCE = siv_cntlr
END

# XPS_SIA00 のドライバの宣言を追加
BEGIN DRIVER
 PARAMETER DRIVER_NAME = generic
 PARAMETER DRIVER_VER = 1.00.a
 PARAMETER HW_INSTANCE = sia_cntlr
END
```

4. UCF File の編集

sz410-yyyymmdd¥data¥xps_proj.ucf ファイルを開き以下の記述を追記してください。

```
# XPS_GPIO(IIC)用のピンアサイン追加
NET IIC_DATA<0>         LOC=E2   | IOSTANDARD = LVCMOS33 | PULLUP;
NET IIC_DATA<1>         LOC=D2   | IOSTANDARD = LVCMOS33 | PULLUP;

# XPS_SIV00 用のピンアサイン追加
NET DEC_SFL             LOC=D15  | IOSTANDARD = LVCMOS33;
NET DEC_P<0>            LOC=E15  | IOSTANDARD = LVCMOS33;
NET DEC_P<1>            LOC=F15  | IOSTANDARD = LVCMOS33;
NET DEC_P<2>            LOC=P4   | IOSTANDARD = LVCMOS33;
NET DEC_P<3>            LOC=P5   | IOSTANDARD = LVCMOS33;
NET DEC_P<4>            LOC=P1   | IOSTANDARD = LVCMOS33;
NET DEC_P<5>            LOC=P2   | IOSTANDARD = LVCMOS33;
NET DEC_P<6>            LOC=L2   | IOSTANDARD = LVCMOS33;
NET DEC_P<7>            LOC=M2   | IOSTANDARD = LVCMOS33;
NET DEC_LLC             LOC=Y7   | IOSTANDARD = LVCMOS33;
NET ENC_CLK_27M         LOC=W7   | IOSTANDARD = LVCMOS33;
# ISE/EDK10.1i 以降の場合 ENC_CLK_27M に CLOCK_DEDICATED_ROUTE = FALSE を追記
NET nRESET              LOC=H5   | IOSTANDARD = LVCMOS33;
NET ENC_SCRSTnRTC       LOC=U9   | IOSTANDARD = LVCMOS33;
NET ENC_CLOCK           LOC=V10  | IOSTANDARD = LVCMOS33;
NET ENC_P<0>            LOC=L1   | IOSTANDARD = LVCMOS33;
NET ENC_P<1>            LOC=M1   | IOSTANDARD = LVCMOS33;
NET ENC_P<2>            LOC=G4   | IOSTANDARD = LVCMOS33;
```

```
NET ENC_P<3>            LOC=G5   | IOSTANDARD = LVCMOS33;
NET ENC_P<4>            LOC=G2   | IOSTANDARD = LVCMOS33;
NET ENC_P<5>            LOC=F2   | IOSTANDARD = LVCMOS33;
NET ENC_P<6>            LOC=F1   | IOSTANDARD = LVCMOS33;
NET ENC_P<7>            LOC=E1   | IOSTANDARD = LVCMOS33;

# XPS_SIA00 用のピンアサイン追加
NET COD_BCLK            LOC=N4   | IOSTANDARD = LVCMOS33;
# ISE/EDK10.1i 以降の場合 COD_BCLK に CLOCK_DEDICATED_ROUTE = FALSE を追記
NET COD_DOUT            LOC=M3   | IOSTANDARD = LVCMOS33;
NET COD_LRCIN           LOC=M4   | IOSTANDARD = LVCMOS33;
NET COD_DIN             LOC=H4   | IOSTANDARD = LVCMOS33;

# 使用を禁止にする
CONFIG PROHIBIT = E14;     #CON2_7  DEC_nINTRQ
CONFIG PROHIBIT = N2;      #CON2_17 COD_CLKOUT
```

Revision History

Date	Version	Revision
2008/01/18	1.0	初版作成
2008/03/14	1.1	誤記修正
2008/06/20	1.2	ISE/EDK10.1 対応 UCF に制約(CLOCK_DEDICATED_ROUTE = FALSE)追記