

# Design and Development of Test Jig for CAIR DSP

Dr. Kesavan Gopal

School of Electronics & Electrical Engineering  
Lovely Professional University, Punjab, India

**Abstract:** In this paper, a new design and development of Test Jig for CAIR DSP is proposed and implemented. The design, development and production of the “TestJig” units are to be realized around CAIR ASIC in association with DSP and FPGA with other peripherals and protocol interfaces. The functional demarcations between DSP and FPGA are left to the discretion of the designer which covers the interface of various peripherals and interfaces protocols. The functionality of the developed “TestJig” shall be demonstrated for the integrity of peripherals and protocol interfaces by developing necessary test programs. Technicalities as above and proposes to participate in the above mentioned design, development, deliver and warranty as per requirement and is submitting this paper.

**Keywords:** CAIR DSP, FPGA, Test Jig, Interface device driver Development.

## 1.0 Introduction

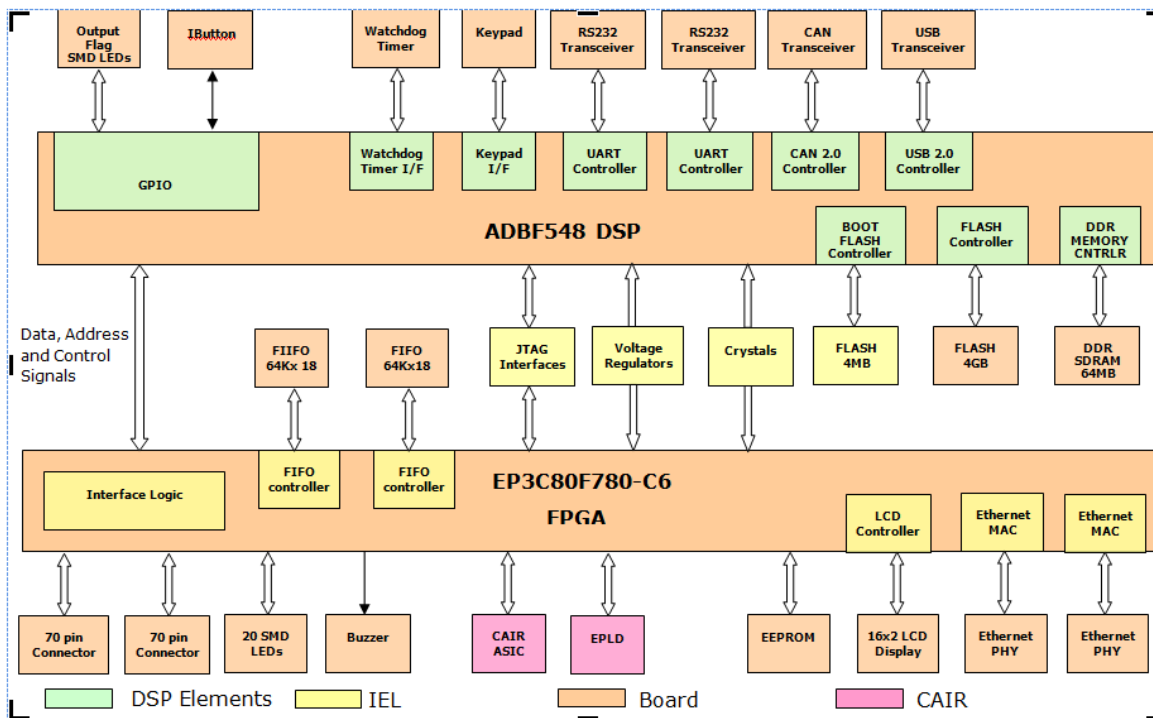
The Hardware design shall comprise of interfacing CAIR ASIC, DSP, FPGA, Memory components and all the peripheral interfaces. The hardware design, development and realization of “TestJig” shall be carried-out around CAIR ASIC, DSP, EPLD and FPGA. The inter-connectivity of all these major devices, associated memory components, memory controller functions, peripheral connectivity through FPGA & DSP shall be carried-out. FPGA based: NIOS soft processor porting and interfacing to modules realized in VHDL, CRC-32, FEC encoder & Decoder, FIFO interface and Ethernet MAC shall be realized in FPGA. The CAIR ASIC shall be interfaced to FPGA and related application can be realized in VHDL or in NIOS and necessary interface signals also can be realized. The functions in EPLD and FPGAs shall be realized on Altera QuartusII along with SOPC builder. The recommended EPLD & FPGA devices shall be used as per the RFP document. DSP based: Boot-FLASH accessing, FLASH memory, SDRAM, Keypad & LCD interface, iButton, CAN interface and USB functionalities shall be realized as DSP based functions. The DSP development shall be carried-out using Visual DSP.

The “TestJig” development consists of Architecture design, component selection, circuit design, VHDL code development, BSP development, PCB design, component procurement, PCB CAD work, Fabrication & populating, testing and proving the functionality of the “TestJig” along with all the peripherals and related interfaces. The following are the hardware elements interfaced around FPGA and DSP to realize “TestJig”. All components shall be selected to meet the electrical & environmental conditions as per the specifications.

. ADBF548 DSP processor, EP3C80F780 Cyclone III FPGA, . EPLD- EPM240, . 4MB Flash Memory, . Flash Memory of 4GB, . SDRAM- 64MB, . 64Kx18 FIFO, . Configuration EEPROM, . 10/100 Ethernet controller, . Front-panel keypad and display, . CAN 2.0 Interface, . USB 2.0 Interface, . High Speed RS232, . On board DC-DC converters for power supply

## Architecture Diagram:

The block diagram of “TestJig” is shown here to depict the connectivity of peripherals and interfaces to DSP, EPLD and FPGA.



### DSP Design Flow

As per the block diagram, the following interfaces are realized on Analog Devices DSP, AD BF548.

1. CAN
2. USB
3. RS-232
4. Keypad
5. Watchdog Timer
6. iButton
7. Display LEDs
8. Boot Flash Controller
9. Additional Flash Controller
10. DDR Memory Controller

### 1.1 Peripherals on DSP ADBF548

#### 1.1.1 CAN

The ADSP-BF548 processor offers up to two CAN controller interfaces that implement the Controller Area Network (CAN) 2.0B (active) protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. We select infineon TLE6250E/TLE6250C high speed CAN transceiver for interfacing to CAN controller port of DSP.

#### 1.1.2 USB

The USB OTG controller present in the AD BF548 DSP processor provides a low-cost connectivity solution for direct interfacing to the USB devices. We select the ST STULP101A/B USB transceiver for interfacing to USB On-The-Go (OTG) feature of USB 2.0 Specification, operates on both host and peripheral modes. This transceiver handles connection detection functionality as well as providing the analog electrical signaling required meeting the USB specification supporting USB 2.0 full speed (12 Mbps) & High Speed (480 Mbps).

### 1.1.3 RS-232

The AD-BF548 DSP processor supports up to four full-duplex universal asynchronous receiver/transmitter (UART) ports. Each UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. A UART port includes support for five to eight data bits, one or two stop bits, and none, even, or odd parity. Each UART port's baud rate, serial data format, error code generation and status, and interrupts are programmable. An RS 232 transceiver of analog devices AD 7306 is interface to this UART port of the DSP processor. The Baud rate of RS 232 interface is programmable and selected through the Key and display menu.

### 1.1.4 Keypad

The keypad interface is a 16 pin interface module that is used to detect the key pressed in a keypad matrix (with left, right, top, bottom, and enter). The interface is capable of filtering the bounce on the input pins, which is common in keypad applications. The width of the filtered bounce is programmable. The Interface module is capable of generating an interrupt request to the core once it identifies that any key has been pressed (using the Keypad IRQ). The interface supports a press-release-press mode and infrastructure for a press-hold mode. Push button arranged in the corners of the square with enter button center of the square. Push button is placed as per the requirement of the CAIR. Key selection is displayed as a menu in the LCD. The menu includes the bit rate selection of the Ethernet and RS 232.

### 1.1.5 Watchdog Timer

The AD BF548 processor includes a 32-bit timer that can be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state through generation of a hardware reset, non-maskable interrupt (NMI), or general-purpose interrupt, if the timer expires before being reset by software. The programmer initializes the count value of the timer, enables the appropriate interrupt, then enables the timer. Thereafter, the software must reload the counter before it counts to zero from the programmed value. This protects the system from remaining in an unknown state where software, which would normally reset the timer, has stopped running due to an external noise condition or software error.

### 1.1.6 iButton

iButton chip is interfaced to the GPIO pins of the DSP AD BF548 via Dallas probe (DS 9092). The communication between the iButton and the processor is through 1-wire protocol operating at overdrive mode at 142kbps. iButton is connected to this probe the unique identification (address) is displayed on the LCD. The iButton content is modified with known pattern and is read to the PC via RS 232 interface for validation. Same way the iButton content written from the PC is read by DSP processor and displayed on the LCD.

### 1.1.7 LED

LEDs are interfaced to the GPIO pins of the DSP processor through suitable drivers for displaying the Interrupt status & flags of the DSP.

### 1.1.8 Boot Flash controller

The AD-BF548 processor has many mechanisms automatically loading internal and external memory after a reset. The boot mode is defined by four BMODE input pins dedicated to this purpose. There are two categories of boot modes: In master boot modes the processor actively loads data from parallel or serial memories. In slave boot modes the processor receives data from external host devices. As per CAIR requirement these four pins are set to either 0001 & 0011 for booting from EEPROM flash.

### 1.1.9 Additional Flash Memory Controller

A 4GB Compact NAND Flash is interfaced to AD-BF548's NAND Flash Controller (NFC) port as part of the external bus interface. These NAND flash device interface provide high-density, low-cost memory. However, NAND flash devices also have long random access times, invalid blocks, and lower reliability over device lifetimes. Because of this, NAND flash is often used for read-only code storage. In this case, all DSP code can be stored in NAND flash and then transferred to a faster memory (such as DDR or SRAM) before execution. Another common use of NAND flash is for storage of multimedia files or other large data segments. Applications like Block level accessing, Deleting etc shall be developed.

### 1.1.10 DDR Memory Controller

Through the External Bus Interface Unit (EBIU) the AD-BF548 processors provide glueless connectivity to external 16-bit wide memories, such as DDR SDRAM, Mobile DDR, SRAM, NOR flash, NAND flash, and FIFO devices. To provide the best performance, the bus system of the DDR interface is completely separate from the other parallel interfaces. The DDR/Mobile DDR memory controller can gluelessly manage up to two banks of double-rate synchronous dynamic memory (DDR1 SDRAM). The 16-bit wide interface operates at SCLK frequency enabling maximum throughput of 532 Mbyte/s. The DDR controller is augmented with a queuing mechanism that performs efficient bursts onto the DDR. The controller is an industry standard DDR SDRAM controller with each bank supporting from 64 Mbit to 512 Mbit device sizes and 4-, 8-, or 16-bit widths. The controller supports up to 512 Mbytes in one bank, but the total in two banks is limited to 512 Mbytes. Each bank is independently programmable and is contiguous with adjacent banks regardless of the sizes of the different banks or their placement. Traditional 16-bit asynchronous memories, such as SRAM, EPROM, and flash devices, can be connected to one of the four 64 MByte asynchronous memory banks, represented by four memory select strobes. Alternatively, these strobes can function as bank-specific read or write strobes preventing further glue logic when connecting to asynchronous FIFO devices.

C code shall be developed and compiled on Visual DSP++ environment for interfacing and developing drivers for DDR.

### 1.1.11 Interface:

All the DSP Address, Data Bus, Control signal and port interrupts are connected to the FPGA for interfacing.

Status of the SIC registers, wait registers and general purpose registers are brought to the GPIO pins in order to be displayed on the LEDs connected to DSP GPIO pins

On power-ON, The DSP boots from the external Flash memory by placing the jumpers on the Boot selection pins of the DSP (for example the pattern 0001 for external flash & 0011 for EEPROM Flash).

## 2.0 FPGA design flow

As per the block diagram, the following interfaces are realized on Altera FPGA EP3C80F780-C6:

1. FIFO controller – 2Nos
2. LCD Controller – 1 No
3. Ethernet MAC- 2 Nos
4. DSP Interface Logic
5. 70 pin connector interface- 2 Nos
6. SMD LED interface - 20 Nos
7. Buzzer interface – 1 No
8. CAIR ASIC interface – 1 No
9. EPLD interface – 1 No

## 2.1 Peripherals on FPGA:

FIFOs are planned to be interfaced to FPGA for the following reasons.

- Dual mode operation of FIFO can be realized
- Handling of both FIFOs at the same time or in ping pong condition is possible.

Ethernet is also planned to be interfaced to FPGA for following reasons:

- Ethernet MAC in HDL is available with IEL, PHY shall be an external element
- If realized on DSP, both MAC and PHY shall be external components.

LCD interface

- Low frequency component
- Used for displaying status or other variables parallel to the working conditions.
- In-Built memory of FPGA can be used for LUT.

## 2.2 Pin Requirement:

The selected FPGA has 413 user I/Os. The below table describes the pin requirement as per the block diagram

SI No	Interface	Pin description	No of pins
1.	FIFO Interface	Input & output Data Bus(36), Control signals(12), Status signals(6),	108
2.	LCD Interface	Data (8), Handshake (3)	11
3.	Ethernet Interface	10/100 Ethernet	12
4.	DSP interface Logic	Data bus (16), Address (16) Interrupts (2)	34
5.	70 pin connector interface	GPIOs for testing & Monitoring	140
6.	SMD LED interface	Status signals	20
7.	Buzzer interface	For a defined condition Audio Alarm S/W programmable	1
8.	CAIR ASIC interface	To be defined (a)	a
9.	EPLD interface	To be defined (b)	b
<b>Total</b>			326+a+b

The CAIR ASIC and EPLD interface signals should not exceed more than 50, leaving about 30 I/Os on FPGA as design ethics.

Apart from the above the FPGA has specified pins for PROM, JTAG, input clocks (Global) and voltage supplies.

## 2.3 Memory Requirement:

The selected FPGA has 2745 Kbits of memory. The memory shall be required for LUT for LCD display and for buffering Ethernet data.

The table below describes the memory requirement:

SI No	Interface	Memory requirement
1.	Ethernet MAC	4K Bytes
2.	LCD LUT	100 Bytes



## 2.4 Crystal requirement:

Separate clock signals may be required for CAIR ASIC, EPLD and FPGA. The reason could be they operate at different speeds and all of them may not get configured at power-on to generate clocks

## 2.5 Design aspects:

The table below shows the device drivers/ application development to be realized in FPGA

SI No:	Description	Status	Schedule in Days	Remarks
1.	FIFO controller: Reading and writing onto both the FIFO	Device driver Available  Application to be developed	2	FIFO device driver is available and application to be developed. Minor modifications may be required to meet CAIR specifications. Currently realized in ping-pong method.
2.	LCD controller: MAC IDs of the PCs connected to Ethernet ports to be displayed	Device driver Available  Application to be developed	3	LCD device driver is available and application to be developed. Minor modifications/ additions in LUT may be required to meet CAIR specifications.
3.	Ethernet MAC: Communication between PC connected to ONE Ethernet port of TestJig and another PC connected through HUB on other port of TestJig	Ethernet MAC Available  Application to be developed	5	Ethernet MAC is available and application to be developed and interface to PHY to be developed. Minor modifications may be required to meet CAIR specifications.
4.	NIOS soft-processor implementation	Expertise available to port & integrate application.	5	Already ported NIOS processor on FPGA for accessing memory devices. CAIR specifications may be addressed specifically.
5.	Implementation of CRC-32	Available/ To be integrated	5	Available in Megafunction wizard of Altera Tool. This has to be integrated onto NIOS processor using SOPC builder and to be demonstrated
6.	Implementation of FEC encoder-decoder	Available/ To be integrated	5	Available in Megafunction wizard of Altera Tool. This has to be integrated onto NIOS processor using SOPC builder and to be demonstrated
7.	20 LEDS to blink at different rates	To be developed	1	Minimum blinking period to be slightly greater than the persistence of vision
8.	Data received from DSP processor to be inverted and send back to DSP	To be developed	1	Whether the data to be send immediately or bunch of data to be received and later to be send back
9.	FPGA shall generate interrupts to DSP processor to toggle an	To be developed	1	The minimum frequency to be decided.

	LED			
10.	Configuration of FPGA. Either from JTAG or PROM or DSP	Available	3	Selection criteria to be decided.
11.	Buzzer interface	To be developed	2	
12.	70 pin connector	To be developed	1	Logic levels and directions if any to be decided
13.	Interface between DSP and FPGA			DSP DataBus, AddrBus & Controls connected.

3.0 In this paper, a design and development of test jig for CAIR ASIC DSP and FPGA's has been proposed and implemented. The design involves the level of upto-32-bit interface, FPGA can support both PCI and ISA bus of width 128-bit with JTAG support and capable of working at 1.6 GHz frequency ranges.

### References:

- [1] S. C. Chang, M. Marek-Sadowska, and T. T. Hwang, "Technology mapping for the fpgas based on decomposition of binary decision diagrams," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 10, pp. 1226–1236, Oct. 1996.
- [2] M. Rawski, L. J'ozwiak, and T. Łuba, "Functional decomposition with an efficient input support selection for sub-functions based on information relationship measures," *Journal of Systems Architecture*, vol. 47, pp. 137–155, 2001.
- [10] C. Scholl, *Functional Decomposition with Application to FPGA Synthesis*. Kluwer: Academic Publishers, 2001.
- [3] T. Łuba, H. Selvaraj, M. Nowicka, and A. Kra'sniewski, "Balanced multilevel decomposition and its applications in fpga-based synthesis," in *Logic and Architecture Synthesis*, G. Saucier and A. Mignotte, Eds., 1995.
- [4] M. Nowicka, T. Łuba, and M. Rawski, "Fpga-based decomposition of boolean functions: Algorithms and implementation," in *Proc. of Sixth International Conference on Advanced Computer Systems*, Szczecin, Poland, 1999, pp. 502–509.
- [5] B. J. Falkowski, "Haar transform: Calculation, generalizations, and applications in logic design and signal processing," in *Proc. of International Workshop on Transforms and Filter Banks (2nd IWTFB)*, Brandenburg, Germany, Mar. 1999, pp. 101–120.
- [6] "Compact representations of logic functions for lossless compression of grey scale images," *IEE Proc., Computers and Digital Techniques*, United Kingdom, vol. 151, no. 3, pp. 221–230, May 2004.
- [7] R. M. Rao and A. S. Bopardikar, *Wavelet Transform: Introduction to Theory and Applications*. Addison-Wesley, 1998.
- [8] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, Oct. 1991.
- [9] C. M. Brislawn, C. B. J. Bradley, R. Onyshczak, and H. T., "The fbi compression standard for digitized fingerprint images," in *Proc. of SPIE Conference 2847*, Denver, USA, 1996, pp. 344–355.

[10] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," Mathematics of Computation, vol. 19, pp. 297–301, 1965.

[11] R. G. Lyons, Understanding Digital Signal Processing. Upper Saddle River: Prentice Hall, 200

