SECTION 9
IEEE 1149.1-COMPLIANT INTERFACE

The MPC509 includes dedicated user-accessible test logic that is fully compatible with the *IEEE 1149.1-1990 Standard Test Access Port and Boundary Scan Architecture*. Problems associated with testing high-density circuit boards have led to development of this standard under the sponsorship of the Test Technology Committee of IEEE and the Joint Test Action Group (JTAG). The MPC509 supports circuit-board test strategies based on this standard.

This section is intended to be used with the supporting IEEE 1149.1-1990 standard. The scope of this description includes those items required by the standard to be defined and, in certain cases, provides additional information specific to the implementation. For internal details and applications of the standard, refer to the IEEE 1149.1-1990 document.

An overview of the JTAG pins on the MPC509 is shown in **Figure 9-1**.

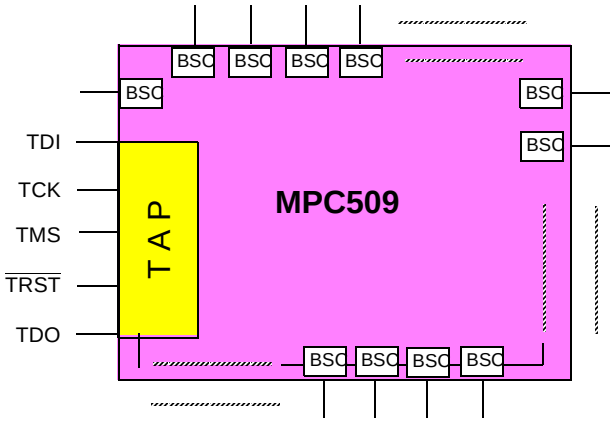

Figure 9-1 JTAG Pins

Boundary scan cells (BSC) are placed at the digital boundary of the chip (normally the package pins). The boundary scan cells are chained together to form a boundary scan register (BSR). The data is serially shifted in through the serial port (TDI) and serially shifted out through the output port (TDO).

9.1 JTAG Interface Block Diagram

A block diagram of the MPC509 implementation of the IEEE 1149.1-1990 test logic is shown in **Figure 9-2**.

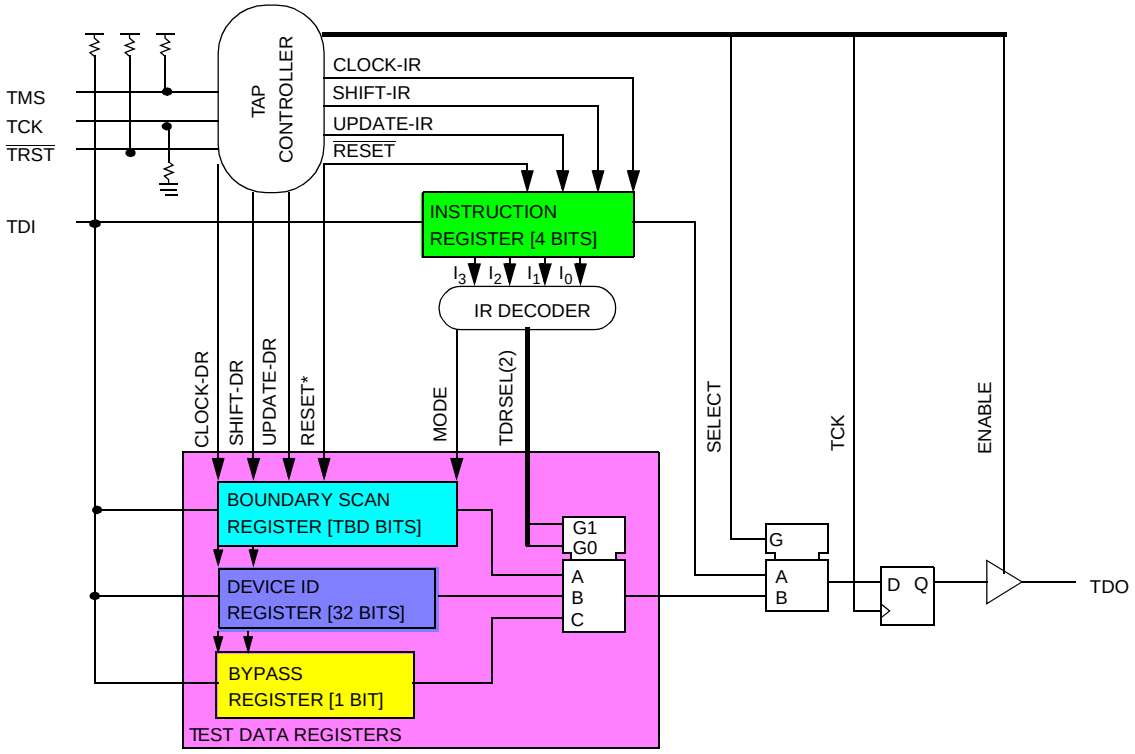

Figure 9-2 Test Logic Block Diagram

9.2 JTAG Signal Descriptions

The MPC509 has five dedicated JTAG pins, which are described in **Table 9-1**. The TDI and TDO scan ports are used to scan instructions as well as data into the various scan registers for JTAG operations. The scan operation is controlled by the test access port (TAP) controller, which in turn is controlled by the TMS input sequence.

Table 9-1 JTAG Interface Pin Descriptions

Signal Name	Input/Output	Internal Pull-Up/ Pulldown Provid-ed	Description
TDI	Input	Pull-up	Test data input pin. Sampled on the rising edge of TCK. Has pull-up resistor.
TDO	Output	None	Test data output pin. Actively driven during the shift-IR and shift-DR controller states. Changes on the falling edge of TCK. Can be placed in high-impedance state.
TMS	Input	Pull-up	Test mode select pin. Sampled on the rising edge of TCK to sequence the test controller's state machine. Has a pull-up resistor.
TCK	Input	Pulldown	Test clock input to synchronize the test logic. Has a pull-down resistor.
TRST	Input	Pull-up	TAP controller asynchronous reset. Provides initialization of the TAP controller and other logic as required by the standard. Has a pull-up resistor.

9.3 Operating Frequency

The TCK frequency must be between 5 MHz and 10 MHz. This pin is internally driven to a low value when disconnected.

9.4 TAP Controller

TRST is used to reset the TAP controller asynchronously. The TRST pin ensures that the JTAG logic does not interfere with the normal operation of the chip. This pin is optional in the JTAG specification.

The TAP controller changes state either on the rising edge of TCK or when TRST is asserted.

The TDO signal remains in a high-impedance state except during the shift-DR or shift-IR controller states. During these controller states, TDO is updated on the falling edge of TCK.

The TAP controller states are designed as specified in the IEEE 1149.1 standard.

9.5 Instruction Register

The MPC509 implementation of the IEEE 1149.1 interface includes the three mandatory public instructions (BYPASS, SAMPLE/PRELOAD, and EXTEST) and five public instructions (CLAMP, HIGHZ, EXTEST_PULLUP, TMSCAN, and IDCODE). The MPC509 contains a four-bit instruction register without parity consisting of a shift register with four parallel outputs. Data is transferred form the shift register to the parallel outputs during update-IR controller state. The four bits are used to decode eight unique instructions as shown in **Table 9-2**.

Table 9-2 Instruction Register Encoding

Code				Instruction
B3	B2	B1	B0	
1	1	1	1	BYPASS
1	1	1	0	SAMPLE/PRELOAD
1	1	0	1	IDCODE
1	1	0	0	TMSCAN
1	0	x	x	Reserved
0	1	x	x	Reserved
0	0	1	1	CLAMP
0	0	1	0	HIGHZ
0	0	0	1	EXTEST_PULLUP
0	0	0	0	EXTEST

The parallel output of the instruction register is reset to 1101 in the test-logic-reset controller state.

NOTE

This preset state is equivalent to the IDCODE instruction. In the capture-IR state, 1101 is loaded into the instruction shift register stage. New instructions can be shifted into the instruction shift register stage on Shift-IR state.

9.5.1 EXTEST (0000)

The external test (EXTEST) instruction enables the boundary scan register between TDI and TDO, including cells for all device signal and clock pins and associated control signals. The XTAL, EXTAL, XFC_S, XFC_C and XFC_I pins are associated with analog signals and are not included in the boundary scan register.

EXTEST also asserts internal reset for the MPC509 system logic for the duration of EXTEST in order to force a predictable internal state while performing external boundary scan operations.

By using the TAP, the boundary scan register is capable of:

- Scanning user-defined values into the output buffer;
- Capturing values presented to input signals; and
- Controlling the direction and value of bi-directional pins.

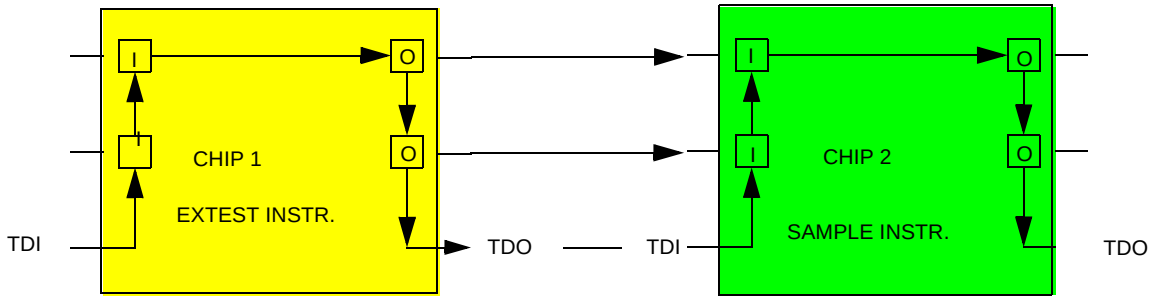

Figure 9-3 Sample EXTEST Connection

The following steps show an example of how the EXTEST instruction is initialized and invoked for board interconnection test.

1. Shift in the PRELOAD instruction in chip 1.
2. Shift in data to the boundary scan cells of chip 1 through the TDI (preload).
3. Shift the EXTEST instruction into chip 1. As soon as the Update-IR state is reached, the data in the boundary scan cells of chip 1 will be driven immediately from chip 1 into its external connections.
4. Shift the SAMPLE instruction into chip 2 to capture the logic level on the input pins including those driven by the EXTEST instruction of chip 1. Then shift the boundary scan register out to TDO for discrepancy checking.

EXTEST_PULLUP asserts internal reset for the MPC509 system logic for the duration of the instruction. This forces a predictable internal state while external boundary scan operations are performed.

9.5.2 BYPASS (1111)

The BYPASS instruction enables the single-bit BYPASS register between TDI and TDO as shown in **Figure 9-4**. This creates a shift-register path from TDI to the bypass register and finally to the TDO signal, circumventing the boundary scan register. This instruction is used to enhance test efficiency by shortening the overall path between TDI and TDO when no test operation of a component is required. In this instruction, the MPC509 system logic is independent of the test access port. When this instruction is selected, the test logic shall have no effect on the operation of the on-chip system logic as required in the IEEE 1149.1-1990 specification.

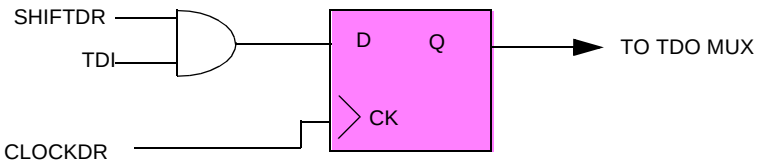

Figure 9-4 Bypass Register

9.5.3 SAMPLE/PRELOAD (1110)

The SAMPLE/PRELOAD instruction enables the boundary scan register between TDI and TDO as test data register. When this instruction is selected, the operation of the test logic shall have no effect on the operation of the on-chip system logic or on the flow of signal between the system pin and the on-chip system logic as required in the 1149.1 specification.

This instruction provides two separate functions. First, it provides a means to obtain a snapshot of system data and control signals (SAMPLE). The snapshot occurs on the rising edge of TCK in the Capture-DR controller state. The data can be observed by shifting it transparently through the boundary scan register. In a normal system configuration many signals require external pull-ups to ensure proper system operation. Consequently, the same is true for the SAMPLE/PRELOAD functionality. The data latched into the boundary scan register during the Capture-DR state may not match the drive state of the package signal if the system-required pull-ups are not present within the test environment.

The second function of the SAMPLE/PRELOAD instruction is to initialize the boundary scan register output cells (PRELOAD) prior to selection of CLAMP, EXTEST or EXTEST_PULLUP. This initialization ensures that known data will appear on the outputs when executing the EXTEST instruction. The data held in the shift register stage is transferred to the output stage on the falling edge of TCK in the Update-DR controller state.

NOTE

Since there is no internal synchronization between the IEEE 1149.1 clock (TCK) and the system clock (CLK), the user must provide some form of external synchronization to achieve meaningful results when sampling system values.

9.5.4 CLAMP (0011)

The CLAMP instruction enables the single-bit BYPASS register between TDI and TDO as test data register. It is provided as a public instruction. When the CLAMP instruction is invoked, the package output signals will respond to the preconditioned values within the update latches of the boundary scan register, even though the bypass register is enabled as the test data register.

In-circuit testing can be facilitated by setting up guarding signal conditions that control the operation of logic not involved in the test with use of the SAMPLE/PRELOAD or EXTEST instructions. Then, as the chip enters into the CLAMP instruction, the state and drive of all signals remain static until a new instruction is invoked. While the signals continue to supply the guarding inputs to the in-circuit test site, the bypass is enabled and thus should minimize overall test time.

CLAMP asserts internal reset for the MPC509 system logic for the duration of the instruction. This forces a predictable internal state while external boundary scan operations are performed.

The CLAMP instruction performs the same task as the EXTEST instruction. Unlike the EXTEST instruction, however, once the data in the boundary scan cell is updated, it remains unchanged until a new instruction is shifted in or reset.

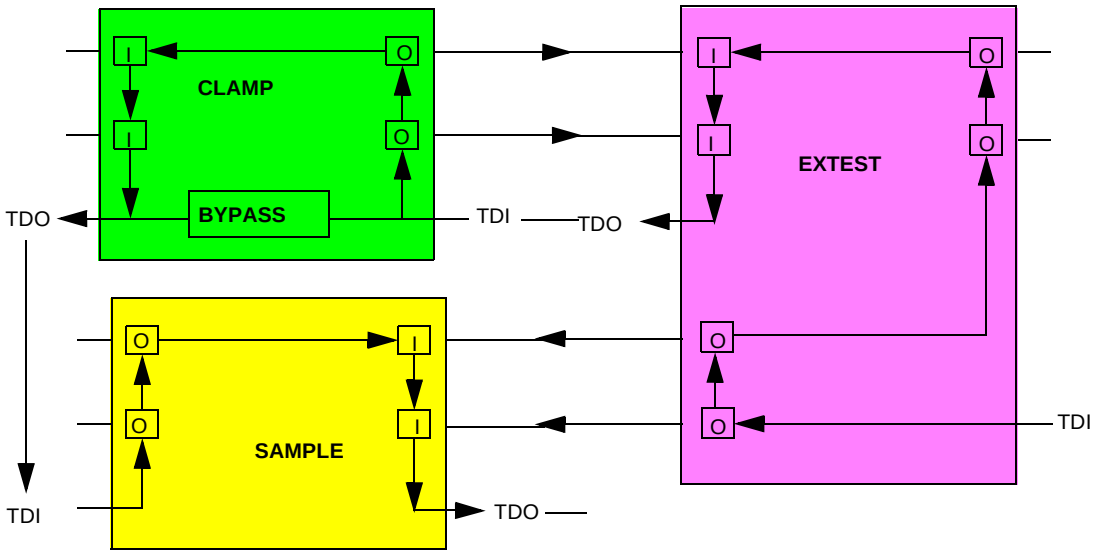

Figure 9-5 Typical Clamp Example

9.5.5 HIGHZ (0010)

The HIGHZ instruction enables the single-bit BYPASS register between TDI and TDO to function as test data register. HIGHZ is provided as a public instruction in order to avoid having to backdrive the output signals during circuit board testing. When the HIGHZ instruction is invoked, all output drivers are placed in an inactive-drive state.

HIGHZ also asserts internal reset for the MPC509 system logic for the duration of HIGHZ in order to force a predictable internal state while performing external boundary scan operations.

9.5.6 EXTEST_PULLUP (0001)

The EXTEST_PULLUP instruction is not included in the IEEE 1149.1-1990 standard. It is provided as a public instruction to aid in fault diagnosis during boundary scan testing of a circuit board. This instruction functions identically to EXTEST except for the presence of a weak pull-up device on all input signals. The MPC509 is a CMOS design and could therefore suffer from a logically indeterminate input value if an input or bi-directional signal programmed as an input was inadvertently left unconnected. The pull-up current will, given an appropriate charging delay, supply a deterministic logic 1 result on an open input.

Note that when this instruction is used in board level testing with heavily loaded nodes, it may require a charging delay greater than the two TCK periods needed to change from the Update-DR state to the Capture-DR state. Two methods of providing an increase delay are available:

- Traverse into the Run-Test/Idle state for extra TCK periods of charging delay; or
- Limit the maximum TCK frequency (slow down the TCK) so that two TCK periods are adequate

9.5.7 IDCODE (1101)

The IDCODE enables the IDREGISTER between TDI and TDO as test data register. It is provided as a public instruction to allow the manufacturer, part number, and version of a component to be determined through the TAP. **Figure 9-6** shows the IDREGISTER configuration.

Once the IDCODE instruction is decoded, it selects the IDREGISTER, a 32-bit test data register. The bypass register loads a logic 0 at the start of a scan cycle, whereas an IDREGISTER loads a constant logic 1 into its least significant bit (LSB). Examination of the first bit of data shifted out of a component during a test data scan sequence immediately following exit from the Test-Logic-Reset controller state will therefore show whether such a register is included in the design.

When the IDCODE instruction is selected, the operation of the test logic has no effect on the operation of the on-chip system logic, as required in the IEEE 1149.1-1990 specification.

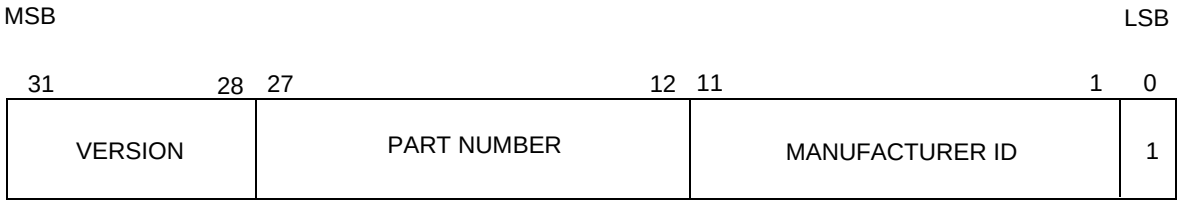

Figure 9-6 IDREGISTER Configuration

One application of the device identification register is to distinguish the manufacturer(s) of components on a board when multiple sourcing is used. As more components emerge which conform to IEEE 1149.1-1990, it is desirable to allow for a system diagnostic controller unit to blindly interrogate a board design in order to determine the type of each component in each location. This information is also available for factory process monitoring and for failure mode analysis of assembled boards.

9.5.8 TMSCAN (1100)

The TMSCAN instruction enables the 22-bit TMREG register between TDI and TDO as test data register. It is provided as a Motorola private instruction in order to serially shift in stimulus data to the on-chip test module and serially shift out test result from the test module.

9.6 Restrictions

The control afforded by the output enable signals using the boundary scan register and the EXTEST or CLAMP instructions requires a compatible circuit board test environ-

ment to avoid device-destructive configurations. The user must avoid situations in which the MPC509 output drivers are enabled into actively driven networks.

9.7 Non-IEEE 1149.1-1990 Operation

In non-IEEE 1149.1-1990 operation, the IEEE 1149.1-1990 test logic must be kept transparent to the system logic by forcing the TAP controller into the Test-Logic-Reset controller state and keeping it there. There are two methods of forcing the controller to this state. The first is to assert the TRST signal, forcing the TAP into the test-logic-reset controller state. The second is to provide at least five TCK pulses with TMS held high.

To ensure that the controller remains in the test-logic-reset state, several options are available:

- If TMS either remains unconnected or is connected to V_{CC}, then the TAP controller cannot leave the test-logic-reset state regardless of the state of the TCK pin.
- TRST can be asserted either by connecting it directly to ground or by means of a logic network.
- The controller will remain in the test-logic-reset state in the absence of a rising edge on the TCK pin regardless of the state of the TMS.

9.8 Boundary Scan Descriptor Language (BSDL)

This section provides an example of the boundary scan descriptor language (BSDL).

```
-- Motorola MPC509 Model BSDL description
-- Version 1.1 Modified 11/29 by Keeho Kang to accept pad ring fixes
for cpu rev B
   entity MPC509 is
   generic(PHYSICAL_PIN_MAP:string := “XX_Package”),

-- in          = input only
-- buffer      = two-state (0 1) output
-- out         = three-state or open drain output
-- inout       = bidirectional
-- linkage     = “other” than above (power, analog, etc.)
-- bit         = single pin
-- bit_vector = multiple pins with integer suffix (a01, a1, a2,
etc.)

   port(TDI:       in      bit;
        TDO:       out     bit;
        TMS:       in      bit;
        TCK:       in      bit;
        TRST_L:    in      bit;
        CS:        inout   bit_vector(0 to 11);
        CSBOOT_L:  buffer  bit;
        PDWU:      buffer  bit;
        CT:        inout   bit_vector(0 to 3);
        BE_L:      inout   bit_vector(0 to 3);
        BI_L:      inout   bit;
        BURST_L:   inout   bit;
```

```
        BDIP_L:   inout   bit;
        ARETRY_L: inout   bit;
        CR_L:     inout   bit;
        ECROUT:   buffer  bit;
        CLKOUT:   buffer  bit;
        SRESET_L: buffer  bit;
        RESET_L:  in      bit;
        IRQ_L:    inout   bit_vector(0 to 6);
        DSCK:     in      bit;
        DSDI:     in      bit;
        MODCK:    in      bit;
        WP:       inout   bit_vector(0 to 5);
        VFLS:     inout   bit_vector(0 to 1);
        VF:       inout   bit_vector(0 to 2);
        PLLL:     inout   bit;
        AT:       inout   bit_vector(0 to 1);
        A:        inout   bit_vector(12 to 29);
        D:        inout   bit_vector(0 to 31);
        BB_L:     inout   bit;
        BG_L:     inout   bit;
        BR_L:     inout   bit;
        TEA_L:    inout   bit;
        TA_L:     inout   bit;
        AACK_L:   inout   bit;
        TS_L:     inout   bit;
        WR_L:     inout   bit;
        VDDKAP1:  linkage bit;
        VDDKAP2:  linkage bit;
        VSSE:     linkage bit_vector(0 to 12);
        VDDE:     linkage bit_vector(0 to 12);
        VSSIL:    linkage bit;
        VDDIL:    linkage bit;
        VSSIB:    linkage bit;
        VDDIB:    linkage bit;
        VSSIR:    linkage bit;
        VDDIR:    linkage bit;
        VSSIT:    linkage bit;
        VDDIT:    linkage bit;
        VSSSN:    linkage bit;
        VDDSN:    linkage bit;
        XFCP:     linkage bit;
        XFCN:     linkage bit;
        XTAL:     linkage bit;
        EXTAL:    linkage bit
       );


   use STD_1149_1_1990.all;

   attribute PIN_MAP of MPC509 : entity is PHYSICAL_PIN_MAP;
-- Begin package description for XX_Package
-- 160-PIN QFP  (XX Suffix)

-- package pins in same order as port list (a0, a1, a2, etc.) for
bit_vectors
```

```
   constant XX_Package : PIN_MAP_STRING :=
        "VSSE: (  1, 13, 29,  41,  60, 74, 81, 95, 107, 120, 127,
137, 153 ),
"  &
       "VDDE:  ( 160, 14, 26,  40,  61, 75, 80, 94, 106, 121, 126,
136, 152 ),
" &
       "VSSIL:    19,  "  &
       "VDDIL:    20,  "  &
       "VSSIB:    66,  "  &
       "VDDIB:    67,  "  &
       "VDDIR:   100,  "  &
       "VSSIR:   101,  "  &
       "VDDIT:   142,  "  &
       "VSSIT:   143,  "  &
       "CS:     ( 7,  6,  5,  4,  3,  2,  159, 158, 157, 156, 155,
        154),   " &
       "CSBOOT_L:    8, "  &
       "CT:     ( 12, 11, 10,  9 ),  "  &
       "BE_L:   ( 18, 17, 16, 15 ),  " &
       "CR_L:     25, "  &
       "ARETRY_L: 24, " &
       "BDIP_L:   23, "  &
       "BURST_L:  22, "  &
       "BI_L:     21, "  &
       "ECROUT:   27, "  &
       "CLKOUT:   28, "  &
       "PDWU:     30, "  &
       "VDDKAP1:  31, "  &
       "XTAL:     32, " &
       "EXTAL:    33, " &
       "VSSSN:    34, "  &
       "XFCN:     35, "  &
       "XFCP:     36, "  &
       "VDDSN:    37, "  &
       "SRESET_L: 38, "  &
       "RESET_L:  39, "  &
       "IRQ_L:  ( 43, 42, 53, 54, 55, 56, 57), "  &
       "DSCK:     44, "  &
       "DSDI:     45, "  &
       "VDDKAP2:  46, " &
       "MODCK:    47, "  &
       "TRST_L:   48,  "  &
       "TMS:      49,  "  &
       "TCK:      50,  "  &
       "TDO:      51,  "  &
       "TDI:      52,  "  &
       "WP:     (65, 64, 63, 62, 59, 58),  "  &
       "VFLS:   ( 69, 68 ),  "  &
       "VF:     ( 72, 71, 70 ), " &
       "PLLL:     73,  "  &
       "AT:     ( 77, 76 ),  " &
       "A:      ( 78, 79, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91,
        92, 93, 96, 97, 98, 99 ),  " &
       "D: (102, 103, 104, 105, 108, 109, 110, 111, 112, 113,114,
```

```
          115, 116, 117,  "  &
         "         118, 119, 122, 123, 124, 125, 128, 129, 130, 131,
          132, 133, 134, 135,  "  &
         "          138, 139, 140, 141 ),  "  &
         "BB_L:     144,  "  &
         "BG_L:     145,  "  &
         "BR_L:     146,  "  &
         "TEA_L:    147,  "  &
         "TA_L:     148,  "  &
         "AACK_L:   149,  "  &
         "TS_L:     150,  "  &
         "WR_L:     151   " ;

-- Other Pin Maps here when documented

   attribute TAP_SCAN_IN of TDI:signal is true;
   attribute TAP_SCAN_OUT of TDO:signal is true;
   attribute TAP_SCAN_MODE of TMS:signal is true;
   attribute TAP_SCAN_RESET of TRST_L:signal is true;
   attribute TAP_SCAN_CLOCK of TCK:signal is (10.0e6, BOTH);

   attribute INSTRUCTION_LENGTH of MPC509:entity is 4;

   attribute INSTRUCTION_OPCODE of MPC509:entity is
     "EXTEST         (0000),"  &
     "EXTEST_PULLUP  (0001),"  &
     "HIGHZ          (0010),"  &
     "CLAMP          (0011),"  &
     "TMSCAN         (1100),"  &
     "IDCODE         (1101),"  &
     "SAMPLE         (1110),"  &
     "BYPASS         (1111) ";

   attribute INSTRUCTION_CAPTURE of MPC509:entity is "0001";
   attribute INSTRUCTION_PRIVATE of MPC509:entity is "TMSCAN";

   attribute IDCODE_REGISTER of MPC509:entity is
     "0000" & -- version
     "000010" & -- design center
     "0000000000" & -- sequence number
     "00000001110" & -- motorola
     "1";          -- required by 1149.1

   attribute REGISTER_ACCESS of MPC509:entity is
     "BOUNDARY (EXTEST_PULLUP)  ";

   attribute BOUNDARY_CELLS of MPC509:entity is
     "BC_2, BC_4, BC_6";

   attribute BOUNDARY_LENGTH of MPC509:entity is 217;

   attribute BOUNDARY_REGISTER of MPC509:entity is

-- PORT DESCRIPTION DECODE
-- port = port name or port name with suffix such as a(1)
```

```
-- cell = BC_4 for inputs, BC_6 for bidirectionals, BC_2 all other
-- function =
--    input    = input only
--    bidir    = directional
--    controlr = control with jtag_reset
--    output2  = output two state (0 1)
-- safe =
--    X for input, output2, bidir
--    0 for control cells (0 = input)
-- ccell = controlling cell
-- dsval = disable value = 0 (0 = input)
-- rslt =  result of putting dsval in control cell = Z

   --num  cell   port      function  safe  ccell dsval rslt
   --tdo = first bit to be shifted out during ShiftDR

     "0   (BC_4, MODCK,    input,    X), "    &
     "1   (BC_4, DSDI,     input,    X), "    &
     "2   (BC_4, DSCK,     input,    X), "    &
     "3   (BC_6, IRQ_L(0), bidir,    X,       4,    0,   Z), "    &
     "4   (BC_2, *,        controlr, 0), "   &
     "5   (BC_6, IRQ_L(1), bidir,    X,       6,    0,   Z), "    &
     "6   (BC_2, *,        controlr, 0), "    &
     "7   (BC_4, RESET_L,  input,    X), "    &
     "8   (BC_2, SRESET_L, output2,  X), "    &
     "9   (BC_2, PDWU,     output2,  X), "    &
     "10  (BC_2, CLKOUT,   output2,  X), "    &
     "11  (BC_2, ECROUT,   output2,  X), "    &
     "12  (BC_6, CR_L,     bidir,    X,      13,    0,   Z), "    &
     "13  (BC_2, *,        controlr, 0), "    &
     "14  (BC_6, ARETRY_L, bidir,    X,      15,    0,   Z), "    &
     "15  (BC_2, *,        controlr, 0), "    &
     "16  (BC_6, BDIP_L,   bidir,    X,      17,    0,   Z), "    &
     "17  (BC_2, *,        controlr, 0), "    &
     "18  (BC_6, BURST_L,  bidir,    X,      19,    0,   Z), "    &
     "19  (BC_2, *,        controlr, 0), "    &
   --num  cell   port      function  safe  ccell dsval rslt
     "20  (BC_6, BI_L,     bidir,    X,      21,    0,   Z), "    &
     "21  (BC_2, *,        controlr, 0), "    &
     "22  (BC_6, BE_L(0),  bidir,    X,      23,    0,   Z), "    &
     "23  (BC_2, *,        controlr, 0), "    &
     "24  (BC_6, BE_L(1),  bidir,    X,      25,    0,   Z), "    &
     "25  (BC_2, *,        controlr, 0), "    &
     "26  (BC_6, BE_L(2),  bidir,    X,      27,    0,   Z), "    &
     "27  (BC_2, *,        controlr, 0), "    &
     "28  (BC_6, BE_L(3),  bidir,    X,      29,    0,   Z), "    &
     "29  (BC_2, *,        controlr, 0), "    &
     "30  (BC_6, CT(0),    bidir,    X,      31,    0,   Z), "    &
     "31  (BC_2, *,        controlr, 0), "     &
     "32  (BC_6, CT(1),    bidir,    X,      33,    0,   Z), "    &
     "33  (BC_2, *,        controlr, 0), "    &
     "34  (BC_6, CT(2),    bidir,    X,      35,    0,   Z), "    &
     "35  (BC_2, *,        controlr, 0), "    &
     "36  (BC_6, CT(3),    bidir,    X,      37,    0,   Z), "    &
     "37  (BC_2, *,        controlr, 0), "    &
```

```
  "38  (BC_2, CSBOOT_L,   output2,  X), "    &
  "39  (BC_6, CS(0),    bidir,    X,      40,    0,   Z), "    &
--num  cell   port      function  safe  ccell dsval rslt
  "40  (BC_2, *,         controlr, 0), "    &
  "41  (BC_6, CS(1),    bidir,    X,      42,    0,   Z), "    &
  "42  (BC_2, *,         controlr, 0), "    &
  "43  (BC_6, CS(2),    bidir,    X,      44,    0,   Z), "    &
  "44  (BC_2, *,         controlr, 0), "    &
  "45  (BC_6, CS(3),    bidir,    X,      46,    0,   Z), "    &
  "46  (BC_2, *,         controlr, 0), "    &
  "47  (BC_6, CS(4),    bidir,    X,      48,    0,   Z), "    &
  "48  (BC_2, *,         controlr, 0), "    &
  "49  (BC_6, CS(5),    bidir,    X,      50,    0,   Z), "    &
  "50  (BC_2, *,         controlr, 0), "    &
  "51  (BC_6, CS(6),    bidir,    X,      52,    0,   Z), "    &
  "52  (BC_2, *,         controlr, 0), "    &
  "53  (BC_6, CS(7),    bidir,    X,      54,    0,   Z), "    &
  "54  (BC_2, *,         controlr, 0), "    &
  "55  (BC_6, CS(8),    bidir,    X,      56,    0,   Z), "    &
  "56  (BC_2, *,         controlr, 0), "    &
  "57  (BC_6, CS(9),    bidir,    X,      58,    0,   Z), "    &
  "58  (BC_2, *,         controlr, 0), "    &
  "59  (BC_6, CS(10),   bidir,    X,      60,    0,   Z), "    &
--num  cell   port      function  safe  ccell dsval rslt
  "60  (BC_2, *,         controlr, 0), "    &
  "61  (BC_6, CS(11),   bidir,    X,      62,    0,   Z), "    &
  "62  (BC_2, *,         controlr, 0), "    &
  "63  (BC_6, WR_L,     bidir,    X,      64,    0,   Z), "    &
  "64  (BC_2, *,         controlr, 0), "    &
  "65  (BC_6, TS_L,     bidir,    X,      66,    0,   Z), "    &
  "66  (BC_2, *,         controlr, 0), "    &
  "67  (BC_6, AACK_L,   bidir,    X,      68,    0,   Z), "    &
  "68  (BC_2, *,         controlr, 0), "    &
  "69  (BC_6, TA_L,     bidir,    X,      70,    0,   Z), "    &
  "70  (BC_2, *,         controlr, 0), "    &
  "71  (BC_6, TEA_L,    bidir,    X,      72,    0,   Z), "    &
  "72  (BC_2, *,         controlr, 0), "    &
  "73  (BC_6, BR_L,     bidir,    X,      74,    0,   Z), "    &
  "74  (BC_2, *,         controlr, 0), "    &
  "75  (BC_6, BG_L,     bidir,    X,      76,    0,   Z), "    &
  "76  (BC_2, *,         controlr, 0), "    &
  "77  (BC_6, BB_L,     bidir,    X,      78,    0,   Z), "    &
  "78  (BC_2, *,         controlr, 0), "    &
  "79  (BC_6, D(31),    bidir,    X,      80,    0,   Z), "    &
--num  cell   port      function  safe  ccell dsval rslt
  "80  (BC_2, *,         controlr, 0), "    &
  "81  (BC_6, D(30),    bidir,    X,      82,    0,   Z), "    &
  "82  (BC_2, *,         controlr, 0), "    &
  "83  (BC_6, D(29),    bidir,    X,      84,    0,   Z), "    &
  "84  (BC_2, *,         controlr, 0), "    &
  "85  (BC_6, D(28),    bidir,    X,      86,    0,   Z), "    &
  "86  (BC_2, *,         controlr, 0), "    &
  "87  (BC_6, D(27),    bidir,    X,      88,    0,   Z), "    &
  "88  (BC_2, *,         controlr, 0), "    &
  "89  (BC_6, D(26),    bidir,    X,      90,    0,   Z), "    &
```

```
  “90  (BC_2, *,        controlr, 0), “    &
  “91  (BC_6, D(25),    bidir,    X,      92,    0,   Z), “    &
  “92  (BC_2, *,        controlr, 0), “    &
  “93  (BC_6, D(24),    bidir,    X,      94,    0,   Z), “    &
  “94  (BC_2, *,        controlr, 0), “    &
  “95  (BC_6, D(23),    bidir,    X,      96,    0,   Z), “    &
  “96  (BC_2, *,        controlr, 0), “    &
  “97  (BC_6, D(22),    bidir,    X,      98,    0,   Z), “    &
  “98  (BC_2, *,        controlr, 0), “    &
  “99  (BC_6, D(21),    bidir,    X,     100,    0,   Z), “    &
--num  cell   port      function  safe  ccell dsval rslt
  “100 (BC_2, *,        controlr, 0), “    &
  “101 (BC_6, D(20),    bidir,    X,     102,    0,   Z), “    &
  “102 (BC_2, *,        controlr, 0), “    &
 “103 (BC_6, D(19),    bidir,    X,     104,    0,   Z), “    &
  “104 (BC_2, *,        controlr, 0), “    &
  “105 (BC_6, D(18),    bidir,    X,     106,    0,   Z), “    &
  “106 (BC_2, *,        controlr, 0), “    &
  “107 (BC_6, D(17),    bidir,    X,     108,    0,   Z), “    &
  “108 (BC_2, *,        controlr, 0), “    &
  “109 (BC_6, D(16),    bidir,    X,     110,    0,   Z), “    &
  “110 (BC_2, *,        controlr, 0), “    &
  “111 (BC_6, D(15),    bidir,    X,     112,    0,   Z), “    &
  “112 (BC_2, *,        controlr, 0), “    &
  “113 (BC_6, D(14),    bidir,    X,     114,    0,   Z), “    &
  “114 (BC_2, *,        controlr, 0), “    &
  “115 (BC_6, D(13),    bidir,    X,     116,    0,   Z), “    &
  “116 (BC_2, *,        controlr, 0), “    &
  “117 (BC_6, D(12),    bidir,    X,     118,    0,   Z), “    &
  “118 (BC_2, *,        controlr, 0), “    &
  “119 (BC_6, D(11),    bidir,    X,     120,    0,   Z), “    &
--num  cell   port      function  safe  ccell dsval rslt
  “120 (BC_2, *,        controlr, 0), “    &
  “121 (BC_6, D(10),    bidir,    X,     122,    0,   Z), “    &
  “122 (BC_2, *,        controlr, 0), “    &
 “123 (BC_6, D(9),     bidir,    X,     124,    0,   Z), “    &
  “124 (BC_2, *,        controlr, 0), “    &
  “125 (BC_6, D(8),     bidir,    X,     126,    0,   Z), “    &
  “126 (BC_2, *,        controlr, 0), “    &
  “127 (BC_6, D(7),     bidir,    X,     128,    0,   Z), “    &
  “128 (BC_2, *,        controlr, 0), “    &
  “129 (BC_6, D(6),     bidir,    X,     130,    0,   Z), “    &
  “130 (BC_2, *,        controlr, 0), “    &
  “131 (BC_6, D(5),     bidir,    X,     132,    0,   Z), “    &
  “132 (BC_2, *,        controlr, 0), “    &
  “133 (BC_6, D(4),     bidir,    X,     134,    0,   Z), “    &
  “134 (BC_2, *,        controlr, 0), “    &
  “135 (BC_6, D(3),     bidir,    X,     136,    0,   Z), “    &
  “136 (BC_2, *,        controlr, 0), “    &
  “137 (BC_6, D(2),     bidir,    X,     138,    0,   Z), “    &
  “138 (BC_2, *,        controlr, 0), “    &
  “139 (BC_6, D(1),     bidir,    X,     140,    0,   Z), “    &
--num  cell   port      function  safe  ccell dsval rslt
  “140 (BC_2, *,        controlr, 0), “    &
  “141 (BC_6, D(0),     bidir,    X,     142,    0,   Z), “    &
```

```
 “142 (BC_2, *,        controlr, 0), “    &
“143 (BC_6, A(29),   bidir,   X,    144,   0,   Z), “   &
 “144 (BC_2, *,        controlr, 0), “    &
 “145 (BC_6, A(28),   bidir,   X,    146,   0,   Z), “   &
 “146 (BC_2, *,        controlr, 0), “    &
 “147 (BC_6, A(27),   bidir,   X,    148,   0,   Z), “   &
 “148 (BC_2, *,        controlr, 0), “    &
 “149 (BC_6, A(26),   bidir,   X,    150,   0,   Z), “   &
 “150 (BC_2, *,        controlr, 0), “    &
 “151 (BC_6, A(25),   bidir,   X,    152,   0,   Z), “   &
 “152 (BC_2, *,        controlr, 0), “    &
 “153 (BC_6, A(24),   bidir,   X,    154,   0,   Z), “   &
 “154 (BC_2, *,        controlr, 0), “    &
 “155 (BC_6, A(23),   bidir,   X,    156,   0,   Z), “   &
 “156 (BC_2, *,        controlr, 0), “    &
 “157 (BC_6, A(22),   bidir,   X,    158,   0,   Z), “   &
 “158 (BC_2, *,        controlr, 0), “    &
 “159 (BC_6, A(21),   bidir,   X,    160,   0,   Z), “   &
--num  cell   port     function  safe  ccell dsval rslt
 “160 (BC_2, *,        controlr, 0), “    &
 “161 (BC_6, A(20),   bidir,   X,    162,   0,   Z), “   &
 “162 (BC_2, *,        controlr, 0), “    &
“163 (BC_6, A(19),   bidir,   X,    164,   0,   Z), “   &
 “164 (BC_2, *,        controlr, 0), “    &
 “165 (BC_6, A(18),   bidir,   X,    166,   0,   Z), “   &
 “166 (BC_2, *,        controlr, 0), “    &
 “167 (BC_6, A(17),   bidir,   X,    168,   0,   Z), “   &
 “168 (BC_2, *,        controlr, 0), “    &
 “169 (BC_6, A(16),   bidir,   X,    170,   0,   Z), “   &
 “170 (BC_2, *,        controlr, 0), “    &
 “171 (BC_6, A(15),   bidir,   X,    172,   0,   Z), “   &
 “172 (BC_2, *,        controlr, 0), “    &
 “173 (BC_6, A(14),   bidir,   X,    174,   0,   Z), “   &
 “174 (BC_2, *,        controlr, 0), “    &
 “175 (BC_6, A(13),   bidir,   X,    176,   0,   Z), “   &
 “176 (BC_2, *,        controlr, 0), “    &
 “177 (BC_6, A(12),   bidir,   X,    178,   0,   Z), “   &
 “178 (BC_2, *,        controlr, 0), “    &
 “179 (BC_6, AT(0),   bidir,   X,    180,   0,   Z), “   &
--num  cell   port     function  safe  ccell dsval rslt
 “180 (BC_2, *,        controlr, 0), “    &
 “181 (BC_6, AT(1),   bidir,   X,    182,   0,   Z), “   &
 “182 (BC_2, *,        controlr, 0), “    &
“183 (BC_6, PLLL,    bidir,   X,    184,   0,   Z), “   &
 “184 (BC_2, *,        controlr, 0), “    &
 “185 (BC_6, VF(0),   bidir,   X,    186,   0,   Z), “   &
 “186 (BC_2, *,        controlr, 0), “    &
 “187 (BC_6, VF(1),   bidir,   X,    188,   0,   Z), “   &
 “188 (BC_2, *,        controlr, 0), “    &
 “189 (BC_6, VF(2),   bidir,   X,    190,   0,   Z), “   &
 “190 (BC_2, *,        controlr, 0), “    &
 “191 (BC_6, VFLS(0), bidir,   X,    192,   0,   Z), “   &
 “192 (BC_2, *,        controlr, 0), “    &
 “193 (BC_6, VFLS(1), bidir,   X,    194,   0,   Z), “   &
 “194 (BC_2, *,        controlr, 0), “    &
```

```
  “195 (BC_6, WP(0),    bidir,    X,     196,    0,   Z), “    &
  “196 (BC_2, *,         controlr, 0), “    &
  “197 (BC_6, WP(1),    bidir,    X,     198,    0,   Z), “    &
  “198 (BC_2, *,         controlr, 0), “    &
  “199 (BC_6, WP(2),    bidir,    X,     200,    0,   Z), “    &
--num  cell   port       function  safe  ccell dsval rslt
  “200 (BC_2, *,         controlr, 0), “    &
  “201 (BC_6, WP(3),    bidir,    X,     202,    0,   Z), “    &
  “202 (BC_2, *,         controlr, 0), “    &
 “203 (BC_6, WP(4),    bidir,    X,     204,    0,   Z), “    &
  “204 (BC_2, *,         controlr, 0), “    &
  “205 (BC_6, WP(5),    bidir,    X,     206,    0,   Z), “    &
  “206 (BC_2, *,         controlr, 0), “    &
  “207 (BC_6, IRQ_L(2), bidir,    X,     208,    0,   Z), “    &
  “208 (BC_2, *,         controlr, 0), “    &
  “209 (BC_6, IRQ_L(3), bidir,    X,     210,    0,   Z), “    &
  “210 (BC_2, *,         controlr, 0), “    &
  “211 (BC_6, IRQ_L(4), bidir,    X,     212,    0,   Z), “    &
  “212 (BC_2, *,         controlr, 0), “    &
  “213 (BC_6, IRQ_L(5), bidir,    X,     214,    0,   Z), “    &
  “214 (BC_2, *,         controlr, 0), “    &
  “215 (BC_6, IRQ_L(6), bidir,    X,     216,    0,   Z), “    &
  “216 (BC_2, *,         controlr, 0)  “    ;
-- tdi
end MPC509;
```