Blue Gene/L compute chip: Control, test, and bring-up infrastructure

R. A. Haring
R. Bellofatto
A. A. Bright
P. G. Crumley
M. B. Dombrowa
S. M. Douskey
M. R. Ellavsky
B. Gopalsamy
D. Hoenicke
T. A. Liebsch
J. A. Marcella
M. Ohmacht

The Blue Gene®/L compute (BLC) and Blue Gene/L link (BLL) chips have extensive facilities for control, bring-up, self-test, debug, and nonintrusive performance monitoring built on a serial interface compliant with IEEE Standard 1149.1. Both the BLL and the BLC chips contain a standard eServer™ chip JTAG controller called the ***access macro****. For BLC, the capabilities of the access macro were extended 1) to accommodate the secondary JTAG controllers built into embedded PowerPC® cores; 2) to provide direct access to memory for initial boot code load and for messaging between the service node and the BLC chip; 3) to provide nonintrusive access to device control registers; and 4) to provide a suite of chip configuration and control registers. The BLC clock tree structure is described. It accommodates both functional requirements and requirements for enabling multiple built-in self-test domains, differentiated both by frequency and functionality. The chip features a debug port that allows observation of critical chip signals at full speed.*

Introduction

The control, test, and bring-up infrastructure for the Blue Gene*/L system is based on the concept that an external computer, called the *service node*, has to be able to control the system, including the individual Blue Gene/L link (BLL) and Blue Gene/L compute (BLC) chips, to the lowest levels of granularity [1]. The service node, an IBM pSeries* server running Linux**, manages a private 100-Mb/s Ethernet network dedicated to system management. The leaf nodes on this control management network are control–FPGA (field-programmable gate array) chips, one on each service card, node card, or link card. The control–FPGA (CFPGA) chip converts Ethernet packets, as received from the service node, to the appropriate protocol to communicate with various chips on each card. For the BLL and BLC chips, the Ethernet packets are converted to JTAG (IEEE Standard 1149.1, developed by the Joint Test Action Group) [2] commands. The CFPGA chip drives up to 36 separate JTAG ports in a star configuration: one JTAG port for each BLC chip deployed on a node card or one JTAG port for each BLL chip on a link card. By making use of the broadcast capabilities inherent in the Ethernet network and in the CFPGA chip, the service node can control any set of BLC or BLL chips simultaneously. Finally, the CFPGA chip collects the replies from the various chips and sends the information back to the service node as Ethernet packets.

Thus, from the viewpoint of each BLL or BLC chip, all control, test, and bring-up is governed through its JTAG port communicating with the service node. The on-chip logic interfacing with the JTAG port is known as the test access port (TAP) controller. In common with many eServer* chips, the access macro [3] has been chosen as TAP controller.

For the BLL chip, which is a relatively simple application-specific integrated circuit (ASIC) containing neither cores nor arrays, the standard access macro has all the functionality needed for control, test, and bring-up. The BLC chip, on the other hand, is a complex system-on-a-chip (SoC) design that integrates several hard and soft cores with application-specific chip logic.

0018-8646/05/$5.00 © 2005 IBM

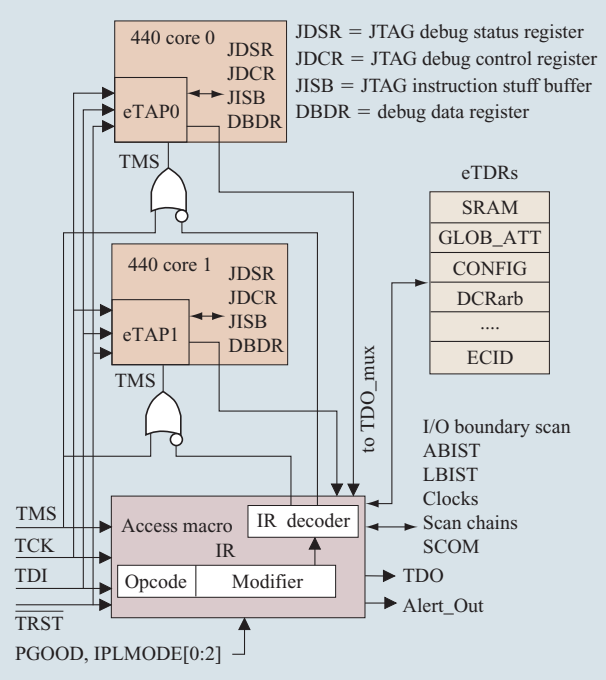

Figure 1

Arrangement of the access macro and embedded TAP (eTAP) controllers of the PowerPC 440 cores. The extended Test Data Registers (eTDRs) are discussed in the text and in Table 1.

In particular, the chip includes two embedded PowerPC* 440 processor cores (PPC440), each containing a small IEEE 1149.1-compliant TAP controller for debugging purposes [4]. To accommodate these secondary TAP controllers and meet other requirements, it was necessary to design various extensions to the access macro.

Access controller and extensions

IEEE 1149.1 describes the JTAG port as having four input signals: test clock (TCK), test mode select (TMS), test data input (TDI), and test reset (TRST), as well as one output: test data output (TDO). These are the primary input/output (I/O) signals of the chip.

For chips with the access macro as the TAP controller, two more I/O signals are added: the power good (PGOOD) input and Alert_Out output. Alert_Out is dot-ORed across a group of BLC chips on a node card (or across a group of BLL chips on a link card) to drive an input of the CFPGA chip. It provides a mechanism for a BLC or BLL chip to signal back to the external service node in case of a severe error or interrupt that requires service node intervention. PGOOD is a common signal for a group of chips on a node or link card. It is controlled by the CFPGA chip—and, thus, ultimately by the service node—to signal that the power supplies are on and stable. Upon a rising edge of PGOOD, the chip goes through an initialization sequence.

In addition, and optionally, IPLMODE[0:2] inputs can be defined that are sampled by the access macro at power-on and that control whether or not to automatically run logic built-in self test (LBIST) and/or array built-in self test (ABIST), and whether or not to start the clocks automatically as part of the power-on sequence. For the BLL and BLC chips, the choice was made not to run these tests automatically at power-on, but to leave the self tests and clock start under explicit control of the service node.

The access macro provides the following functions:

- Boundary scan, in accordance with IEEE Standard 1149.1.
- Scan communications (SCOM) interface to read and write on-chip registers.
- Control of clock tree, LBIST, and ABIST via the SCOM interface.
- Ability to connect individual latch scan chains between TDI and TDO for low-level debug.

The above standard access macro facilities are sufficient for the BLL chip. However, the BLC chip poses additional requirements:

- To provide a suite of chip configuration and control registers.
- To accommodate the TAP controllers of the embedded PowerPC cores.
- To provide direct access to memory for initial boot code load and for messaging between the service node and the BLC chip.
- To provide nonintrusive access to device control registers.

These requirements were met by a number of changes and extensions to the original access macro, as shown in **Figure 1**.

Test data registers

IEEE Standard 1149.1 [2] allows great flexibility in defining test data registers (TDRs) and the instructions to select them. This flexibility can be used to provide a suite of configuration and status registers and to access the chip logic for test, bring-up, and debugging purposes.

The standard access macro contains a limited set of TDRs. For the BLC chip, the TDR set was significantly extended with external TDRs. An overview of the TDRs for the BLC chip is given in **Table 1**.

The instruction register (IR) of access is 32 bits long and is divided into an opcode field and a modifier field.

Table 1 Test data registers (TDRs) for the Blue Gene/L compute chip. ("v" indicates a valid bit.)

TDR	*Bits*	*JTAG access*	*Description*
		Standard access TDRs	
Access_Status	32	R/W	Error reporting on access instruction execution.
Access_Options	32	R/W	Mask errors, timeouts, etc. for access.
Shadow IR	32	R	Shadow instruction register: contains last access instruction. On error, holds offending instruction.
1-bit bypass	1		IEEE Standard 1149.1 1-bit bypass.
32-bit bypass/CRC read	32	R	32-bit bypass; will contain last cyclic redundancy check (CRC) on read.
CRC register	32	R	Access transactions can be safeguarded using a CRC checking mechanism.
SCOM scan register	64	R/W	Staging register for SCOM reads and writes.
SCOM status	32	R	Error reporting on SCOM transactions.
JTAG IDcode	32	R	Uniquely identifies chip type, version, and manufacturer per IEEE Standard 1149.1.
Access version/DI/WTmode	32	R/W	Access version; bits to control I/O drive inhibit and wire test mode.
GP1_REG	32	R/W	General-purpose registers: PLL multiplier, range, tune bits, and miscellaneous control bits.
GP2_REG	32	R/W	
GP3_REG	32	R/W	
		Extended TDRS	
GP1_REG_MASK	32 + v	R/W	Selectively allow software on PPC440 cores to influence the corresponding GPn_REG settings.
GP2_REG_MASK	32 + v	R/W	
GP3_REG_MASK	32 + v	R/W	
PSRO	64	R/W	Readout facility for internal performance screen ring oscillator.
SRAM_CNTL	32 + v	R/W	Provide direct access to a segment of static random access memory; discussed in text.
SRAM_DATA	64 + v	R/W	
GLOB_ATT	32 + v	R/W	Global attention register: mechanism for signaling events between service node and BLC chip.
GLOB_ATT_MASK	32 + v	R/W	
DCR_ADDRESS	32 + v	R/W	Mechanism for nonintrusive access to device control registers; discussed in text.
DCR_DATA	32 + v	R/W	
STATUS	160 + v	R	Detailed status of self tests.
CONFIG	32 + v	R/W	Configuration bits to enable functional units and I/O ports. CONFIG_MASK selectively allows software on PPC440 cores to influence the settings.
CONFIG_MASK	32 + v	R/W	
ECID	112 + v	R	Electronic chip identification: per-chip unique identifier, used for tracking purposes.
DEBUG_CFG	32 + v	R/W	Configures debug port; discussed in text.
CLOCKS	32 + v	R/W	Enable clock subdomains. Discussed in text. CLOCKS_MASK selectively allows software on PPC440 cores to influence the settings.
CLOCKS_MASK	32 + v	R/W	
CLOCK_INTF_status	32 + v	R	PLL observation bits.
MACHINE_CHKS	64 + v	R	Capture machine check signals from on-chip logic. MACHINE_CHKS_MASK determines whether a particular machine check is signaled back to service node.
MACHINE_CHKS_MASK	64 + v	R/W	
PATCH	256 + v	R/W	Miscellaneous bits, associated with switching optional features on and off.

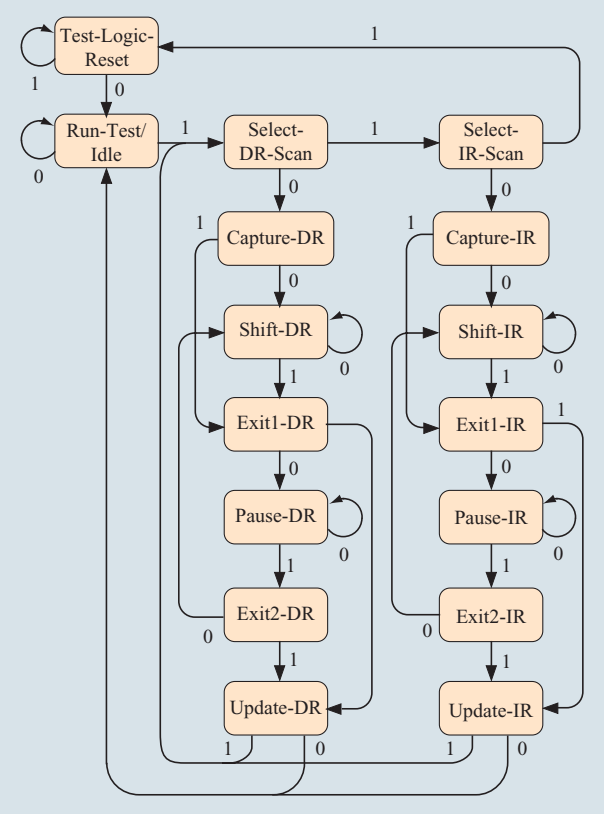

Figure 2

IEEE 1149.1 TAP controller state diagram. State transitions occur according to the indicated values of TMS, sampled at a rising edge of TCK. (From [2], reprinted with permission; ©1993 IEEE. All rights reserved.)

Opcodes define the IEEE 1149.1-compliant commands, such as TDR scan, SCOM read and write commands, and scan-ring-related commands. Typically, TDR addresses, SCOM addresses, and scan-ring addresses are encoded in the modifier field.

The standard access TDRs are listed at the top of Table 1, up to and including GP3_REG. The extended TDRs comprise the rest of the table. It is noteworthy that in the Bits column of Table 1, the extended TDRs are typically shown to have a *valid bit* in addition to the actual register contents. The purpose of this bit, introduced by the embedded TAP controllers of the PPC4xx series cores discussed below, is as follows. As can be seen in the JTAG TAP controller state diagram of **Figure 2**, the TAP controller state machine always follows a trajectory that traverses shift-DR and update-DR in order. This implies that new data, shifted into a TDR during shift-DR, is made visible to the on-chip logic at the update-DR step. A "blind" nondestructive read (capture-DR, shift-DR with results being shifted out and zeros simultaneously being shifted in, and no update-DR) is therefore not immediately possible. The addition of a "valid bit" to the actual shift register rectifies this situation. If the shift register valid bit is zero after shift-DR completes, then on update-DR, the transfer from the shift register of the TDR to the output latches is suppressed. If the valid bit is set to 1 after shift-DR completes, then on update-DR, the content of the TDR shift register is transferred to the output latches. Thus, effectively, when the service node issues a capture-DR/shift-DR/update-DR sequence with *valid bit = 0*, it nondestructively reads from the TDR; with *valid bit = 1*, it reads from and writes to the TDR.

Accommodation of embedded PowerPC cores

The BLC SoC design integrates several hard cores and soft cores with the application-specific chip logic. In particular, the chip includes two embedded PPC440 processor cores, each of which contains a small IEEE 1149.1-compliant TAP controller for debugging purposes [4]. Access to these embedded TAP (eTAP) controllers is coordinated with the main access TAP controller in the following manner, which is a variant of the configurations described in [5]. First, as shown in Figure 1, the embedded TAP controllers, eTAP0 and eTAP1, share the primary TCK, TDI, and TRST inputs with access. However, the access instruction register decoder controls the TMS inputs of the eTAPs. Referring to the JTAG TAP controller state diagram in Figure 2, the procedure to use the PPC440 eTAP consists of the following steps:

1. Cycle the master TAP controller (access) to the shift-IR state.
2. Shift an instruction into the master IR to activate the targeted eTAP. This instruction is defined such that the opcode is a no-op for the master TAP controller, and the modifier field contains a value that activates both the TMS gating of the selected eTAP and the appropriate selection of the TDO multiplexer. In the update-IR state of the master TAP controller (following completion of shift-IR), the newly selected eTAP is then switched between the TDI and TDO pins, and TMS for the selected eTAP is enabled. The newly selected eTAP "wakes up" in the test-logic-reset state.
3. With TMS = 0 for one or more TCK cycles, the master TAP transitions from update-IR to the run-test/idle state; simultaneously, the selected eTAP transitions from test-logic-reset to run-test/idle. The master TAP and the selected eTAP are now synchronized for as long as the current eTAP stays selected.
4. Master TAP and the selected eTAP are then synchronously cycled to the shift-IR state, and an

instruction is shifted in. Because of the parallel configuration of the master TAP IR and eTAP IR, the same instruction bits are shifted into both instruction registers. The IR of the PPC440 eTAPs is four bits wide and therefore decodes the leftmost four bits of the instruction to select the PPC440 TDRs indicated in Figure 1: JTAG debug status register (JDSR), JTAG debug control register (JDCR), JTAG instruction stuff buffer (JISB), and debug data register (DBDR).

These leftmost four instruction bits (which cannot be x "0" or x "F") are ignored by the master TAP; the rightmost 28 instruction bits have to stay defined so as to keep selecting the current eTAP.

5. In the update-IR state, after completion of shift-IR, the selected eTAP TDR (JDSR, JDCR, JISB, or DBDR) is now switched between TDI and TDO.
6. Data can now be captured into and updated from the selected TDR by cycling the eTAP through the capture-DR/shift-DR/update-DR states any number of consecutive times, as required.

If the sequence of JTAG instructions for a particular eTAP is interrupted by another JTAG instruction for access or for another eTAP, the modifier field decoding logic switches off the TMS gate in the update-IR step, forcing the TMS input of the previously selected eTAP to "1." This causes that eTAP to return to the test-logic-reset state after three TCK cycles.

Alternative configurations of embedded TAP controllers [5] use TRST selection logic instead of TMS selection logic. We decided to use TMS to guarantee an orderly clocked progression of states, as opposed to the asynchronous reset action of TRST.

IBM RISCWatch software [4], a debugger for embedded PowerPC cores, was upgraded to work with the master/slave TAP controller configuration described above.

Direct access to memory

The BLC chip allows direct access from the JTAG port to a 16-KB section of memory, physically implemented as static random access memory (SRAM) and located in the L2 area of the chip. Logically, this SRAM is located at the top of the 32-bit decoded address space of the PPC440 cores and contains, by default, the reset vector. This is the address of the first instruction to be executed when the PPC440 is released from reset. Thus, direct access from JTAG to this 16-KB SRAM allows the service node to load the initial PPC440 boot code.

After boot-up, the JTAG-to-SRAM facility can be used for messaging between the service node and the PPC440 cores. Of course, the SRAM can be used for other general memory purposes as well.

The JTAG-to-and-from-SRAM operations are defined via two TDRs: SRAM_CNTL and SRAM_DATA. The SRAM_CNTL register contains fields for opcodes, error information, and address. The operations are as follows:

- *Read:* From the SRAM address into the SRAM_DATA register.
- *Write:* From the SRAM_DATA register to the SRAM address.
- *Swap:* Read from the SRAM address into the SRAM_DATA register and scan out, while simultaneously scanning new data into the SRAM_DATA register, which is then written to the SRAM address.
- *Stuff:* Write the contents of SRAM_DATA identically to multiple consecutive SRAM addresses.

The efficiency of these operations is enhanced by a prefetch mechanism for read operations and an automatic address increment mechanism for consecutive SRAM accesses. In addition to regular data reads and writes, which apply error checking and correction (ECC), uncorrected data or ECC overrides can be read or written.

Access from JTAG to the SRAM is arbitrated with other SRAM accesses. The JTAG state machine progresses asynchronously and without handshaking, thereby imposing some real-time requirements. Thus, the JTAG read and write requests to the SRAM are given highest priority in arbitration, but, of course, demand very little bandwidth.

Device control registers

The PPC440 core supports a device control register (DCR) interface for the purpose of controlling other logic on the chip through software running on the PPC440 core. The DCR interface consists of an address bus, an input data bus, an output data bus, and some control signals. Most functional units are connected to the DCR bus via a DCR slave interface. Multiple slaves can be connected in a single ring, in multiple rings, or in a star topology.

The original DCR bus architecture supports a single processor core as master. To allow for two PPC440 processor cores, and thus multiple DCR masters, we extended this architecture by implementing a DCR bus arbiter. The DCR bus arbiter uses a *least recently used* selection scheme.

Being controlled by the PPC440 cores, the DCRs comprise essentially the user-accessible or system-

software-accessible status and configuration of the chip. It is important for debugging purposes that the service node be able to read and write the DCRs as well, even when the PPC440 cores are hung. To this end, two JTAG TDRs, DCR_ADDRESS and DCR_DATA, are interfaced with the DCR bus arbiter and act as a third DCR master.

In addition to the debug functionality, this JTAG-to-DCR master mechanism also enables nonintrusive performance monitoring without affecting programs running on the PPC440 cores. It does this by allowing the service node to read performance counters and other status indicators implemented as DCRs.

The converse mechanism, JTAG TDR-to-DCR slave, was also implemented, because it was deemed important for the software running on the PPC440 cores to at least be able to observe the contents of the JTAG TDRs, and in some cases, where explicitly allowed by the service node, to actually take over facilities that are nominally under TDR control. For example, selected clock-enables can be put under software control for dynamic power-down purposes. This was implemented by interfacing most of the JTAG TDRs in Table 1 with one or more DCR registers.

Clock tree structure

The BLC SoC chip integrates the functions of a number of different chips in a more traditional computer design. Consequently, it contains several different clock domains that are differentiated either by function or by frequency. The clock tree structure of the chip was designed with the following criteria in mind:

- Variable frequency ratio between the compute and memory system on the one side, and the torus [6] and collective[1] communication systems on the other. This allows the compute and memory system to run off a fixed 700-MHz frequency (and divisions thereof), while the serial parts of the torus and collective logic are driven off the central system clock, running at either 350 or 700 MHz. (In the first-pass design, the logic also supported a 1400-MHz system clock. In the second-pass design, this option was dropped.)
- Flexibility in gating off clocks for functional subdomains. This is used functionally: BLC chips used as compute chips do not use the Gigabit Ethernet subsystem, and BLC chips used as I/O chips do not use the torus subsystem. Also, subdomain clock gating allows more flexibility in debug situations.
- Support for built-in self test (BIST): both LBIST and ABIST. As much as possible, these built-in self-test functions exercise the logic at speed.
- Support for debug functions, such as scanning of internal scan chains and IEEE 1149.1-compliant boundary scan.

The latter two items are part of the standard IBM Rochester clock-tree design methodology used to design many chips for eSeries computers, but the first two items are unique to the BLC chip.

The structure of the clock tree is schematically depicted in **Figure 3** and further detailed in **Table 2**.

The system clock is distributed as a 1.5-V high-speed transceiver logic (HSTL) [7] differential signal and is received onto the chip using a differential receiver I/O book (IHSTLT in Figure 3). This clock immediately drives the double-data-rate (DDR) send and receive part of the high-speed serial I/Os associated with the torus and collective networks (domains 1A and 1B, respectively). The reference clock input of the phase-locked loop (PLL) is derived from a point equivalent to a leaf cell of the clock tree for the high-speed serial logic. This clock signal is then divided by 4 to match the speed ratio between the DDR serial bits of the I/Os and the byte-based internal torus and collective logic. This torus and collective internal logic is driven from the A-output of the PLL (domains 5J, 5K). The feedback input of the PLL is taken from a point equivalent to a leaf cell of the clock tree in this low-speed portion of the collective logic (domain 5K). With this arrangement, the internal byte-wide torus and collective logic (domains 5J, 5K) is phase-locked to the corresponding high-speed serial clock of domains 1A and 1B at one-quarter frequency, i.e., at 87.5 or 175 MHz, respectively. The range and multiplier settings on the PLL are set via the GP3_REG TDR (see Table 1) to keep the internal voltage-controlled oscillator (VCO) operating at 1400 MHz, irrespective of whether the system clock comes in at 350 or 700 MHz.

The fixed-frequency logic on the chip, i.e., processor units, the memory subsystem, and fixed-frequency parts of the I/O subsystems, are clocked off the B-output of the PLL.

A processor unit (PU) consists of a PPC440 hard core and an associated "double-hummer" floating-point unit (FPU) hard core. The BLC chip has two such PUs: PU0 and PU1. The 1400-MHz B-output of the PLL is routed to the PUs and locally divided by 2 to drive these units at 700 MHz (domains 2 and 3). Also, the I-cache read part of the L2 logic is driven at 700 MHz (domain 4). The rest of the L2 logic, and most of the rest of the memory subsystem, is driven at half this frequency, 350 MHz (domain 5). The logic surrounding the L3

[1] D. Hoenicke, M. A. Blumrich, D. Chen, A. Gara, M. E. Giampapa, P. Heidelberger, L.-K. Liu, M. Lu, V. Srinivasan, B. D. Steinmacher-Burow, T. Takken, R. B. Tremaine, A. R. Umamaheshwaran, P. Vranas, and T. J. C. Ward, "Blue Gene/L Global Collective and Barrier Networks," private communication.

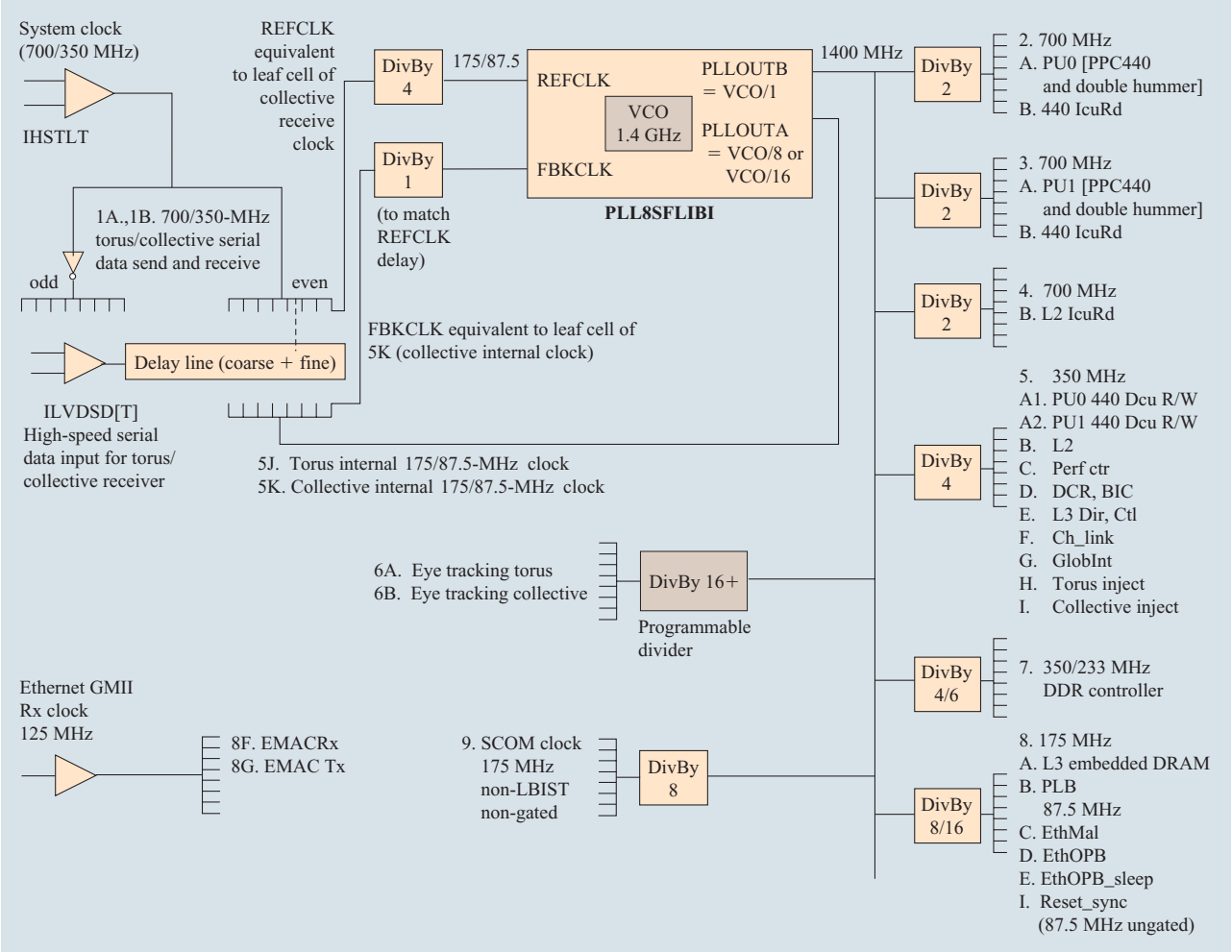

Figure 3

Blue Gene/L compute chip clock tree structure. ©2005 IEEE. Reprinted with permission. Adapted from A. A. Bright, "Creating the Blue Gene/L Supercomputer from Low Power System-on-a-Chip ASICs," *Digest of Technical Papers*, 2005 IEEE International Solid-State Circuits Conference.

embedded dynamic random access memory (embedded DRAM) is at 175 MHz (domain 8A).

The interface to the external DDR synchronous DRAM (SDRAM) memory chips can run at either 700 MHz divided by 2 (350 MHz) or divided by 3 (233 MHz) to match the speeds of available DDR SDRAM chips. The clock tree provides this variable division (domain 7).

The Gigabit Ethernet subsystem comprises a number of soft cores. The synchronous part of this logic is driven at 175 MHz and 87.5 MHz (domains 8B–8E). The Gigabit Ethernet I/O is implemented as a Gigabit Media Independent Interface (GMII) [8], driven from an external 125-MHz clock, asynchronous to the rest of the BLC chip logic (domains 8F–8H).

Finally, in the high-speed receive logic (domains 1A, 1B above), the eye of the incoming data signals is determined. The position of the eye with respect to the receiver delay line taps is subject to slow drift due to temperature and voltage variations. The eye-tracking logic operates off a very slow clock, which is programmable. In normal functional operation, this is effectively the PLL output B divided by 256 or more (domain 6, <5.5 MHz).

The major clock domains, domains 1 through 8 as described above, correspond largely to different regions for LBIST. However, for functional reasons, most major domains were further divided into functional clock subdomains. For example, domain 5 is subdivided into domains 5A through 5K. Table 2 gives an overview of the

Table 2 Overview of BLC DD2.0 clock domain structure.

LBIST domain	*LBIST speed* (MHz)	*Subdomain*	*Unit*	*Relation to PLL*	*Full speed* (MHz)	*Half speed* (MHz)
1	700	1A	Torus serial data capture/send	pre-PLL	700	350
	700	1B	Collective serial data capture/send	pre-PLL = PLL refclk	700	350
2	DC	2	PU0: PPC440 + FPU	PLL B/2	700	← same
	DC	2B	PU0: ICache Read	PLL B/2	700	← same
3	DC	3	PU1: PPC440 + FPU	PLL B/2	700	← same
	DC	3B	PU1: ICache Read	PLL B/2	700	← same
4	700	4B	L2 (ICache Read)	PLL B/2	700	← same
5A	DC	5A2	PU0: DCache R/W	PLL B/4	350	← same
	DC	5A3	PU1: DCache R/W	PLL B/4	350	← same
5BI	350	5B	L2 (DCache R/W) + L2 other: SRAM ctl, L2/L3 interface, lock box	PLL B/4	350	← same
	350	5C	Performance counter	PLL B/4	350	← same
	350	5D	DCR, BIC	PLL B/4	350	← same
	350	5E	L3 dir, L3 control	PLL B/4	350	← same
	350	5F	Ch_link	PLL B/4	350	← same
	350	5G	Global interrupts	PLL B/4	350	← same
	350	5H	Torus injection	PLL B/4	350	← same
	350	5I	Collective injection	PLL B/4	350	← same
5JK	175	5J	Torus internal	PLL A	175	87.5
	175	5K	Collective internal	PLL A output = PLL fbkclk	175	87.5
6	87.5	6A	Torus eye tracking	PLL B output divided down ≥16	87.5	← same
	87.5	6B	Collective eye tracking	PLL B output divided down ≥16	87.5	← same
7	350	7	External DRAM controller	PLL B/(4 or 6)	350	← same
8A	175	8A	L3 embedded DRAM self-refresh	PLL B/8	175	← same
	scan only	8B	PLB	PLL B/8	175	← same
	scan only	8C	EthMal	PLL B/16	87.5	← same
	scan only	8D	EthOPB	PLL B/16	87.5	← same
	scan only	8E	EthOPB_sleep	PLL B/16	87.5	← same
	scan only	8F	EMAC4 Rx	Primary input	125	← same
	scan only	8G	EMAC4 Tx (GMII)	Primary input	125	← same
	scan only	8H	EMAC4 Rx1 (PMA)	—	0	← same
	scan only	8I	Testint Sync – ungated	PLL B/16	87.5	← same
Misc	No LBIST		SCOM – ungated	Bufrefclk	175	← same

clock domains and subdomains, and the LBIST structure. For each functional subdomain, the CLOCKS TDR contains a clock_enable bit, so that the service node controls the clocking of functional domains. As noted before, BLC chips used as compute chips do not use the Gigabit Ethernet subsystem; on the other hand, BLC chips used as I/O chips do not use the torus subsystem. Clocks for the unused logic can be shut off. As an additional feature, and only where specifically allowed by the service node through the CLOCKS_MASK TDR, software running on the PPC440 cores can also enable or disable functional clock domains by writing to a DCR. For example, on BLC chips used as I/O chips, portions of the Ethernet subsystem (domain 8E) can be put to sleep this way in periods in which there is no data traffic.

Logic built-in self test (LBIST)

The BLC chip logic is designed according to level-sensitive scan design (LSSD) rules [9] and uses master–slave (L1–L2) latches almost exclusively. The chip conforms to all requirements for standard IBM ASIC test methodology [10] and is tested as such after

manufacturing, both at wafer and module test stations, achieving a high standard of testability (more than 99.7% dc test coverage) and reliability.

However, there remains a small chance of chips failing in the system over time. Since the Blue Gene/L supercomputer will contain more than 1,000 BLC chips per system rack, it was decided at an early stage of the design that the chip should be self-testable to aid in-system diagnostics. In addition, during manufacturing test of either chips or cards, at-speed self test can provide for an effective screening against ac defects. The access macro supports an LBIST function. During LBIST operation, a pseudo-random pattern generator (PRPG) generates randomized bit patterns, which are scanned into 248 short scan chains known as *STUMPS channels* [11, 12]. This is followed by a launch/capture cycle at the rated clock speed, during which the scanned-in patterns propagate through the combinatorial logic, with results captured in downstream latches. The captured patterns are scanned out and collected into a multiple-input signature register (MISR). With a deterministic PRPG seed and a fixed number of scan/launch/capture sequences, the MISR generates a stable "golden" signature for a good chip, whereas any fault (if exposed by the patterns) results in a deviation from the golden signature. LBIST is run separately for each clock speed domain indicated in the first column of Table 2, with an expected golden signature for each run. These golden signatures are verified against predicted signatures generated using Cadence Encounter** Test Solutions tools [13] (formerly the IBM TestBench tool).

In addition to the standard LSSD design requirements, the implementation of LBIST poses extra constraints on scan chain and scan clock design. We handled this by explicitly instantiating the LSSD scan chains and scan clocks in the Very high-speed integrated circuit Hardware Description Language (VHDL), labeling both scan chains and scan clocks at the macro level with the subdomain name. In some limited cases, the I/O boundary scan structures [5] required small modifications. Most significantly, however, multiplexing structures to concatenate the individual macro-level scan chains were implemented at the top level of the chip logic hierarchy. These multiplexing structures perform three tasks:

1. At manufacturing wafer and module test, present to the tester 50 length-balanced scan chains compliant with a reduced-pin-count test methodology [10] with on-product MISR [14].
2. For in-system LBIST, configure the scan chains into 248 short STUMPS channels.
3. For low-level debug, configure the scan chains into a limited number of scan rings (typically, one per major clock domain) and, upon a specific JTAG command, switch a selected scan ring (or a set of multiple concatenated scan rings) between TDI and TDO. This allows a dump of the chip state to the service node for detailed register inspection. The term *scan ring* refers to the functionality in the access macro to recirculate the output back into the input under this type of scanning, so that after reading a full scan ring, the chip state is undisturbed.

Table 3 LBIST patterns and coverage per domain.

Domain	*Pattern count* (000s)	*Active logic* (%)	*Coverage of active logic* (%)
Scan chains		38.31	100.00
1	16	1.30	96.17
4B	16	1.10	96.14
5BI	256 + 16	17.09	94.09
5JK	16	11.78	93.32
6	16	2.37	95.90
7	16	4.25	95.25
8	16	2.63	96.14
	Totals	69.16	97.09

Because the BLC is an SoC, not all logic is controlled by the design team. Consequently the BLC chip suffers from some detractors that prevent us from achieving the ideal [15] of near-complete coverage with at-speed LBIST.

At-speed LBIST requires that clock splitters are able to be gated on a single-cycle basis for launch and capture. Unfortunately, a number of the ASIC library hard and soft cores imported into the BLC design do not have this functionality. As a result of these limitations, LBIST can be run at speed only for those subdomains in the table that have a numerical entry in the LBIST speed column of Table 2. It should be noted that during LBIST, all latches in all scan chains participate in the scanning of pseudo-random patterns, so latches in clock domains in which LBIST cannot be run at speed still contribute to a randomized environment for the domains where at-speed LBIST is run.

Coverage metrics for the LBIST domains are given in **Table 3**. In the table, active logic percentage is defined as the total number of single stuck-at faults observable by a given LBIST run on the chip (given the clock gating for the subdomains under test and taking into account any logic masked off during LBIST), divided by the total number of stuck-at faults on the chip. Note that 38.31% of the active logic is tested using scanning alone. This includes some access and clock tree logic, but consists primarily of the latches in the scan chains. Because all scan chains are active during the LBIST of each subdomain, this base percentage is subtracted in order to

arrive at the active logic percentages per domain shown. The active logic percentages per domain (plus the scan) add up to more than the total of 69.16% active logic, because some single stuck-at faults can be active across interfaces between domains and can end up being counted in more than one LBIST subdomain. The remaining 30.84% of stuck-at faults *not* subject to LBIST includes the PPC440 and FPU cores, as well as the Ethernet subsystem. On the 69.16% of the logic that is subject to LBIST and for the pattern counts indicated, the total LBIST coverage is 97.09%. Historically, the expectation for ac transition fault coverage is up to 10% less because of latch adjacency limitations. However, in practice, both ac and dc coverage appear to be much better, probably because of the tendency for faults to be grouped, leading to multiple failures when any fail occurs, and because of the limitations of the single stuck-at fault test coverage model.

Improvements to the LBIST coverage can be gained by using different clock sequences. The original LBIST clock sequence preserves data captured in L1 latches on scan-out. This is an issue with register arrays, where, for test purposes, half the bits are implemented as L1 latches and the other half as L2 latches. For domains 5B–5I (5BI in Table 2), an extra pattern set with a different clock sequence was added to the original 256K patterns. The new sequence preserves L2 latch contents on scan-out and allowed an increase in coverage from 90.60% to 94.09% on this domain using only 16K extra patterns.

Standard IBM ASIC manufacturing testing comprises LSSD testing at dc speeds, both at the wafer and at the module level, as well as an optional ac test, at limited speed, of the final module. Tested modules are then shipped to the electronic card assembly and test (ECAT) vendor. After assembly of the modules on two-way circuit cards, the capability of the chip to run LBIST at full speed is first exercised on a functional test station at the ECAT vendor.

The critical timing paths on the chip are in the PUs, which, as Table 2 shows, are subject only to DC-LBIST. Thus, for the BLC chip, the kernel of DGEMM [16] was used to establish a frequency-screening test point compatible with a 700-MHz performance specification over the lifetime of the system. (DGEMM is a double-precision general matrix multiply subroutine heavily used in the Linpack benchmark [17].)

During card test, at-speed LBIST is run on each BLC chip on the domains amenable to it (see Table 2) to ensure that there are no ac defects in those domains up to the screening frequency. An additional suite of functional tests, including DGEMM, is run to ensure functionality at speed of the subsystems not amenable to LBIST (PUs, Ethernet, and communication with external DRAMs).

Array built-in self test (ABIST)

The access macro provides for an ABIST function used to exercise the BIST macros [18] associated with each SRAM array on the chip and to collect the results. Initialization of SRAM BIST engines is simply done by a zero-flush of the scan chains.

For the BLC chip, we have also enabled the in-system ABIST function for the embedded DRAMs [19]. In contrast to the SRAM ABIST, the BIST engines in the embedded DRAMs must be initialized by a specific scan string, and the results scanned out. This is done by using the low-level scan-ring access capability provided by the access macro. By using the combination of access and embedded DRAMs, the BLC chip is the first IBM ASIC that enabled and uses in-system ABIST of the embedded DRAMs.

Debug port

To facilitate low-level debugging, the BLC chip features a debug port offering the following facilities:

- A clock observation output pin. Every clock associated with each clock subdomain in Table 2, including special clocks for arrays, is routed to this chip output pin via a multiplexer controlled by the DEBUG_CFG TDR (see Table 1). In addition, a number of PLL observation signals are routed through this multiplexer. This facility allows debugging of the PLL and the clock tree.
- A 32-bit synchronous I/O port, intended to be routed to a logic analyzer. This I/O port is overlaid on the Ethernet I/O port and can be used only when the Ethernet is not enabled. Numerous on-chip signals of interest to the chip logic designers are multiplexed onto this port, again under control of the DEBUG_CFG TDR. When this debug port is in use, the clock observation output pin drives a synchronous clock signal for the logic analyzer.

With the debug multiplexers for clock and data under control of a TDR, that is, under control of the service node, the operation of the debug port does not interfere with the normal operation of the chip.

Conclusion

The Blue Gene/L compute chip is built as a system-on-a-chip ASIC. However, it features many of the self-test, bring-up, and debug facilities of a custom-designed microprocessor. By extending the functionality of the standard eSeries access macro (in particular, the JTAG test data register definitions), the BLC chip design makes available to an external service node an extensive suite of chip configuration and control registers, direct access to

SRAM memory, nonintrusive access to device control registers, and access to the debug facilities of the embedded PowerPC cores. The combination of access and the clock tree design supports a variety of frequency domains, multiple modes of operation, and built-in self test for both logic and arrays. In aggregate, the described test, bring-up, and debug facilities are playing a substantial role in the successful bring-up of the Blue Gene/L supercomputer systems.

Acknowledgments

This work has benefited from the cooperation of many individuals in IBM Research (Yorktown Heights, New York), IBM Engineering and Technology Services (Rochester, Minnesota), and IBM Microelectronics (Burlington, Vermont and Raleigh, North Carolina). In particular, we thank Scott A. Bancroft, Daniel K. Beece, Dong Chen, James F. Daily, Tina Douskey, John R. Elliott, Christopher J. Engel, Mark E. Giampapa, Michael J. Hamilton, David F. Heidel, Brent Hilgart, Thomas J. Irene, Fariba Kasemkhani, Brian Koehler, Gerard V. Kopcsay, Anthony J. Marsala, Steven F. Oakland, Dennis R. Olson, Michael R. Ouellette, Dennis M. Rickert, Nabil A. Rizk, Jeremy Rowland, Bruce G. Rudolph, Eric L. Skuldt, Peilin Song, Scott A. Strissel, Richard A. Swetz, Yelena Tsyrkina, Gregory S. Ulsh, Stephen D. Wyatt, and Christian Zoellin.

The Blue Gene/L project has been supported and partially funded by the Lawrence Livermore National Laboratory on behalf of the United States Department of Energy, under Lawrence Livermore National Laboratory Subcontract No. B517552.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Linus Torvalds or Cadence Design Systems, Inc. in the United States, other countries, or both.

References

1. P. Coteus, H. R. Bickford, T. M. Cipolla, P. G. Crumley, A. Gara, S. A. Hall, G. V. Kopcsay, A. P. Lanzetta, L. S. Mok, R. Rand, R. Swetz, T. Takken, P. La Rocca, C. Marroquin, P. R. Germann, and M. J. Jeanson, "Packaging the Blue Gene/L Supercomputer," *IBM J. Res. & Dev.* **49**, No. 2/3, 213–248 (2005, this issue).
2. *IEEE Standard 1149.1a-1993*, "IEEE Standard Test Access Port and Boundary-Scan Architecture," IEEE, New York, October 1993. ISBN: 1559373504.
3. S. M. Douskey, M. C. Cogswell, G. R. Currier, J. R. Elliott, S. D. Vincent, J. M. Wallin, and P. L. Wiltgen, "Multi-Core Chip Providing External Core Access with Regular Operation Function Interface and Predetermined Service Operation Services Interface Comprising Core Interface Units and Masters Interface Unit," U.S. Patent 6,115,763, February 2000.
4. See *http://www.ibm.com/chips/products/powerpc/tools/riscwatc.html*.
5. IBM Corporation, IBM ASIC Application Note SA2282-04: IEEE 1149.1 Boundary Scan in IBM ASICS, 2004.
6. N. R. Adiga, M. A. Blumrich, D. Chen, P. Coteus, A. Gara, M. E. Giampapa, P. Heidelberger, S. Singh, B. D. Steinmacher-Burow, T. Takken, M. Tsao, and P. Vranas, "Blue Gene/L Torus Interconnection Network," *IBM J. Res. & Dev.* **49**, No. 2/3, 265–276 (2005, this issue).
7. *Electronic Industries Association (EIA)/JEDEC Standard EIA/JESD8-6*. See *http://www.jedec.org/download/search/jesd8-6.pdf*.
8. *IEEE Standard 802.3, clause 35*, "Reconciliation Sublayer (RS) and Gigabit Media Independent Interface (GMII)," IEEE, Piscataway, NJ, 2000. See *http://www.ieee.org*.
9. E. B. Eichelberger and T. W. Williams, "A Logic Design Structure for LSI Testability," *Proceedings of the 14th Design Automation Conference*, 1977, pp. 462–468.
10. P. S. Gillis, T. S. Guzowski, B. L. Keller, and R. H. Kerr, "Test Methodologies and Design Automation for IBM ASICs," *IBM J. Res. & Dev.* **40**, No. 4, 461–474 (1996).
11. P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Logic Modules," *Proceedings of the IEEE International Test Conference*, 1982, pp. 200–204.
12. B. L. Keller and T. J. Snethen, "Built-In Self-Test Support in the IBM Engineering Design System," *IBM J. Res. & Dev.* **34**, No. 2/3, 406–415 (1990).
13. See *http://www.cadence.com/products/digital_ic/encountertest/index.aspx*.
14. C. Barnhart, V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, "OPMISR: The Foundation for Compressed ATPG Vectors," *Proceedings of the International Test Conference*, 2001, pp. 748–757.
15. M. P. Kusko, B. J. Robbins, T. J. Koprowski, and W. V. Huott, "99% AC Test Coverage Using Only LBIST on the 1-GHz IBM S/390 zSeries 900 Microprocessor," *Proceedings of the International Test Conference*, 2001, pp. 586–592.
16. J. J. Dongarra, J. Du Croz, S. Hammarling, and I. S. Duff, "A Set of Level 3 Basic Linear Algebra Subprograms," *ACM Trans. Math. Software* **16**, No. 1, 1–17 (1990).
17. J. J. Dongarra, P. Luszczek, and A. Petitet, "The LINPACK Benchmark: Past, Present, and Future," *Concurrency & Computation: Practice & Experience* **15**, No. 9, 803–820 (2003).
18. C. Chai, J. H. Fischer, M. R. Ouellette, and M. H. Wood, "Compilable Address Magnitude Comparator for Memory Array Self-Testing," U.S. Patent 6,658,610, December 2003.
19. P. Jakobsen, J. Dreibelbis, G. Pomichter, D. Anand, J. Barth, M. Nelms, J. Leach, and G. Belansek, "Embedded DRAM Built-In Self-Test and Methodology for Test Insertion," *Proceedings of the International Test Conference*, 2001, pp. 975–984.

Received May 6, 2004; accepted for publication July 12, 2004; Internet publication April 1, 2005

Ruud A. Haring *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (ruud@us.ibm.com).* Dr. Haring is a Research Staff Member at the IBM Thomas J. Watson Research Center. He received B.S., M.S., and Ph.D. degrees in physics from Leyden University, the Netherlands, in 1977, 1979, and 1984, respectively. Upon joining IBM in 1984, he initially studied surface science aspects of plasma processing. Beginning in 1992, he became involved in electronic circuit design on both microprocessors and application-specific integrated circuits (ASICs). He is currently responsible for the synthesis, physical design, and test aspects of the Blue Gene chip designs. Dr. Haring has received an IBM Outstanding Technical Achievement Award for his contributions to the z900 mainframe, and he holds several patents. His research interests include circuit design and optimization, design for testability, and ASIC design. Dr. Haring is a Senior Member of the IEEE.

Ralph Bellofatto *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (ralphbel@us.ibm.com).* Mr. Bellofatto is a Senior Software Engineer. He has been responsible for various aspects of hardware system verification and control system programming on the Blue Gene/L project. He received B.S. and M.S. degrees from Ithaca College in 1979 and 1980, respectively. He has worked as a software engineer in a variety of industries. Mr. Bellofatto's interests include computer architecture, performance analysis and tuning, network architecture, ASIC design, and systems architecture and design. He is currently working on the control system for Blue Gene/L.

Arthur A. Bright *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (brightaa@us.ibm.com).* Dr. Bright received B.A. and M.A. degrees in physics from Dartmouth College and a Ph.D. degree in physics from the University of Pennsylvania. He previously worked at Union Carbide doing research on the properties of carbon fibers. He joined IBM in 1978, initially working on Josephson junction technology and subsequently on plasma processing in silicon technology. He served as president of the Thin Films Division of the American Vacuum Society from 1991 to 1992, and was on the editorial board of the *Journal of Vacuum Science and Technology* from 1992 to 1994. Upon becoming a circuit designer, he worked on custom microprocessor design and system-on-a-chip ASICs. In 1999 Dr. Bright became involved with the Blue Gene/L project, where his focus has been on synthesis, timing, and physical design issues. He holds three patents.

Paul G. Crumley *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (pgc@us.ibm.com).* Mr. Crumley has worked in the IBM Research Division for more than 20 years. His work and interests span a wide range of projects, including distributed data systems, high-function workstations, operational processes, and, most recently, cellular processor support infrastructure.

Marc Boris Dombrowa *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (dombrowa@us.ibm.com).* Mr. Dombrowa received his Dipl.-Ing. degree in electrical engineering from the University of Hannover, Germany, in 1997. He was a very large scale integration (VLSI) designer at the IBM VLSI Laboratory in Boeblingen, Germany, from 1997 to 1998, performing memory design verification and synthesis on S/390* Enterprise memory systems. From 1998 to 2000 he was assigned to the S/390 Server Division at the IBM Poughkeepsie facility to perform custom circuit design. He moved to Blue Gene/L cellular systems chip development in 2001 and has been responsible for the high-level design, synthesis, timing, and verification of the test interface of the Blue Gene/L compute chips as well as design-for-testability transformation for the entire chip, clock-tree verification, and simulation setup for instruction program load for the chip verification teams. Mr. Dombrowa received an IBM Outstanding Achievement Award in 1998 for his S/390 contributions. He is co-inventor of one patent. His research interests include computer architecture, design for test, system bring-up, diagnostics, and ASIC design. Mr. Dombrowa is currently working on the manufacturing diagnostic software as well as the system-level rack diagnostic test suite and bring-up for the Blue Gene/L cluster.

Steven M. Douskey *IBM Engineering and Technology Services, 3605 Highway 52 N., Rochester, Minnesota 55901 (douskey@us.ibm.com).* Mr. Douskey is a Senior Engineer. He joined the Advanced Systems Development Group in 1982 after receiving a B.S.E.E. degree from the University of Nebraska. He is the team leader and architect for built-in self test (BIST) and system diagnostics structures on numerous IBM projects, recently including Blue Gene, Netfinity, and eSeries designs. Past IBM assignments have included AS/400 design for test (DFT), AS/400 problem analysis and resolution (PAR), AS/400 processor bus adapter interface hardware, and System/38 processor channel interface hardware. Mr. Douskey holds 11 U.S. patents and has published 23 technical disclosures.

Matthew R. Ellavsky *IBM Engineering and Technology Services, 3605 Highway 52 N., Rochester, Minnesota 55901 (ellavsky@us.ibm.com).* Mr. Ellavsky received his B.S. degree in electrical engineering from the University of Minnesota. He joined the IBM Server Group in 1999, and is currently a member of the Engineering and Technology Services organization. For the past five years Mr. Ellavsky has focused on clock distribution networks and clock control logic for a wide array of ASIC chips.

Balaji Gopalsamy *IBM Engineering and Technology Services, Golden Enclave, Airport Road, Bangalore 560 017 (gbalaji@in.ibm.com).* Mr. Gopalsamy is a Staff Research and Development Engineer. He received a B.E. degree in electrical and electronics engineering from Madurai Kamraj University, India, in 1998. He joined IBM in 2000, working on verification and modeling of ASICs. Mr. Gopalsamy was involved in the verification of the memory subsystem and test interface modules of the Blue Gene/L compute chip design.

Dirk Hoenicke *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (hoenicke@us.ibm.com).* Mr. Hoenicke received a Dipl. Inform. (M.S.) degree in computer science from the University of Tuebingen, Germany, in 1998. Since then, Mr. Hoenicke has worked on a wide range of aspects of two prevalent processor architectures: ESA/390 and PowerPC. He is currently a member of the Cellular Systems Chip Development Group, where he focuses on the architecture, design, verification, and implementation of the Blue Gene system-on-a-chip (SoC) supercomputer family. In particular, he was responsible for the architecture, design, and verification effort of the collective network and defined and implemented many other parts of the BG/L ASIC. His areas of expertise include high-performance computer systems and advanced memory and network architectures, as well as power-, area-, and complexity-efficient logic designs.

Thomas A. Liebsch *IBM Engineering and Technology Services, 3605 Highway 52 N., Rochester, Minnesota 55901 (liebsch@us.ibm.com)*. Mr. Liebsch has worked for IBM as an electrical engineer and programmer since 1988. He received B.S. and M.S. degrees in electrical engineering from South Dakota State University in 1985 and 1987, respectively, along with minors in mathematics and computer science. He has had numerous responsibilities involving IBM server power-on control software design, built-in self-test designs, clocking designs, and various core microprocessor logic design responsibilities. He was the system technical owner for several IBM iSeries and pSeries servers. Mr. Liebsch is currently the chief engineer working on the Blue Gene/L system.

James A. Marcella *IBM Engineering and Technology Services, 3605 Highway 52 N., Rochester, Minnesota 55901 (u560259@us.ibm.com)*. Mr. Marcella is a Senior Engineer working in the area of custom and system-on-a-chip design and implementation. His primary interest is in symmetric multiprocessor (SMP) memory subsystem design. He joined IBM in 1980 after receiving his B.S.E.E. degree from the University of Minnesota. Mr. Marcella has received three IBM Outstanding Technical Achievement Awards for his work on memory controller designs for the iSeries, pSeries, and xSeries servers. He has 11 issued U.S. patents and 19 published disclosures.

Martin Ohmacht *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (mohmacht@us.ibm.com)*. Dr. Ohmacht received his Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the University of Hannover, Germany, in 1994 and 2001, respectively. He joined the IBM Research Division in 2001 and has worked on memory subsystem architecture and implementation for the Blue Gene project. His research interests include computer architecture, design and verification of multiprocessor systems, and compiler optimizations.