Notes on ES24D1B Board Design

©Duy-Ky Nguyen 00-06-01

Table of Contents

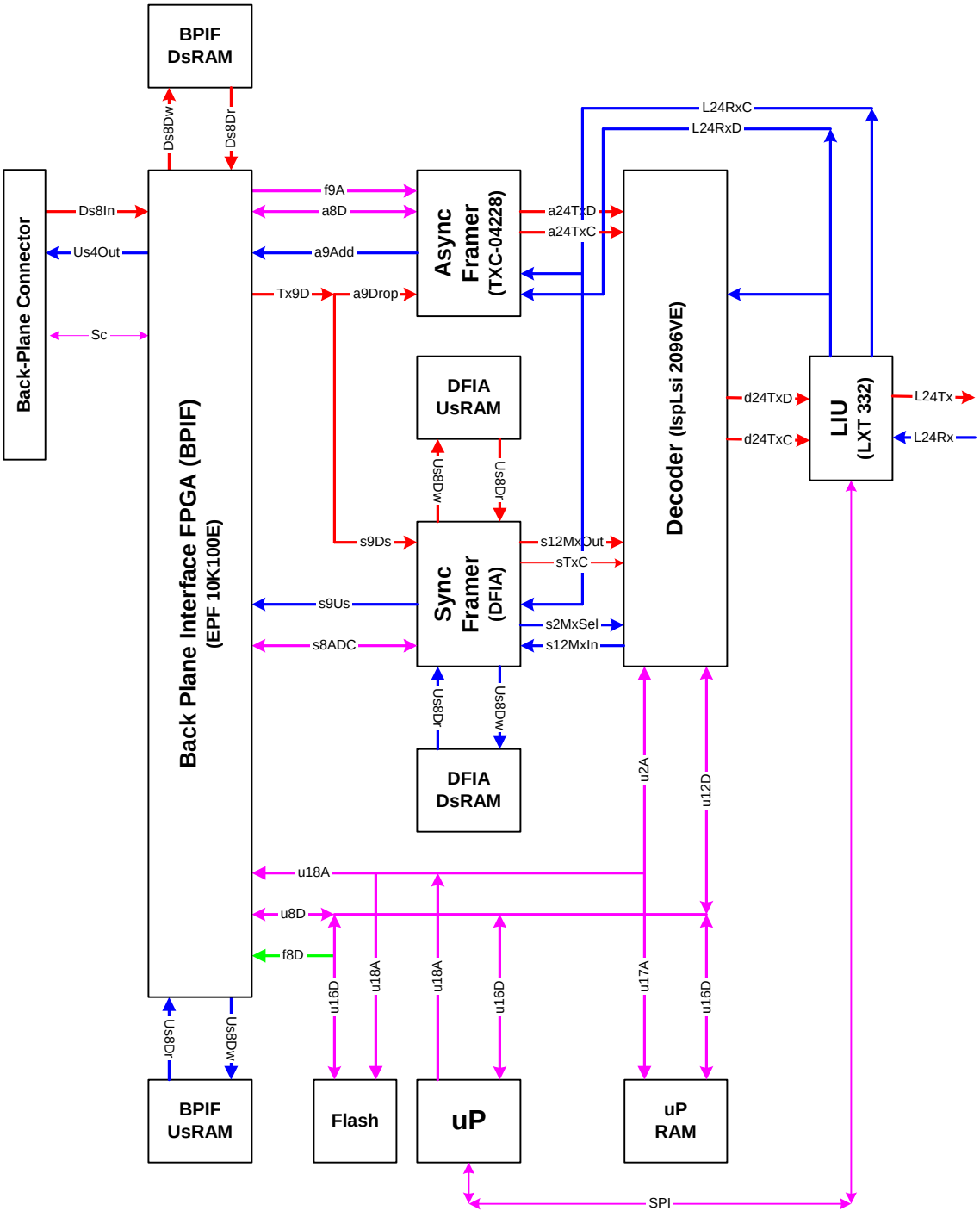
1.1.1. Abstract

This document specifies the function of USAM Expansion Shelf 24 Async/Sync DS1 Unit (ES24D1). At the boundary, signal is either in synchronous TDM format or in asynchronous SONET VT1.5 format at one side (with Back-Plane), either in DSX format or in VT1.5 format, respectively, at the other side (with 24 DS1 Line). This choice is set by uP.

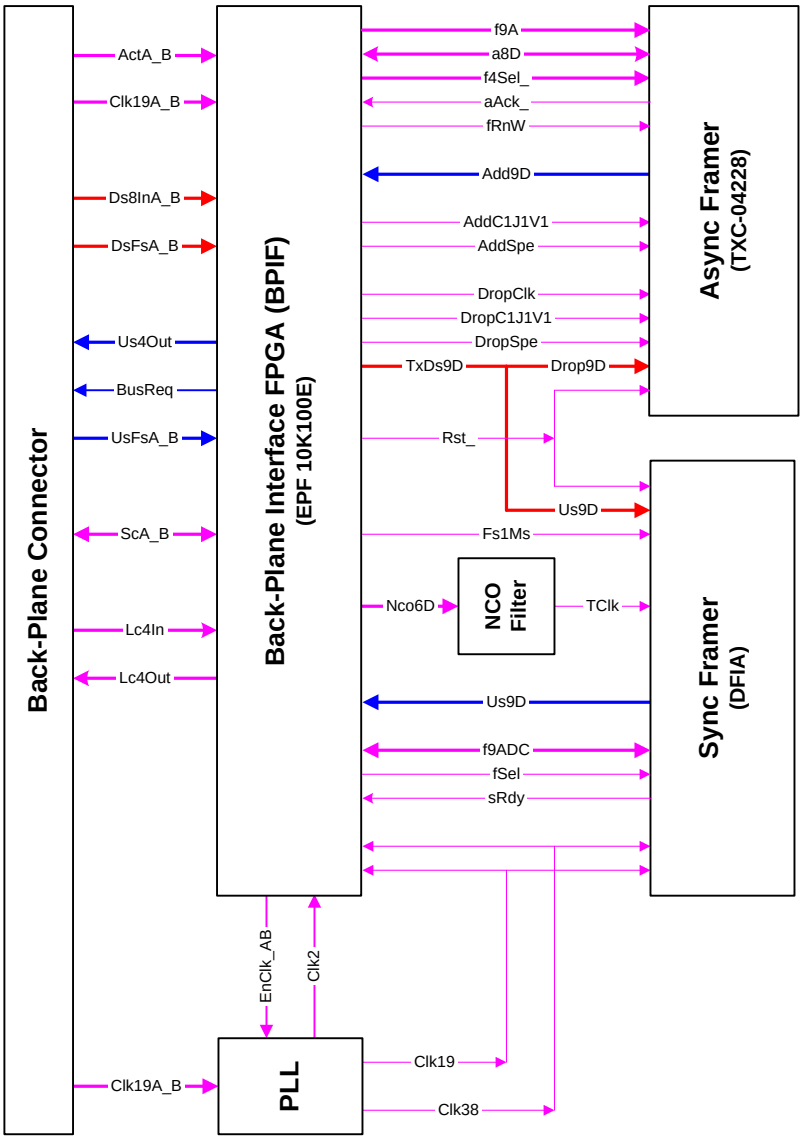
2. Acronyms

ADC	Address-Data Channel
ASIC	Application Specific Integrated Circuit
BP	Back-Plane
CuP	Central uP
DS	Down-Stream
DS0	Digital Signal 0
DS1	Digital Signal 1
DSX	Digital Signal Cross-Connect
DFIA	DS1 Framer and Interface ASIC
DuP	DFIA uP
HDLC	High-Level Data-Link Channel
LIU	Line-Card Interface Unit
LOH	Line Over-Head
POH	Path Over-Head
SL	Serial Link
SOH	Section Over-Head
TDM	Time Division Multiplexing
TOH	Transport Over-Head
US	Up-Stream
VT	Virtual Tributary

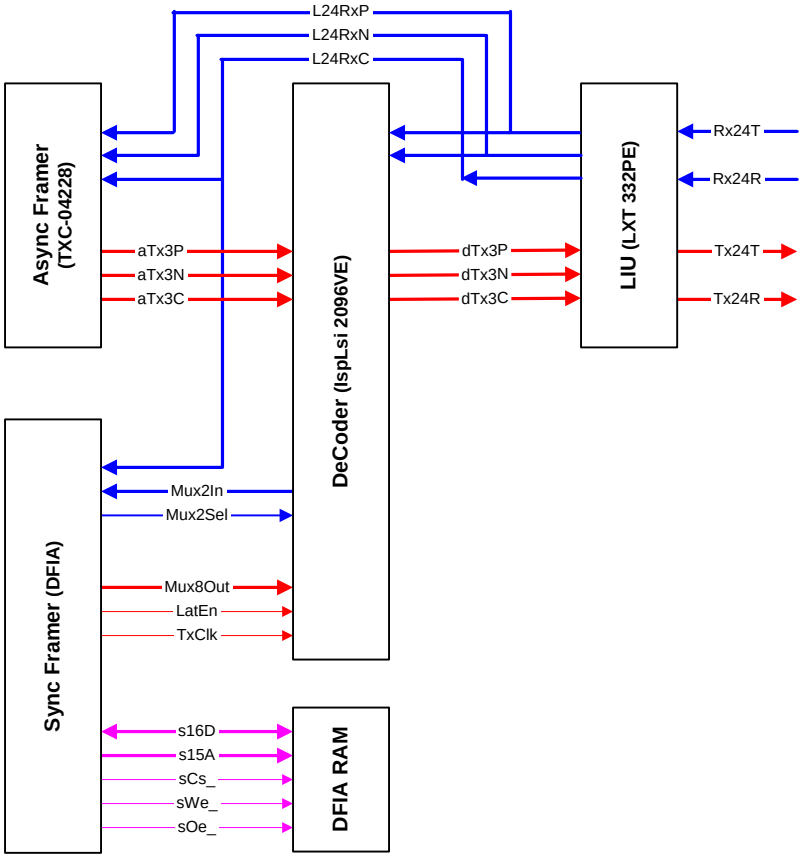
3. Block Diagram



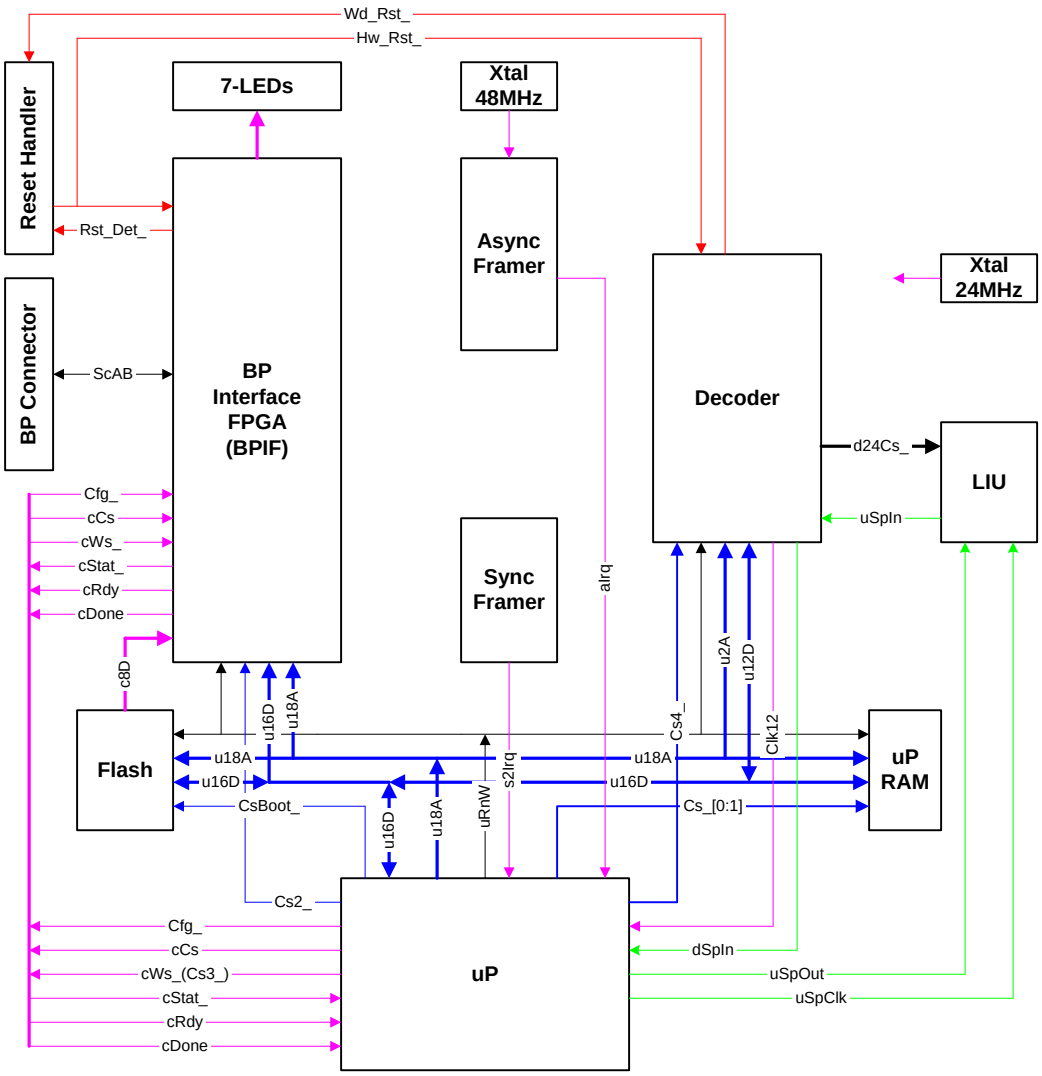
4. Back-Plane FPGA Interface



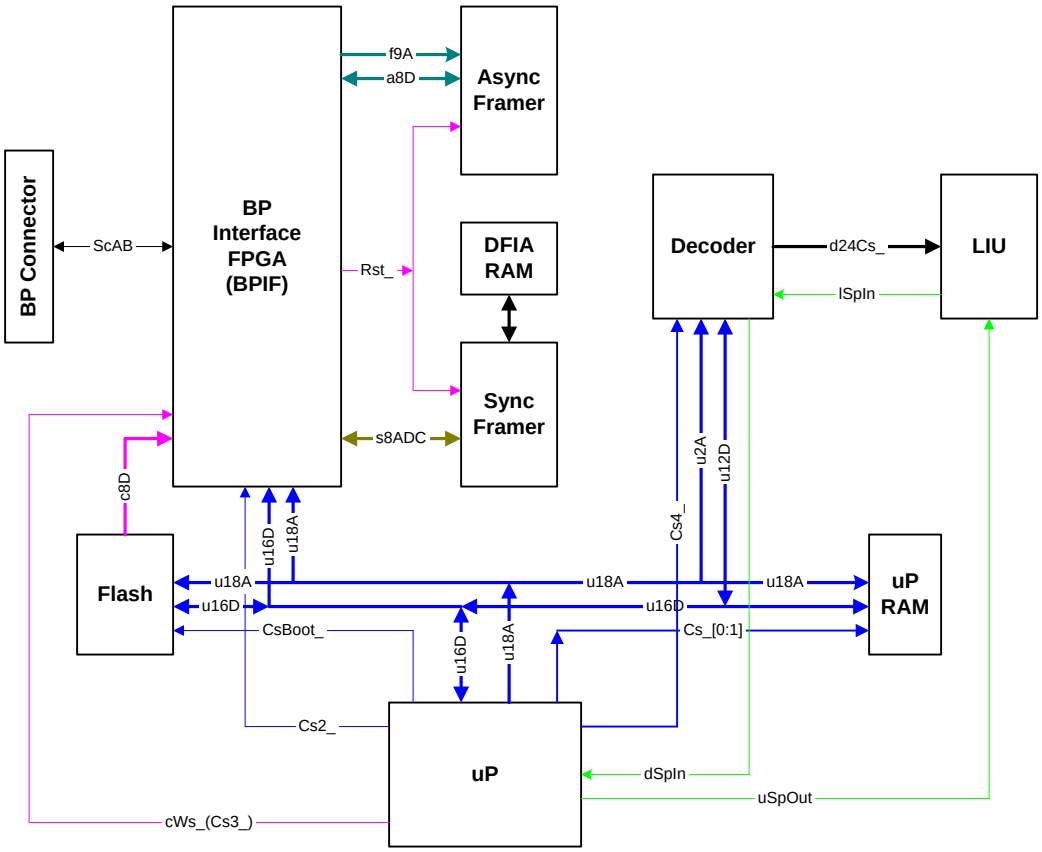
5. Logic Decoder Interface



6. Control Interface



7. Resource Accessibility



uP accesses to its RAM via its bus using Cs0_ and Cs1_ for low and high memory bank. It accesses to BPIF using Cs2_. It accesses to CPLD Decoder using Cs4_. It accesses to LIU's (for LIU initialization) by Motorola SPI protocol via Decoder CPLD for Cs. It accesses to SMCC via the serial link protocol.

BPIF accesses to Async framer via a separate A/D.

The 8-bit ADC protocol is implemented between BPIF and DFIA so BPIF can access to DFIA RAM via this link. The uP can access DFIA RAM via uP A/D buses and ADC link.

SMCC accesses to BPIF and then BPIF RAM via the serial link protocol. It can then access to DFIA and DFIA RAM through the ADC protocol.

SMCC cannot directly access uP RAM, and LIU; but it can indirectly access via the Mail-Box protocol using DL RAM @ 0x4000 of FPGA.

8. Memory Map

8.1. Map for MC68LK331 @ 12 MHz

Absolute Base Address	Devices	Size (Byte)	Width (Bit)	Type	Chip Select	Ack (Wait-State)
$00 0000	Flash	512 K	16	RW	CSBOOT	0(1)
$08 0000	uP RAM	256 K	16	RW	CS0: Lo-Byte CS1:Hi-Byte	0(2)
$10 0000	FPGA	64 K	16	RW	CS2	External
$20 0000	FPGA Config	128 K	16	W	CS3	0(3)
$30 0000	CPLD	2 K	16	RW	CS4	0(4)

For 12 MHz, we have

$$t_{u\_cyc} = 84\,nS \qquad (1)$$

For CS of MC68LK331to generate internal DsAck, we have

$$t_{u\_rw} = (3 + n \times WS)\,t_{u\_cyc} = 252 + n \times WS \times t_{u\_cyc}\ \langle nS \rangle \qquad (2)$$

(1) A flash of 120 nS is used in this project

$$t_{f\_rd} \le 120\ \langle nS \rangle \qquad (3)$$

so the required condition is

$$t_{u\_rw} \ge 120\ \langle nS \rangle \qquad (4)$$

and we have 0 Wait-State to read this flash.

(2) For a RAM of 70 nS in reading

$$t_{m\_rd} \le 70\ \langle nS \rangle \qquad (5)$$

so the required condition is

$$t_{u\_rw} \ge 70\ \langle nS \rangle \qquad (6)$$

and we have 0 Wait-State to read this RAM.

To write this RAM

$$t_{m\_wr} \ge 10\ \langle nS \rangle \qquad (7)$$

so the required condition is

$$t_{u\_rw} \ge 10\ \langle nS \rangle \qquad (8)$$

and we have 0 Wait-State to write this RAM.

(3) For the Passive Parallel Asynchronous Configuration (PPA) of the FPGA in use, we have

$$t_{a\_wr} \ge 200\ \langle nS \rangle \qquad (9)$$

so the required condition is

$$t_{u\_rw} \ge 200\ \langle nS \rangle \qquad (10)$$

and we have 0 Wait-State to write into FPGA for configuration.

(4) For CPLD, data read are valid 2 clocks delay to avoid glitches, for clock of 12 MHz, we have

$$t_{d\_rd} = 2 \times t_{u\_cyc} = 2 \times 84 = 168\ \langle nS \rangle \qquad (11)$$

so the required condition is

$$t_{u\_rw} \ge 168\ \langle nS \rangle \qquad (12)$$

and we have 0 Wait-State to read this CPLD.

8.1.1. Flash (CS Boot @ $0)

Offset Base Address	Data	Type	Description
$00 0000	uP Boot (Top 512 K Flash)	R	Boot code for uP
$06 0000	FPGA Cfg Code (Bottom 512 K Flash)	W	Config code for FPGA (CS3)

8.1.2. CPLD Registers (CS4 @ $30,000)

Offset Base Address	Data	Type	Description
0x0000	15~3: Reserved 2~0 = 5	W	Watch Dog Service Reg Writing 5 to this reg at most 1.4 secs, otherwise being reset
0x0002	15~5: Reserved 4:0 = 0~23	R/W	LIU CS Reg for 24 LIU's
0x0004	15~12: Reserved 11~0 = 1: Async, 0: Sync	R/W	Async Sel Reg for Channels 1~12
0x0006	15~12: Reserved 11~0 = 1: Async, 0: Sync	R/W	Async Sel Reg for Channel 13~24

8.2. Memory Map for FPGA

Word access only. Byte implementation is in low byte of a word (Mask $00ff).

Offset Address	Size	Device	Comments
0x0000		Internal Registers	See BPI FPGA Register Map
0x1000	2K x 8	Down-Stream Buffer	CPU only accesses for powerup RAM test purpose. Byte implementation.
0x2000	4K x 8	Upstream Buffer	CPU only accesses for powerup RAM test purpose. Byte implementation.
0x4000	256 x 16	Data-Link Buffer	CC & CPU access for communication purpose. Word implementation.
0x5000	x 8	Async Mapper	Byte implementation.
0x6000	128 x 5	LUT	82 Half-Cell to 24 DS1 Mapping.

1.0.0.

8.2.1. FPGA Registers

Address	Register	Data After Reset	Type	Requirement
0x00	rScomm_En_1	15:8 = 0xa5 7:0 = NU	RW	Read Serial Link Reg 1 Wr Byte #1 to enable SMCC to read from BPIF via Serial Link
0x02	rScomm_En_2	15:8 = 0xc3 7:0 = NU	RW	Read Serial Link Reg 2 Wr Byte #2 to enable SMCC to read from BPIF via Serial Link
0x04	**rBoot_Code_type**	**0x0001**	**R**	**Boot Code embedded in LC (SW)**
0x06	rReset_Act_1	15:8 = 0x69 7:0 = NU	RW	Reset Line-Card Reg 1 Wr Byte #1 to reset Line Card
0x08	rReset_Act_2	15:8 = 0xe7 7:0 = NU	RW	Reset Line-Card Reg 2 Wr Byte #2 to reset Line Card
0x0a ~ 0x0c				Reserved
0x0e	rDs_PAR	0	RW	Ds Par Err sent from SMCC in cell (0,268) Wr 0xFFFF to clear
0x12		15:8 = 0 7:0 = NU	RW	Misc Info
0x14	rLC_Type	0x0004	R	Line Card Type (ES12D1)
0x16~ 0x28				Reserved
0x30	rScomm_Page	7:5 = NU 4:0 = A[17:13]	RW	Page Reg for 18-bit Address > 8K $A_{18} = A_{14}[13] ? \{ Page[4:0], A_{14}[12:0] \} : \{2'd0, A_{14}\};$ where A_{14} sent over serial link **Note**: This is for testing purpose with SComm and transparent to SW
0x32	rPar_Chk	15~2 = NU 1:0 = 0	RW	Enable Parity Check for Word access via Serial Link 1: Enable to check Ds parity via serial link 1: Enable to send Us parity check via serial link
0x34 ~ 0x46				Reserved
0x48	rFrame_OS	15:10 = NU 9:6 = 0xA 5:3 = 0 2:0 = 0	RW	Frame Offset Tx Frame Offset for Sync Tx V1 Id for Async Rx V1 Id for Async
0x4a	rLED	15:2 = NU 3 = 0 2 = 0 1 = 0 0 = 1	RW	LED Register 1: Fail 1: Active 1: Standby 1: Loopback Mode at least 1 channel
0x4c	rClk_Mon_Ref	15:4 = NU 3:0 = 0	RW	Clock Ref Select for 2 Outputs of BP Connector from Mon & Ref Clk of DFIA 3:2 = 11: Mon Clk to Output_1 3:2 = 10: Ref Clk to Output_1

				1:0 = 11: Mon Clk to Output_0 1:0 = 10: Ref Clk to Output_0
0x4d ~ 0x60				Reserved
0x62	rHw_Ctl	15:8 = NU 8 = 1 7 = 0 6 = 0 5 = 0 4 = 0 3 = 0 2 = 0 1 = 0 0 = 0	RW	Hardware Control Register 1: Disable Scomm access from CC 1: Normal operation. 0: Disable LIU; 0: Normal operation. 1: Force Parity Error in testing 0: Normal operation. 1: LB at FPGA for DsData = UsData in testing; 1: Normal operation. 0 to reset 1: Normal operation; 0 to disable writing into Us Buffers 1: Normal operation. 0 to disable writing into Ds Buffers 0: Sync mode; 1: Async mode 1: Start PCM operation
0x64	rHw_Status_rd	15:4 = NU 4 = 0 3 = 0 2 = 0 1 = 0 0 = 0	R	Instant Hardware Status Register 1: Us Par Err for bad UsTdm from Framer NU 1: PwrDown sent from SMCC in bit 6 of cell (4,0). 1: AIS sent from SMCC in bit 7 of cell (4,0) for bad Ds 1: Ds Par Err for bad Ds Data
0x66	rHw_Status	15:4 = NU 4 = 0 3 = 0 2 = 0 1 = 0 0 = 0	RW	Latched Hardware Status Register for the Instant Hw Status Reg above. Us Par NU PowerDown AIS Ds Par Err Write “1” to a bit position to clear this bit.
0x68	rHw_State_Ref	15:4 = N.U. 4 = 0 3 = 0 2 = 0 1 = 0 0 = 0	RW	Hardware State Reference Register User-defined PowerDown User-defined AIS User-defined ActA User-defined ActB User-defined Act
0x6A	rHw_State_Mask	15:4 = N.U. 4 = 0 3 = 0 2 = 0 1 = 0 0 = 0	RW	Hardware State Mask Register for HW State Ref Reg above PowerDown AIS ActA ActB Act 0: to mask out
0x6C	rHw_State_rd	15:4 = N.U. 4 = 0 3 = 0 2 = 0 1 = 0 0 = 0	R	Instant Hardware State Register PowerDown AIS ActA ActB Act 1: current state differs from defined state (@0x68) if not masked out (@0x6A) to generate IRQ (@0x70) if desired.
0x6E	rIrq_En	15:3 = N.U. 2 = 0 1 = 0 0 = 0	RW	Interrupt Enable Register 1: Enable State Change IRQ 1: Enable C2L IRQ from Central to Local uP 1: Enable L2C IRQ from Local to Central uP
0x70	rIrq_State	15:3 = N.U. 2 = 0 1 = 0 0 = 0	RW	Interrupt State Register 1: State change 1: Central uP interrupts Local uP via a cell in a Ds frame 1: Local uP interrupts Central uP via a cell in a Us frame
0x72	rAct_Sel	15:1 = N.U. 0 = 0	RW	Activity Select Register 1: Clear acitvities as a *seconadry* Reset
0x74	rIrq_Req	15:2 = N.U. 1 = 0 0 = 0	RW	Interrupt Request Register. 1: Activate LC to CC IRQ 1: Activate CC to LC IRQ
0x7A	rChip_Id	16'h5581	R	BPIF ID for E12D1
0x7C	rLocal_State	15:4 = N.U. 3 = 0 2:1 = 0 0 = 0	RW	Local Protection State register. Local Active Ack Local State Local Active Write 0xa7 to request. Read to check if bit 3 set. Write 0x58 to clear or 0x59 to activate if Remote in-active
0x7E	rRemote_State	15:4 = N.U. 3 = 0 2:1 = 0 0 = 0	RW	Remote (other side) Protection State register. Remote Active Ack Remote State Remote Active See Section on Protection Switching Control below.
0x82	rDL_Size	0x100	R	CC-LC Link Size
0x100	rDL_C2L	15:5 = NU 1 = 0 3:2 = 0 1 = 0 1 = 0	RW	Fr CC to LC Data Link CcDlEnable NU RxPkt2CcRdy Pkt2LcRdy
0x102	rDL_L2C	15:5 = NU 1 = 0 3:2 = 0 2 = 0 0 = 0	RW	Fr LC to CC Data Link LcDlEnable NU RxPkt2LcRdy Pkt2CcRdy
0x200	rAct_Ch_Lo	15:12= NU	RW	

		11:0 = 0		Set n-th bit to activate n-th DS1 channel
0x202	rAct_Ch_Hi	15:12= NU 11:0 = 0	RW	Set n-th bit to activate n-th DS1 channel
0x204	rAsync_Ch_Lo	15:12= NU 11:0 = 0	RW	Set n-th bit to put n-th DS1 channel in async mode. Otherwise sync mode.
0x206	rAsync_Ch_Hi	15:12= NU 11:0 = 0	RW	Set n-th bit to put n-th DS1 channel in async mode. Otherwise sync mode.

9. Testing Line-Card

9.1. Programming Flash

A 512Kb flash is divided into 2 regions, the first 384Kb is for uP boot code starting from addr 0, the last 128Kb for FPGA config code starting addr $60000. This flash is of Bottom Boot Block Sector in case using Sector Erase.

Boot code in S19 format is down-loaded into RAM from addr $80000 and write into flash from addr 0. Config code is downloaded into RAM from addr $e0000 and

FPGA config is under uP control. UP asserts CsBoot_ to read config code from flash (start from addr $60000) in words and asserts Cs3_ to write to FPGA in bytes, ie. 1 uP word-read word and 2 FPGA byte-write.

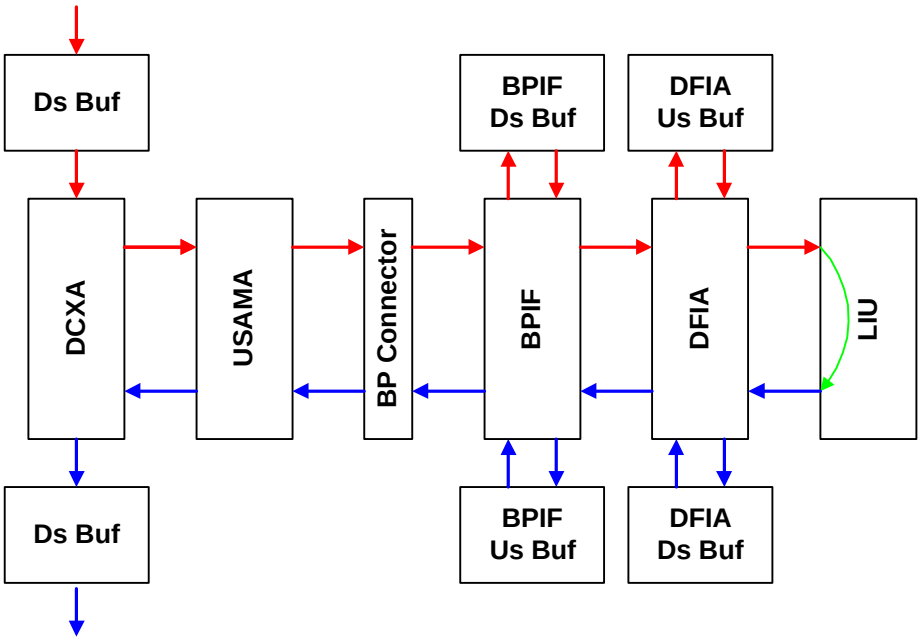
Remark 1: Disable Watch-Dog Timer in CPLD

It's required to write 5 into Watch-Dog Reg in CPLD in every 1.4 secs at most; otherwise the linecard will be reset. It takes more than 1.4 secs to download boot code or config code to RAM via BDM. This process is under BDM control, ie. out of user control. Therefore, it's required to disable the watch dog timer by installing a jumper across a 2x1 header.

9.2. Hardware Test

HW test will do

- RAM test;
- Traffic test with predefined patterns stored in a source buffer and pattern verifications in the remaining buffers.



9.2.1. Minimum BPIF Initialization

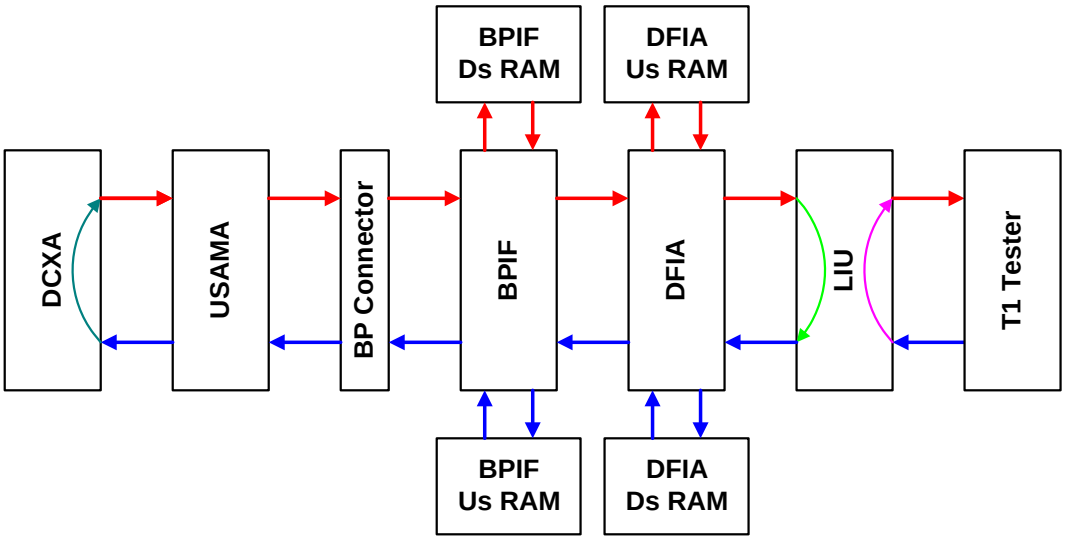
- Init Serial Link (SL): Common Control card (CC) can always write to this line-card (LC), but it cannot read from LC unless SL is initialized. In doing so, it is required to write $a5 @ 0 and $c3 @ 2 in BPIF;
- Init Half-Cell / DS1 Mapping: There are 24 among 82 half-cells selected for 24 DS1 channels. In doing so, the first 82 locations of LUT @ $6000 in BPIF need to be initialized by writing desired DS1 channel (0 ~ 23) into selected 24 locations (half-cells, say the first 24 ones), writing $FF into the remaining unselected 58 locations.

9.2.2. Minimum DFIA Initialization

- Init Signal Convesion Table @ $9000;
- Init Ds DS0 Ctl Reg $10 @ $c000;
- Init Us Trunk Cond Ctl Reg $2000 @$e000;
- Init Channel Ctl Reg $78 @ $1_000;
- Init Ds Protection Switching Id Reg 0 ~ 23 @ $1_0240
- Init Us Protection Switching Id Reg 0 ~ 23 @ $1_0280;
- Init Mode-2 TR-08 Loopback Inversion Control Registers 0 @ $1_0600 (Bug: Clear as unknown reset state !!!);
- Init Global Ctl Reg $01a6 @1_0680

9.2.3. Minimum Initialization of Async Mapper T1Mx28

9.3. Functional Test



Using VxWork via the serial link, SMCC will basically do the following

- reset line-card by writing 0x69 to @0x06 and 0xe7 to @0x08 in BPIF;
- enable SMCC to read BPIF via serial link by writing 0xa5 to @0x00 and 0xc3 to @0x02 in BPIF;
- initialize HDLC_ID by writing 0x5555 to 0x78 in BPIF;
- set line-card active by writing 0xa7 then 0x59 into Local Protection Ctl Reg @0x7c to set active bit for the linecard ready for En_Us/En_LIU to take effect once enabled in Hw Ctl Reg @0x62;
- set sync/async mode via 4 Ch_ID Regs in BPIF and Async_nSync Reg in CPLD;

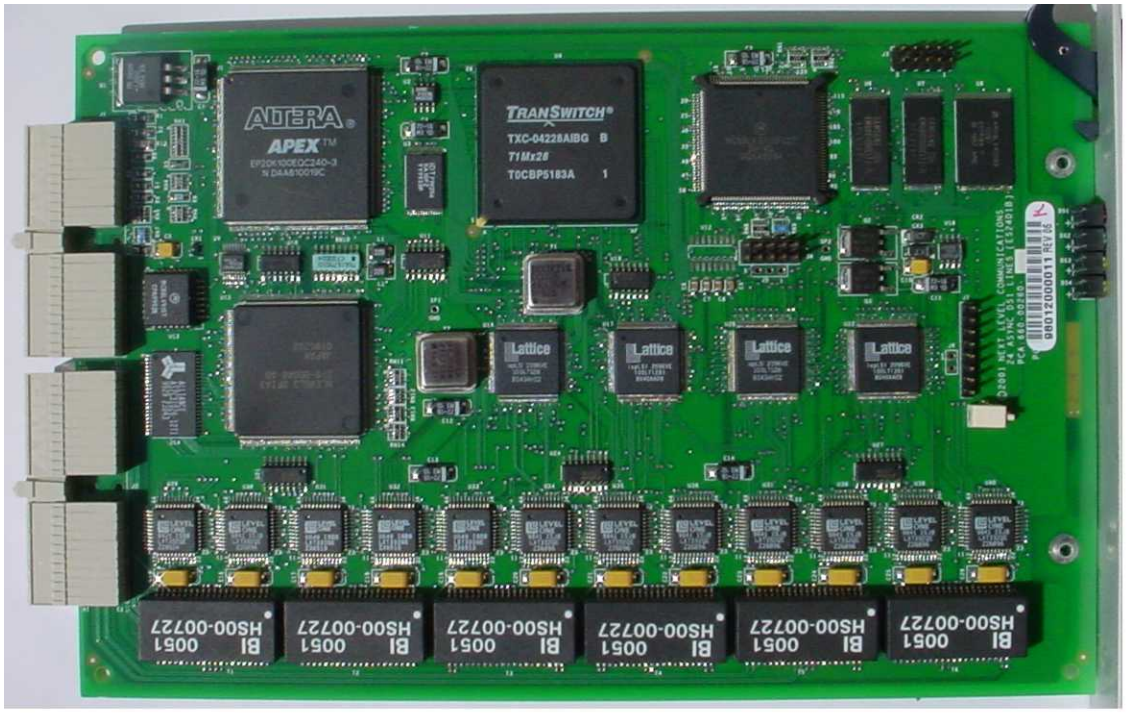
- initialize DFIA to have all 24 channels working with frame ESF of CRC-6, B8ZS code and 2nd column of the three is selected for TDM time-slot in a 27 row x 90 col frame;
- load boot code into DFIA RAM with code for Mail-Box protocol to establish a link between SMCC and uP (via serial link) and uP RAM testing code if so required;
- release reset for uP by setting HwCtlReg in DFIA;
- initialize LIU in dual loopback mode (local & remote loopback). Remote loopback mode will check line transformer;
- generate some random patterns as a reference and writes these patterns into Us Data Link Buf of DFIA; enables DFIA to send. Due to local loopback at LIU, these pattern will go to Ds Data Link Buf of DFIA. Read and verify this buf with reference to check the path from Us DFIA → local-loopbacked LIU → Ds DFIA;
- load a mapping between 82 half-cells and 24 DS1 into the DS1 Src Ptr (LUT) @ 0x01 1800;
- enable TDM slot in USAMA ready for DCXA to communicate with the linecard through USAMA;
- generate some random patterns as a reference and writes these patterns into Data Buf of DCXA; enables DCXA to send. Read and verify Ds Data Buf of BPIF with reference and verify with the reference to check the path from DCXA → USAMA → BP Connector → Ds BPIF. These patterns continue to go to Us DFIA, local loopbacked LIU, back to Ds DFIA and Us BPIF. Read and verify Us Data Buf of BPIF with the referenece to check this path. These patterns then continue to go back to DCXA. Read and verify Sig Buf of DCXA with the reference to check this last path;
- initialize LIU in normal mode (transparent without any loopback) and loopbacl DCXA to check the whole path T1-Tester → LIU → Ds DFIA → Us BPIF → USAMA → Loopbacked DCXA → USAMA → Ds BPIF → Us DFIA → LIU → T1-Tester.

10. Pins Description for EP20K100E

Signal	Description	Pin
ActA_ip	Select A / B side from CC USAM	4
ActB_ip		18
Adc8_op	ADC[8] as RnW or address[8] to DFIA	114
AdcCs_op	ADC Chip Select to DFIA	115
AdcRdy_ip	ADC Ready from DFIA	116
Adc_8bp[0]	ADC bus with DFIA	105
Adc_8bp[1]		106
Adc_8bp[2]		107
Adc_8bp[3]		109
Adc_8bp[4]		110
Adc_8bp[5]		111
Adc_8bp[6]		112
Adc_8bp[7]		113
AsA_C1J1_op	Add C1J1V1 to Async Mapper	68
AsA_Clk_op	Add Clk	66
AsA_SPE_op	Add SPE	69
AsAck_ip_	Ack from async mapper for uP interface	168
AsCs_4op_[0]	Chip Select	190
AsCs_4op_[1]		191
AsCs_4op_[2]		192
AsCs_4op_[3]		193
AsD_C1J1_op	Drop C1J1V1 to to Async Mapper	64
AsD_Clk_op	Drop Clk	63
AsD_SPE_op	Drop SPE	65
AsmA_9op[0]	Address to async mapper for uP interface	129
AsmA_9op[1]		130
AsmA_9op[2]		131
AsmA_9op[3]		133
AsmA_9op[4]		134
AsmA_9op[5]		136
AsmA_9op[6]		140
AsmA_9op[7]		143
AsmA_9op[8]		156
AsmD_8bp[0]	Data with async mapper for uP interface	119
AsmD_8bp[1]		121
AsmD_8bp[2]		123
AsmD_8bp[3]		124
AsmD_8bp[4]		125
AsmD_8bp[5]		126
AsmD_8bp[6]		127

AsmD_8bp[7]		128
AsmRd_op_	RnW to async mapper for uP interface	138
AsmWr_op_	Write Enable for RAM	167
Clk19A_ip	Clk19 from A side of CC USAMA	31
Clk19B_ip		34
Clk19EnA_op_	Enable Clk19 from A side	52
Clk19EnB_op_		53
Clk19_ip	Main Clk19	151
Clk38_ip	Main Clk38 for RAM access	154
Clk3_ip	Clk3 for clk act	209
ClkMon_ip	Clk Mon and Ref for DFIA	197
ClkRef_2op_[0]		201
ClkRef_2op_[1]		202
ClkRef_ip		198
DevRst_op_	Reset sync Framer and async Mapper	175
DsDataA_8ip[0]	Ds Data fromA side of CC USAM	6
DsDataA_8ip[1]		10
DsDataA_8ip[2]		15
DsDataA_8ip[3]		21
DsDataA_8ip[4]		5
DsDataA_8ip[5]		9
DsDataA_8ip[6]		14
DsDataA_8ip[7]		20
DsDataB_8ip[0]	B side	3
DsDataB_8ip[1]		8
DsDataB_8ip[2]		13
DsDataB_8ip[3]		17
DsDataB_8ip[4]		2
DsDataB_8ip[5]		7
DsDataB_8ip[6]		11
DsDataB_8ip[7]		16
DsFsA_ip	Composite Sync from A side	25
DsFsB_ip	B side	24
EnLIU_op	Enable LIU	194
Frz_ip_	Freeze to tristate IO pins	212
IrqC2L_op_	Irq	187
IrqH2L_op_	Irp for end of HDLC message	186
LED_4op[0]	Fail LED	170
LED_4op[1]	Active LED	171
LED_4op[2]	Standby LED	172
LED_4op[3]	Loopback LED	173
LcA_18ip[0]	Address for uP interface	239
LcA_18ip[10]		228
LcA_18ip[11]		227
LcA_18ip[12]		226
LcA_18ip[13]		225
LcA_18ip[14]		224
LcA_18ip[15]		223
LcA_18ip[16]		222
LcA_18ip[17]		221
LcA_18ip[1]		238
LcA_18ip[2]		237
LcA_18ip[3]		236
LcA_18ip[4]		235
LcA_18ip[5]		234
LcA_18ip[6]		233
LcA_18ip[7]		232
LcA_18ip[8]		231
LcA_18ip[9]		230
LcCs_ip_	Chip Select for uP interface	176
LcD_16bp[0]	Data	220
LcD_16bp[10]		195
LcD_16bp[11]		189
LcD_16bp[12]		185
LcD_16bp[13]		181
LcD_16bp[14]		169
LcD_16bp[15]		166
LcD_16bp[1]		219
LcD_16bp[2]		217
LcD_16bp[3]		216
LcD_16bp[4]		215
LcD_16bp[5]		207
LcD_16bp[6]		206

LcD_16bp[7]		204
LcD_16bp[8]		203
LcD_16bp[9]		200
LcDsAck_2op_[0]	Ext DsAck	183
LcDsAck_2op_[1]		184
LcRnW_ip	RnW	178
LcSize_2ip[0]	Size (byte / word)	180
LcSize_2ip[1]		182
Xc_Bus_4ip[0]	Exchange input bus for cross-over switch protection	43
Xc_Bus_4ip[1]		44
Xc_Bus_4ip[2]		46
Xc_Bus_4ip[3]		47
Xc_Bus_4op[0]	Exchange output bus for cross-over switch protection	48
Xc_Bus_4op[1]		49
Xc_Bus_4op[2]		50
Xc_Bus_4op[3]		51
NCO_6op[0]	NCO to resistor ladder to generate clk 1.5 MHz	55
NCO_6op[1]		57
NCO_6op[2]		58
NCO_6op[3]		59
NCO_6op[4]		61
NCO_6op[5]		62
NcoClk_ip	Clk1.5 for LIU	54
Ram_Cs_op_	CS to ext RAM	163
Reset_ip_	Global Reset	196
Reset_op_	Reset output detect from CC USAM	174
RxA_Tdm_9ip[0]	Add bus from async mapper	70
RxA_Tdm_9ip[1]		71
RxA_Tdm_9ip[2]		72
RxA_Tdm_9ip[3]		73
RxA_Tdm_9ip[4]		74
RxA_Tdm_9ip[5]		75
RxA_Tdm_9ip[6]		76
RxA_Tdm_9ip[7]		77
RxA_Tdm_9ip[8]		79
RxS_Tdm_9ip[0]	TDM bus from sync framer	95
RxS_Tdm_9ip[1]		96
RxS_Tdm_9ip[2]		98
RxS_Tdm_9ip[3]		99
RxS_Tdm_9ip[4]		100
RxS_Tdm_9ip[5]		101
RxS_Tdm_9ip[6]		102
RxS_Tdm_9ip[7]		103
RxS_Tdm_9ip[8]		104
ScA_bp	Scomm A side	40
ScB_bp		41
TClkRef_op	Clk1.5 to DFIA	118
Tx_SFrSync_op	Frame sync 1 ms to DFIA	117
Tx_Tdm_9op[0]	TDM output bus to DFIA	80
Tx_Tdm_9op[1]		81
Tx_Tdm_9op[2]		82
Tx_Tdm_9op[3]		83
Tx_Tdm_9op[4]		84
Tx_Tdm_9op[5]		85
Tx_Tdm_9op[6]		88
Tx_Tdm_9op[7]		91
Tx_Tdm_9op[8]		94
UsBusReq_op	Us Bus req to CC USAM	39
UsData_4op[0]	Us Bus data to CC USAM	35
UsData_4op[1]		36
UsData_4op[2]		37
UsData_4op[3]		38
UsFsA_ip	Composite syncfrom CC USAM	23
UsFsB_ip		22


ALTERA
APEX
TRANSWITCH
TXC-04228AIBG
Lattice