Engine Interface Card (EIC) for the RIP Development System, version 2.0 Design Guide

July 2000

Literature Number SPIU014

Texas Instruments

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any semiconductor product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance of customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endoresement thereof.

Figures

Tables

Engine Interface Card Design Guide

This document describes the Engine Interface Card (EIC) for the Texas Instruments RIP™ (Raster Image Processor) Development System, version 2.0, and its interfaces, which include the register read/write mechanism and the video interface.

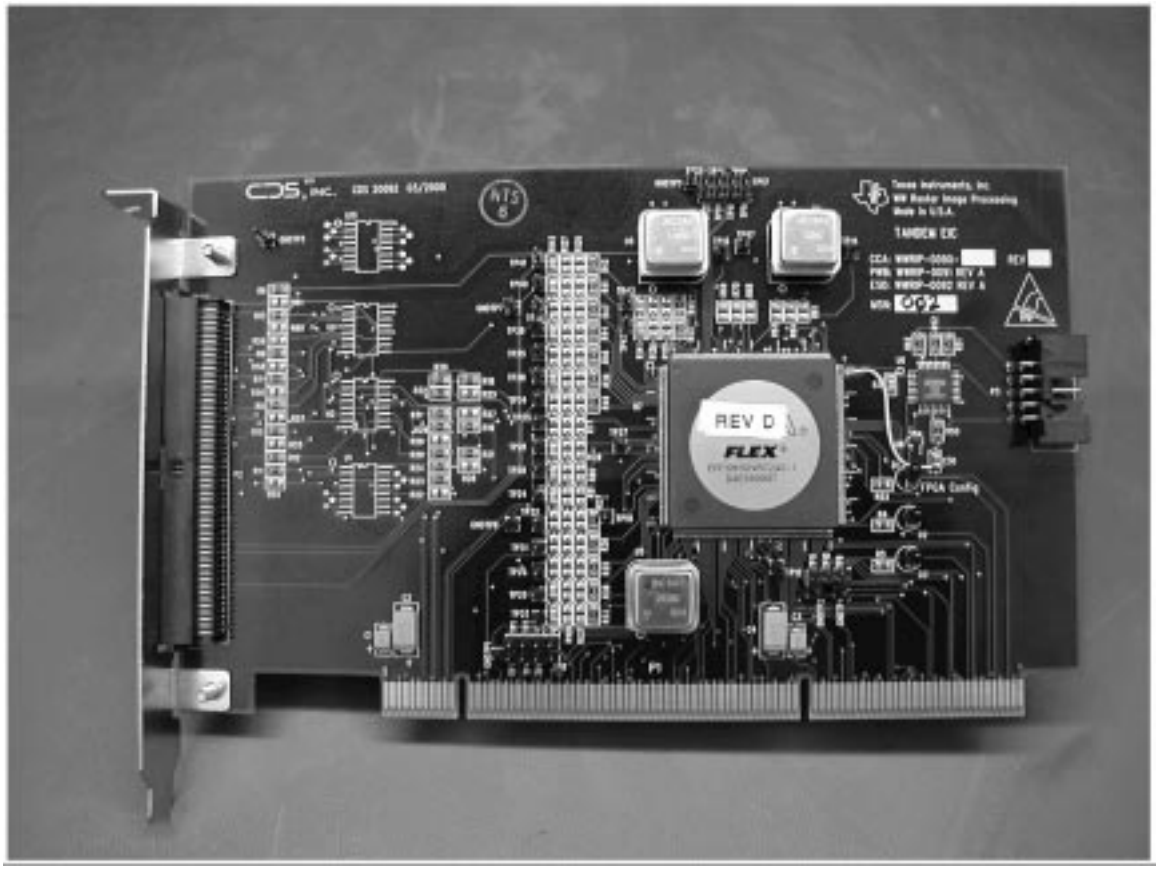
This document is a hardware design guide for customers who desire to modify the Texas Instruments engine interface card or design their own EIC.

1 Overview

The EIC is a FPGA-based add-in card that allows the RIP 2.0 Development System to communicate with an engine, Operator Control Panel (OCP), and video interface. The EIC also has additional resources inside the FPGA and general-purpose I/Os to the engine connector to accommodate additional interfaces to the engine that may be necessary. Therefore, when the RIP Development System version 2.0 is lashed up to different print engines, the EIC is the only thing that must be modified. The RIP 2.0 Development System motherboard can remain the same.

Figure 1 illustrates the RIP 2.0 Development System engine interface card.

Figure 1. RIP 2.0 Development System Engine Interface Card



1.1 Audience

This manual is intended for hardware engineers redesigning firmware in the FPGA on the EIC and for hardware engineers designing a new EIC.

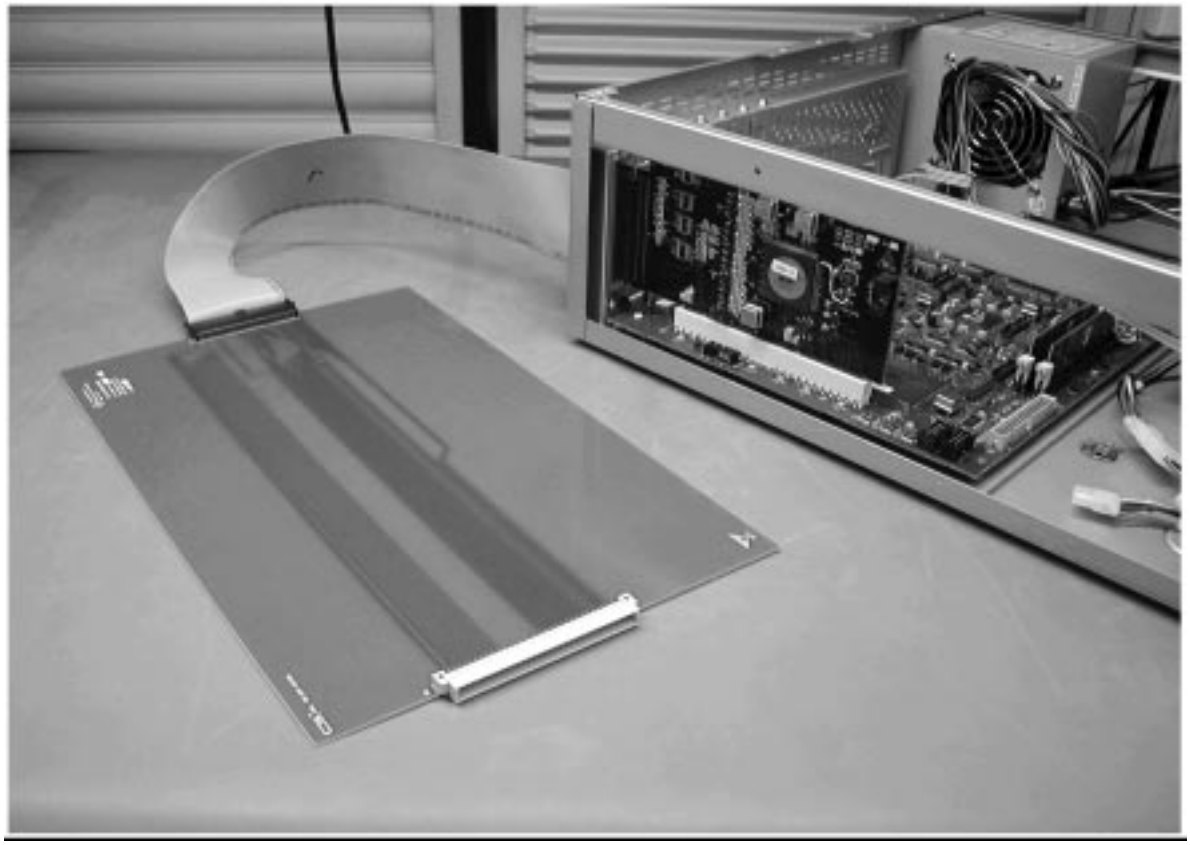
1.2 Features of the Engine Interface Card

The EIC has the following features:

- ❏ Lash-up with both circular or tandem engine
- ❏ Support for on-board/off-board engine communication clock, pixel clock, and OCP/TBD clock
- ❏ Support for differential inputs/outputs for single-bit video, all Vsync signals, and Fsync/Hsync signals
- ❏ Pullup and pulldown pads on all video data and clock signals
- ❏ Pullup and pulldown pads and series termination option on all general purpose I/Os and engine I/Os
- ❏ Interface capability to both the RIP Development System version 1.0 and version 2.0
- ❏ Single field-programmable gate array (FPGA) design for additional flexibility
- ❏ Light-emitting diodes (LEDs) on status signals
- ❏ ID lines to LEDs on the RIP Development System version 2.0 motherboard that can be used to recognize different engine interface cards.

Figure 2 illustrates the RIP 2.0 Development System setup with extender card and cable.

Figure 2. RIP 2.0 Development System Setup



1.3 Related Documentation From Texas Instruments

TMS320C6000 Peripherals Reference Guide (SPRU190C)

2 Architecture

The EIC allows the TIger 2.0 Development System motherboard to communicate with an engine, OCP, and video interface. The card is FPGA based, which allows it to be reprogrammed for a specific printer. The engine, OCP, and video interfaces are generic interfaces that can be easily adapted for print engines from a variety of original equipment manufacturers (OEMs).

The Engine Interface Card can be adapted to various printer engines. Flexibility comes from a single FPGA on board for all programmable hardware, the option of on-board clock oscillators or clock signals from the printer engine, the option of adding differential drivers on four signals going to the engine, and differential receivers on five signals coming from the engine. In addition to these features, the Engine Interface Card also has a footprint for one additional differential driver or receiver. The differential drivers and receivers designed in are low voltage differential swing (LVDS) differential drivers (DS90C031TM from National Semiconductor) and receivers (DS90C032TM from National Semiconductor).

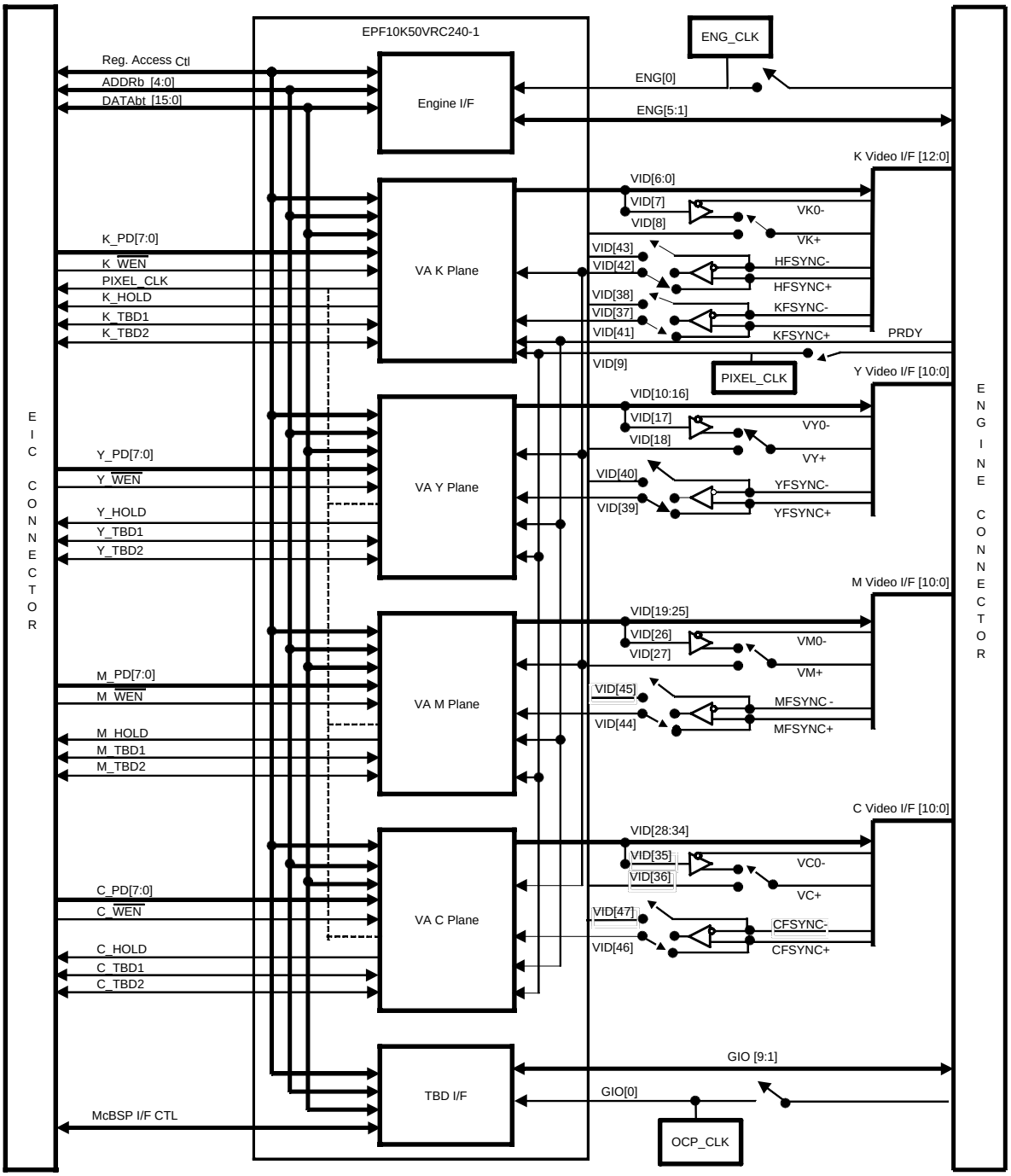
The engine connector signals are in three groups: video, engine, and general purpose I/Os. However, these signals can be reassigned to another group if a redesign makes it necessary. The single FPGA solution provides the designer with this flexibility.

Figure 3 shows the architecture for the tandem engine interface card.

The following notes apply to Figure 3.

1) The switches are implemented as 0-ohm resistors.
2) Split termination is designed in on all video data signals and all clock signals. Pullups and pulldowns are placed on all general-purpose I/Os.
3) Any of the GIO signals can be used for engine communication if needed.
4) Typical FPGA resource usage is 50%.
5) I/Os of FPGA shown are 5-V tolerant. The device itself is a 3.3-V device.
6) Drive strength of each pin of the FPGA shown is 25 mA.

Figure 3. Architecture for Tandem Engine Interface Card



2.1 Engine Interface Card Motherboard Connector

The Engine Interface Card (EIC) interfaces to the TIger 2.0 Development System motherboard via a standard 64-bit, 3.3-V PCI edge connector (PCI Specification 2.1). The power and ground pinout for the connector remains the same as in the PCI specification; however, the remaining pins are used only for communication between the RIP Development System motherboard and the Engine Interface Card.

Note:

The connector **DOES NOT** adhere to the PCI signal conventions and is **ONLY** intended for use in the RIP Development System.

Table 1 shows the pinout for the connector.

Table 1. Engine Interface Motherboard Connector Pinout

Pin	Side B	Side A	Pin	Side B	Side A	Pin	Side B	Side A
1	-12 V	ENGIF_OEz	25	+3.3 V	K_PD[0]	49	GND	DATAbt[9]
2	OCP_OEz†	+12 V	26	REG_Wz		50	GND	GND
3	GND	ENGIF_OEz	27	ID[5]	+3.3 V	51	GND	GND
4	PIXEL_CLK	REG_VA_CSz	28	GND	ADDRb[4]‡	52	DATAbt[8]	ADDRb[0]
5	+5 V	+5 V	29	ID[4]	ADDRb[3]‡	53	DATAbt[7]	+3.3 V
6	+5 V	VA_INT†	30	ID[3]	GND	54	+3.3 V	DATAbt[6]
7	OCP_INT	ENG_INT	31	+3.3 V	ID[2]	55	DATAbt[5]	DATAbt[4]
8	DR‡	+5 V	32	ID[1]	ID[0]	56	DATAbt[3]	GND
9	CLKR‡	FSR‡	33	ADDRb[2]	+3.3 V	57	GND	DATAbt[2]
10	DX‡	+3.3 V	34	GND		58	DATAbt[1]	DATAbt[0]
11	CLKX‡	FSX‡	35	K_WENz	GND	59	+3.3 V	+3.3 V
12	KEY	KEY	36	+3.3 V	PIA_OEz†	60		
13	KEY	KEY	37	K_HOLD	GND	61	+5 V	+5 V
14			38	GND	STATUS†	62	+5 V	+5 V
15	GND	RESETz	39		+3.3 V		64-bit keyway	
16	REG_DSz	+3.3 V	40		REG_OCP_CSz		64-bit keyway	
17	GND		41	+3.3 V	REG_ENG_CSz	63	Y_PD[7]‡	GND
18	K_TBD1‡	GND	42		GND	64	GND	Y_PD[6]‡
19	+3.3 V	K_TBD2‡	43	+3.3 V	$\overline{\text{BVI\_OE}}$†	65	Y_PD[5]‡	Y_PD[4]‡
20	K_PD[7]	K_PD[6]	44	ADDRb[1]	DATAbt[15]	66	Y_PD[3]‡	+3.3 V
21	K_PD[5]	+3.3 V	45	DATAbt[14]	+3.3 V	67	GND	Y_PD[2]‡
22	GND	K_PD[4]	46	GND	DATAbt[13]	68	Y_PD[1]‡	Y_PD[0]‡
23	K_PD[3]	K_PD[2]	47	DATAbt[12]	DATAbt[11]	69	Y_TBD1‡	GND
24	K_PD[1]	GND	48	DATAbt[10]	GND	70	+3.3 V	Y_WENz‡

†The signals OCP_OEz, VA_INT, PIA_OEz, $\overline{\text{BVI\_OE}}$, and STATUS were used on the RIP (TIger) Reference System 1.0 EIC designs but are not required for the TIger Development System 2.0 EIC design. They are shown here as placeholders only so that these pins are not used for anything else on the TIger Development System 2.0 EIC.
‡These are the signals required for the TIger Development System 2.0 EIC that did not exist on the RIP (TIger) Reference System, version 1.0 EIC.

Table 1. Engine Interface Motherboard Connector Pinout (Continued)

Pin	Side B	Side A	Pin	Side B	Side A	Pin	Side B	Side A
71	Y_HOLD‡	Y_TBD2‡	79	+3.3 V	M_WENz‡	87	C_PD[1]‡	GND
72		GND	80	M_HOLD‡	M_TBD1‡	88	+3.3 V	C_PD[0]‡
73	GND	M_PD[7]‡	81	M_TBD2‡	GND	89	C_TBD1‡	
74	M_PD[6]‡	M_PD[5]‡	82	GND		90	C_WENz‡	GND
75	M_PD[4]‡	+3.3 V	83	C_PD[7]‡	C_PD[6]‡	91	GND	C_HOLD‡
76	GND	M_PD[3]‡	84	C_PD[5]‡	+3.3 V	92	C_TBD2‡	
77	M_PD[2]‡	M_PD[1]‡	85	GND	C_PD[4]‡	93		GND
78	M_PD[0]‡	GND	86	C_PD[3]‡	C_PD[2]‡	94	GND	

†The signals OCP_OEz, VA_INT, PIA_OEz, BVI_OEz, and STATUS were used on the RIP (TIger) Reference System 1.0 EIC designs but are not required for the TIger Development System 2.0 EIC design. They are shown here as placeholders only so that these pins are not used for anything else on the TIger Development System 2.0 EIC.
‡These are the signals required for the TIger Development System 2.0 EIC that did not exist on the RIP (TIger) Reference System version 1.0 EIC.

See Section 3, *Electrical Interface*, for signal descriptions.

2.2 Engine Interface Card Printer Connector

The Engine Interface Card interfaces to the printer via a 100-pin male DIN connector (AMP, part number: 1-104069-7). Table 2 describes the Engine Interface Card printer connector pinout.

The EIC typically connects to the print engine through an extender or adapter board which also has a 100-pin male DIN connector and an engine-specific connector. A 100 conductor ribbon cable with a 100-pin female connector at each end is used to connect the EIC and extender board. The cross routing is accomplished in the extender board. The ribbon cable length and the length of signals on the extender board should be kept to a minimum to avoid reflections and signal degradation.

The interfaces to the engine, OCP, and video described in this document are generic. These signals should be mapped to specific signal names for a specific printer. See the engine-specific documents to map actual signal names to the generic names in this document.

Figure 4 shows the RIP 2.0 Development System extender board.

Figure 4. RIP 2.0 Extender Board

Table 2. Engine Interface Card Printer Connector Pinout

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	ENG[0]	26	GND	51	VID[40]	76	VID[24]
2	GND	27	GIO[7]	52	GND	77	VID[25]
3	ENG[1]	28	GND	53	VID[41]	78	GND
4	GND	29	GIO[8]	54	GND	79	VID[26]
5	ENG[2]	30	GND	55	VID[42]	80	GND
6	GND	31	GIO[9]	56	GND	81	VID[27]
7	ENG[3]	32	GND	57	VID[43]	82	GND
8	GND	33	VID[0]	58	GND	83	VID[44]
9	ENG[4]	34	VID[1]	59	VID[9]	84	GND
10	GND	35	VID[2]	60	VID[10]	85	VID[45]
11	ENG[5]	36	VID[3]	61	VID[11]	86	GND
12	GND	37	VID[4]	62	VID[12]	87	VID[46]
13	GIO[0]	38	VID[5]	63	VID[13]	88	GND
14	GND	39	VID[6]	64	VID[14]	89	VID[47]
15	GIO[1]	40	VID[7]	65	VID[15]	90	GND
16	GND	41	GND	66	VID[16]	91	VID[28]
17	GIO[2]	42	GND	67	GND	92	GND
18	GND	43	VID[8]	68	GND	93	VID[29]
19	GIO[3]	44	GND	69	VID[17]	94	VID[30]
20	GND	45	VID[37]	70	VID[18]	95	VID[31]
21	GIO[4]	46	GND	71	VID[19]	96	VID[32]
22	GND	47	VID[38]	72	VID[20]	97	VID[33]
23	GIO[5]	48	GND	73	VID[21]	98	VID[34]
24	GND	49	VID[39]	74	VID[22]	99	VID[35]
25	GIO[6]	50	GND	75	VID[23]	100	VID[36]

3 Electrical Interface

This section describes the electrical interface signals.

3.1 Asynchronous Register Read/Write Access by the TMS320C6x DSP

The TMS320C6x™ digital signal processor (DSP) communicates with all individual design modules using registers in each design and interrupts for status changes. The engine interface is allocated three address spaces of 64 bytes each in the CE3 space of the TMS320C6x™ DSP. The address spaces are identified by three chip selects to the Engine Interface Card: REG_ENG_CSz, REG_VA_CSz, and REG_OCP_CSz.

The registers are accessed with an asynchronous interface that uses data strobes, read/write signal, address, and chip selects.

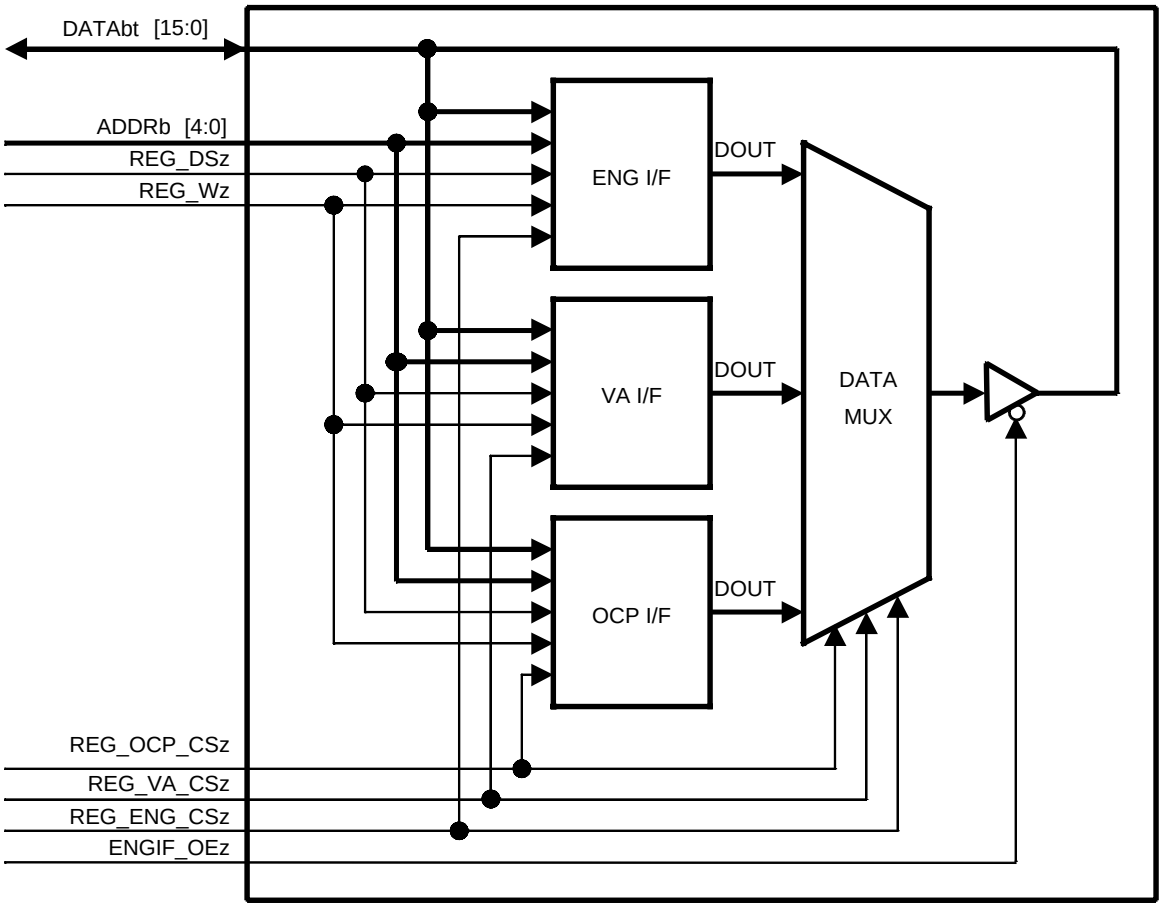
Table 3 lists the interface signals for register accesses performed by the DSP interface. These signals are routed via the motherboard connector.

Table 3. Interface Signals for Register Access by the Processor

Signal Name	Type	Description
RESETz	I	Active low system reset
REG_VA_CSz	I	Selects video adapter for register read/writes
REG_ENG_CSz	I	Selects engine communication for register read/writes
REG_OCP_CSz	I	Selects Operator Control Panel for register read/writes
REG_Wz	I	Active low register write enable
REG_DSz	I	Write data strobe. Indicates that data on data bus is valid and can be loaded into registers during a write. Data is loaded into selected register on falling edge.
ENGIF_OEz	I	Active low output enable. Enables selected module to drive DATAbt[15:0] bus. Each module must 3-state the DATAbt[15:0] bus when its chip select is high. Only the selected module drives the DATAbt[15:0] bus when this signal is active.
ENG_INT	O	Engine interrupt request for the DSP
OCP_INT	O	OCP interrupt request for the DSP
ADDRb[4:0]	I	Address bus for register access. Each address is for a 16-bit register. These address lines, therefore, tie to address lines [6:2] on the ‘C6x DSP address bus.
DATAbt[15:0]	I/O	3-state data bus
ID[5:0]	O	These signals on the TIger 2.0 Development System motherboard drive LEDs. These signals can be used to identify different print engines. These can also be used as general purpose outputs.

Figure 5 shows the logical interface diagram for register read/write.

Figure 5. Logical Interface Diagram for Register Read/Write



3.1.1 Address Map

The engine interface is allocated three address spaces of 64 bytes each in the CE3 space of the ‘C6x DSP. The address spaces are identified by three chip selects to the Engine Interface Card: REG_ENG_CSz, REG_VA_CSz, and REG_OCP_CSz. Each register access in these address spaces is 16 bits wide. The registers are not byte-addressable. Table 4 shows the typical address map inside each of these address spaces. The address offset is from the base address of the address space.

Table 4. Address Map for any Address Space inside EIC FPGA

Registers	Address Offset
Register 1	0x00
Register 2	0x04
Register 3	0x08
.	.
.	.
.	.
Register 31	0x78
Register 32	0x7C

3.1.2 Timing Diagrams for Register Read/Write

The register accesses are asynchronous. Chip selects select the different address spaces in the EIC FPGA. The address selects specific registers in the selected address space in the EIC FPGA. Data strobe is used by the EIC FPGA for register writes and is used as a guideline for timing for register reads. The output enable signal is for the EIC FPGA and is used by all address spaces in the EIC FPGA to enable data output on the data bus. The output enable signal is provided to aid in timing when the data bus is entering a 3-state condition. Table 5 shows the timing parameters specific to register read and write.

Note:
The timings in Table 5 assume 100-MHz clock for the ‘C6x EMIF.

Table 5. Timing Characteristics for Register Read/Write

Parameter	Description	Min	Max	Unit
t_d(ADDRV-CSzV)	Address valid to chip select valid	60	70	ns
t_d(ADDRV-OEzV)	Address valid to output enable valid	60	70	ns
t_d(CSzV-DSzV)	Chip select valid to data strobe valid (This parameter is determined by the setup time T_{setup} of the ‘C6x DSP.)	T_{setup} – 70 ns	T_{setup} –60 ns	ns
t_d(DSzV – Dlatch)	Data strobe valid to data latch in register (determined by FPGA design, and timings met by FPGA)	TBD	TBD	ns
t_d(DSzV-DSPRead)	Data Strobe valid to Data read by DSP (This parameter is determined by strobe time T_{strobe} of the ‘C6x DSP.)	-	T_{strobe} – 20 ns	ns

Figure 6 shows the timing for register write by processor. The data on the data bus is latched into the module register on the falling edge of the REG_DSlz signal, if the chip select for the module is active and the REG_WI is active.

Figure 6. Register Write Timing Diagram

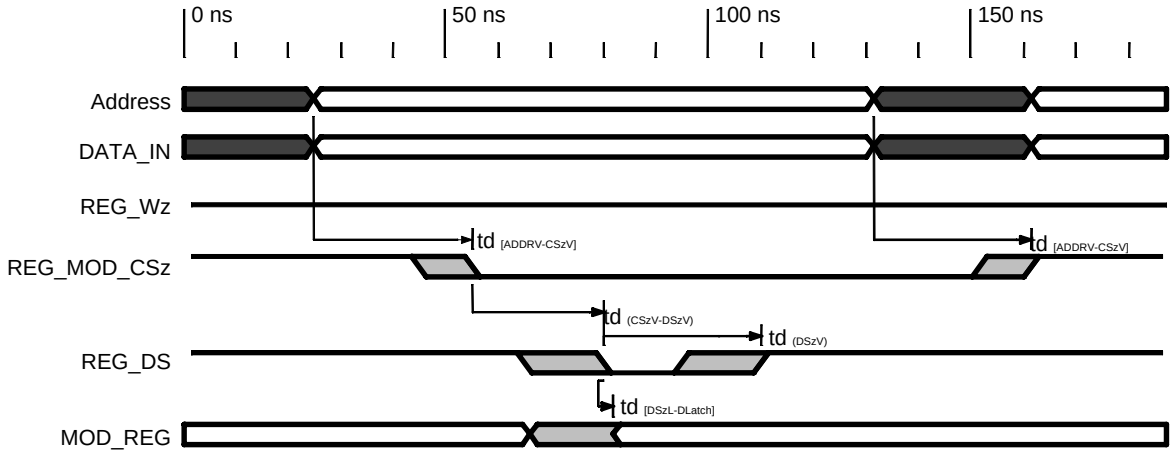


Figure 7 shows timing for register read by processor. The design module outputs data on the data bus when the chip select is valid and the output enable is active. Otherwise, the data bus should be 3-state. The TMS320C6x™ DSP latches in data from the data bus after a period of t_d(DSzV) after the falling edge of REG_DSz. Table 6 illustrates the timing requirements for register read/write operations.

Figure 7. Register Read Timing Diagram

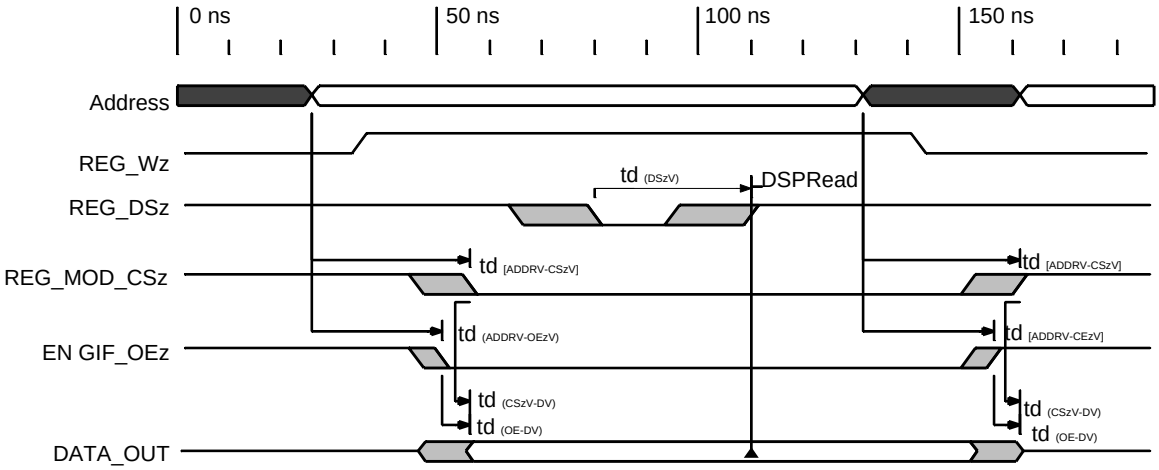


Table 6. Timing Requirements for Register Read/Write

Signal Name	Type † ‡ §	Timing Requirements	
		Min (ns)	Max (ns)
Address	T_{su}/T_{pd}	-	60 ns
REG_Wz	T_{su}/T_{pd}	-	40 ns
REG_DSz	T_{su}/T_{pd}	-	30 ns
REG_MOD_CSz	T_{su}/T_{pd}	-	16 ns
ENGIF_OEz	T_{su}/T_{pd}	-	16 ns
$t_d(\overline{DS}V\text{-DSPRead})$	T_{co}/T_{pd}	-	10 ns ¶

†T_{su} = Setup time
‡T_{pd} = Propagation delay
§T_{co} = Clock to output delay
¶This applies only if data is output on data bus after REG_DSz becomes valid.
†

3.1.3 Sample Source Code for Register Read/Write in VHDL

```
-- Bidirectional Bus separated out into Data In and Data Out bus
DATAINb <= DATAbt;
DATAbt <= DATAOUTb when ENGIF_OEz = ‘0’ and REG_Wlz = ‘1’ and REG_VA_CSz = ‘0’ else
   (others => ‘Z’);
-- purpose: This process writes to Video adapter registers
REG_WRITE: process(REG_DSlz, RESETz)
begin
      if RESETz = '0' then
            VA_Reg1 <= ( others => '0');
            VA_Reg2 <= ( others => '0');
      elsif REG_DSlz'event and REG_DSlz = '0' then
            if REG_VA_CSlz = '0' and REG_Wlz = '0' then
                  case ADDRb(4 downto 0) is
                        when "00000" =>
                              VA_Reg1(15 downto 0) <= DATAINb(15 downto 0);
                        when "00001" =>
                              VA_Reg2(15 downto 0) <= DATAINb(15 downto 0);
                        when others =>
                              null;
                  end case;
             end if;
      end if;
end process REG_WRITE;

-- purpose: This process reads Video Adapter registers
REG_READ: process(RESETz, ADDRb)
begin
      if RESETz = '0' then
            DATAOUTb <= (others => '0');
      else
            case ADDRb(6 downto 2) is
                  when "00000" =>
                        DATAOUTb <= VA_Reg1;
                  when "00001" =>
                        DATAOUTb <= VA_Reg2;
                  when others =>
                        DATAOUTb <= (others => '0');
            end case;
      end if;
end process REG_READ;
```

3.2 Serial Register Access/Serial Communication Using the ‘C6x McBSP Port

The multichannel buffered serial port (McBSP) on the TMS320C6x™ DSP is based on the standard serial port interface. Features of the McBSP include:

- ❏ Full-duplex communication
- ❏ Double-buffered data registers, which allow a continuous data stream
- ❏ Independent framing and clocking for receive and transmit

- ❏ Direct interface to:
  - ■ T1/E1 framers
  - ■ MVIP switching compatible and ST-BUS compliant devices including MVIP framers, H.100 framers, and SCSA framers
  - ■ IOM-2 compliant devices
  - ■ AC97 compliant devices
  - ■ IIS compliant devices
  - ■ SPI devices
- ❏ Data sizes of 8, 12, 16, 20, 24, and 32 bits
- ❏ 8-bit data transfers with the option of LSB or MSB first
- ❏ Programmable internal clock and frame generation
- ❏ Programmable polarity for both frame synchronization and data clocks

For more information on this interface to the ‘C6x DSP, see the *TMS320C6000 Peripherals Reference Guide* (SPRU190C).

Table 7 lists the interface signals for register accesses performed by the DSP through the McBSP. These signals are routed via the motherboard connector.

Table 7. Interface Signals for Register Access by the TMS320C6x™ McBSP

Signal Name	Type	Description
CLKR	I/O/Z	McBSP receive clock
CLKX	I/O/Z	McBSP transmit clock
DR	I	McBSP received serial data
DX	O/Z	McBSP transmitted serial data
FSR	I/O/Z	McBSP receive frame synchronization
FSX	I/O/Z	McBSP transmit frame synchronization

3.3 Interface to Video Interface

Following are the key features of the video interface (VI) on the motherboard:

- ❏ Supports transfer of pixel data stream to the engine simultaneously for black, cyan, magenta, and yellow (KCMY) color planes

- ❏ Can be programmed to insert blank pixels at the beginning and end of each line in the active image plane
- ❏ Can be programmed to insert blank bands anywhere in image plane
- ❏ Supports sending data starting from top-left corner of image plane to bottom-right corner and in reverse direction
- ❏ Can be programmed to skip a few pixels at the beginning of each band

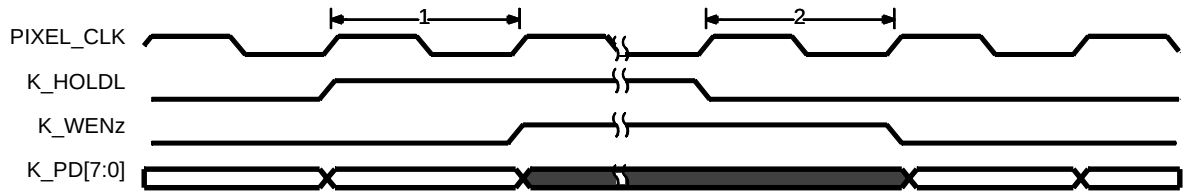
Table 8 lists the video interface signals between the TIger 2.0 Development System motherboard and the Engine Interface Card. These signals are routed via the motherboard connector.

Table 8. Video Interface Signals

Signal Name	Type	Description
PIXEL_CLK	O	Pixel clock output
K/Y/M/C_HOLD	O	Request for not to send pixels for video interface (IF) – K/Y/M/C
K/Y/M/C_WENz	I	Pixel data strobe for video IF – K/Y/M/C
K/Y/M/C_PD[7:0]	I	Pixel data input for video IF – K/Y/M/C
K/Y/M/C_TBD1,2	I/O	TBD signals for video IF – K/Y/M/C
PIXEL_CLK	O	Pixel clock output

The video adapter controls the flow of image pixels from the video interface (VI) to the printer engine. The video interface always tries to send pixels out whenever it has pixels for transfer. The video adapter (VA) controls the flow of pixels using the handshake signals provided by the engine, for example, line synchronization (Line Sync) and plane synchronization (Plane Sync). Based on handshake signals from the print engine, the video adapter will halt or continue transfer of pixels from the video interface. Figure 8 shows the handshake signals between the video adapter and the video interface.

Figure 8. Timings for Pixel Transfer From the VI to the VA



The video adapter indicates to the video interface to stop sending pixel data by asserting the HOLD signal. The video interface stops transferring pixels at the next clock after receiving HOLD from the VA (shown as 1). The video interface starts sending pixels at one clock after HOLD is de-asserted (shown as 2). The video adapter should read in pixel data from the input pixel bus only when the video interface asserts WENz, which indicates valid data on bus.

3.3.1 Timing Requirements for Interface to Video Interface

The timing requirements for the interface to the video interface are specified in terms of period of the PIXEL_CLK, as shown in Table 9. As the PIXEL_CLK requirements change, the timing requirements change as well.

Table 9. Timing Requirements for Interface to the Video Interface

Signal Name	Type	Timing Requirements	
		Min (ns)	**Max (ns)**
[K,C,M,Y]_HOLD	T_{co}	-	T_{pclk} – 20 ns
[K,C,M,Y]_WENz	T_{su}	-	T_{pclk} – 20 ns
[K,C,M,Y]_PD[7:0]	T_{su}	-	T_{pclk} – 20 ns
[K,C,M,Y]_TBD[1:0]	N/A	-	-

Notes: 1) T_{pclk} = clock period of PIXEL_CLK.
2) T_{co} = clock to output time
3) T_{su} = setup time

3.4 Interface to Printer Engine

Table 10 lists the interface signals to the printer engine, OCP, and video interface. These signals are routed via the EIC printer connector. These signals are mapped to the appropriate signals for each print engine. All these signals come from a single FPGA and go to the printer connector. Therefore, re-mapping of these signals is possible.

Table 10. Printer Engine, OCP, and Video Interface Signals

Signal Name	Type	Description
OCP_CLK	I	External clock input. If this clock is needed, the OCP being interfaced specifies its frequency.
ENG_CLK	I	External clock input for engine/status command interface. If this clock is needed, the engine being interfaced specifies its frequency.
VID_CLK	I	External clock input for video transmission to the engine. If this clock is needed, the engine being interfaced specifies its frequency.
GIO[9:0]	I/O	Generic signals defined by the OCP/other engine interfaces
ENG[5:0]	I/O	Generic signals defined by the engine being interfaced
VID[47:0]	I/O	Generic signals defined by the engine being interfaced

3.5 Pin Assignments for Tandem EIC FPGA

Table 11 lists the pin assignments for the tandem EIC FPGA.

Table 11. Pin Assignments for Tandem EIC FPGA

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
6	GIO[9]	49	ENG[3]	90	REG_DSz	132	ADDRb[2]
7	GIO[8]	50	ENG[2]	91	ENG_CLKp	133	ADDRb[1]
8	GIO[7]	51	ENG[1]	92	RESETz	134	ADDRb[0]
9	GIO[6]	53	N/C	94	Y_PD[6]	136	REG_OCP_CSz
11	GIO[5]	54	N/C	95	Y_PD[5]	137	OCP_INT
12	GIO[4]	55	N/C	97	Y_PD[4]	138	ENG_INT
13	GIO[3]	56	N/C	98	Y_PD[3]	139	N/C
14	GIO[2]	61	K_PD[7]	99	Y_PD[2]	141	K_TBD2
15	GIO[1]	62	K_PD[6]	100	Y_PD[1]	142	K_TBD1
17	LED[0]	63	K_PD[5]	101	Y_PD[0]	143	K_HOLD
18	LED[1]	64	K_PD[4]	102	M_PD[7]	144	K_WENz
19	N/C	65	K_PD[3]	103	M_PD[6]	146	Y_HOLD
20	N/C	66	K_PD[2]	105	M_PD[5]	147	Y_WENz
21	N/C	67	K_PD[1]	106	M_PD[4]	148	Y_TBD2
23	VID[47]	68	K_PD[0]	107	M_PD[3]	149	Y_TBD1
24	VID[46]	70	DATAbt[15]	108	M_PD[2]	151	M_HOLD
25	VID[43]	71	DATAbt[14]	109	M_PD[1]	152	M_WENz
26	VID[45]	72	DATAbt[13]	110	M_PD[0]	153	M_TBD2
28	VID[44]	73	DATAbt[12]	111	C_PD[7]	154	M_TBD1
29	VID[42]	74	DATAbt[11]	113	C_PD[6]	156	C_HOLD
30	VID[41]	75	DATAbt[10]	114	C_PD[5]	157	C_WENz
31	VID[40]	76	DATAbt[9]	115	C_PD[4]	158	C_TBD2
33	VID[39]	78	DATAbt[8]	116	C_PD[3]	159	C_TBD1
34	N/C	79	DATAbt[7]	117	C_PD[2]	161	DX
35	N/C	80	DATAbt[6]	118	C_PD[1]	162	CLKX
36	VID[38]	81	DATAbt[5]	119	C_PD[0]	163	FSX
38	VID[37]	82	DATAbt[4]	120	N/C	164	CLKR
43	N/C	83	DATAbt[3]	126	REG_VA_CSz	166	FSR
44	N/C	84	DATAbt[2]	127	REG_Wz	167	DR
45	N/C	86	DATAbt[1]	128	REG_ENG_CSz	168	ID[5]
46	ENG[5]	87	DATAbt[0]	129	ADDRb[4]	169	ID[4]
48	ENG[4]	88	Y_PD[7]	131	ADDRb[3]	171	ID[3]

Table 11. Pin Assignments for Tandem EIC FPGA (Continued)

Pin	Signal	Pin	Signal
172	ID[2]	212	ENGIF_OEz
173	ID[1]	213	VID[36]
174	ID[0]	214	VID[27]
175	N/C	215	VID[18]
181	N/C	217	N/C
182	VID[35]	218	N/C
183	VID[34]	219	N/C
184	VID[33]	220	N/C
185	VID[32]	221	N/C
186	VID[31]	222	N/C
187	VID[30]	223	N/C
188	VID[29]	225	N/C
190	VID[28]	226	N/C
191	VID[26]	227	N/C
192	VID[25]	228	N/C
193	VID[24]	229	N/C
194	VID[23]	230	N/C
195	VID[22]	231	VID[8]
196	VID[21]	233	VID[7]
198	VID[20]	234	VID[6]
199	VID[19]	235	VID[5]
200	VID[17]	236	VID[4]
201	VID[16]	237	VID[3]
202	N/C	238	VID[2]
203	VID[15]	239	VID[1]
204	VID[14]	240	VID[0]
206	VID[13]		
207	VID[12]		
208	VID[11]		
209	VID[10]		
210	GIO[0]		
211	PXL_CLKp		

4 Mechanical

The Engine Interface Card follows the PCI mechanical specifications for a 3.3-volt, 64-bit short card. The Engine Interface Card (EIC) interfaces to the RIP Development System 2.0 motherboard via a standard 64-bit, 3.3-V PCI edge connector (PCI Specification 2.1). The power and ground pinout for the connector remains the same as in the PCI specification. However, the rest of the pins are used for communication between the RIP Development System 2.0 motherboard and the Engine Interface Card. Therefore, it **DOES NOT** adhere to the PCI signal conventions and is **ONLY** intended for use in the TIger 2.0 Development System. Table 2 on page 12 shows the pinout for the connector.

Figure 9 shows the physical outline of the card. See the PCI specifications for the complete mechanical requirements.

Figure 9. Engine Interface Card Physical Outline