# JTAG Interface

# 14

The R4000 processor provides a boundary-scan interface that is compatible with Joint Test Action Group (JTAG) specifications, using the industry-standard JTAG protocol.

This chapter describes that interface, including descriptions of boundary scanning, the pins and signals used by the interface, and the Test Access Port (TAP).

## 14.1 What Boundary Scanning Is

With the evolution of ever-denser integrated circuits (ICs), surface-mounted devices, double-sided component mounting on printed-circuit boards (PCBs), and buried vias, in-circuit tests that depend upon making physical contact with internal board and chip connections have become more and more difficult to use. The greater complexity of ICs has also meant that tests to fully exercise these chips have become much larger and more difficult to write.

One solution to this difficulty has been the development of *boundary-scan* circuits. A boundary-scan circuit is a series of shift register cells placed between each pin and the internal circuitry of the IC to which the pin is connected, as shown in Figure 14-1. Normally, these boundary-scan cells are bypassed; when the IC enters test mode, however, the scan cells can be directed by the test program to pass data along the shift register path and perform various diagnostic tests. To accomplish this, the tests use the four signals described in the next section: **JTDI, JTDO**, **JTMS**, and **JTCK**.

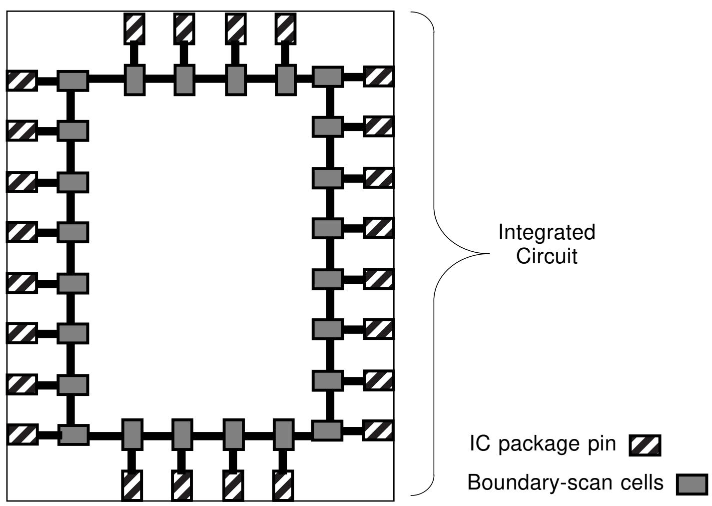


*Figure 14-1 JTAG Boundary-scan Cells*

## 14.2 Signal Summary

The JTAG interface signals are listed below and shown in Figure 14-2.

| | |
|---|---|
| **JTDI** | JTAG serial data in |
| **JTDO** | JTAG serial data out |
| **JTMS** | JTAG test mode select |
| **JTCK** | JTAG serial clock input |

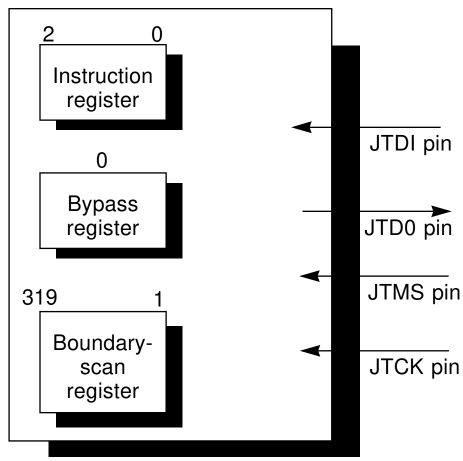


*Figure 14-2 JTAG Interface Signals and Registers*

The JTAG boundary-scan mechanism (referred to in this chapter as *JTAG mechanism*) allows testing of the connections between the processor, the printed circuit board to which it is attached, and the other components on the circuit board.

In addition, the JTAG mechanism provides rudimentary capability for low-speed logical testing of the secondary cache RAM. The JTAG mechanism does not provide any capability for testing the processor itself.

## 14.3 JTAG Controller and Registers

The processor contains the following JTAG controller and registers:

- *Instruction* register
- *Boundary-scan* register
- *Bypass* register
- Test Access Port (TAP) controller

The processor executes the standard JTAG EXTEST operation associated with External Test functionality testing.

### Instruction Register

The JTAG *Instruction* register includes three shift register-based cells; this register is used to select the test to be performed and/or the test data register to be accessed. As listed in Table 14-1, this encoding selects either the *Boundary-scan* register or the *Bypass* register.

*Table 14-1 JTAG Instruction Register Bit Encoding*

| MSB. . . . . LSB | Data Register |
|---|---|
| 0 0 0 | Boundary-scan register (external test only) |
| x x 1 | Bypass register |
| x 1 x | Bypass register |
| 1 x x | Bypass register |

The *Instruction* register has two stages:

- shift register
- parallel output latch

Figure 14-3 shows the format of the *Instruction* register.

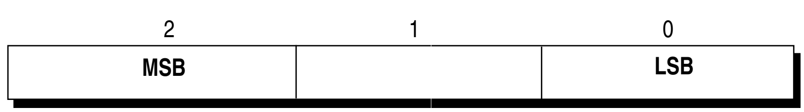


*Figure 14-3 Instruction Register*

## Bypass Register

The *Bypass* register is 1 bit wide. When the TAP controller is in the Shift-DR (Bypass) state, the data on the **JTDI** pin is shifted into the *Bypass* register, and the *Bypass* register output shifts to the **JTDO** output pin.

In essence, the *Bypass* register is a short-circuit which allows bypassing of board-level devices, in the serial boundary-scan chain, which are not required for a specific test. The logical location of the *Bypass* register in the boundary-scan chain is shown in Figure 14-4. Use of the *Bypass* register speeds up access to boundary-scan registers in those ICs that remain active in the board-level test datapath.

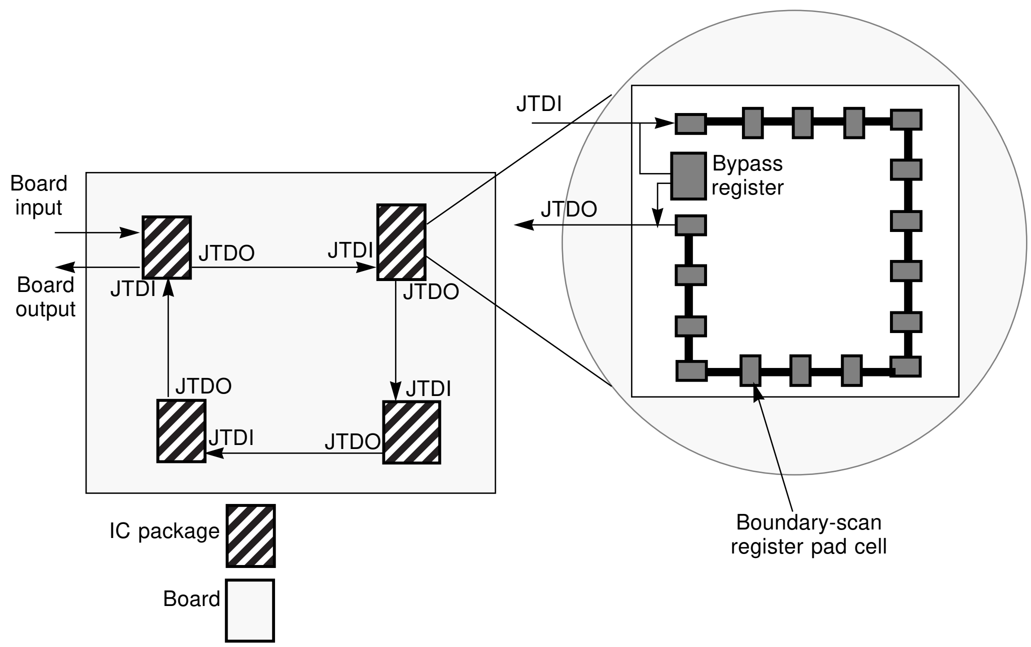


*Figure 14-4 Bypass Register Operation*

## Boundary-Scan Register

The *Boundary-scan* register is a single, 319-bit-wide, shift register-based path containing cells connected to all input and output pads on the R4000 processor. Figure 14-5 shows the three most-significant bits of the *Boundary-scan* register; these three bits control the output enables on the various bidirectional buses.

| 319 | 318 | 317 | 316 … 1 |
|---|---|---|---|
| **OE3** | **OE2** | **OE1** | See Table 14-2 |

*Figure 14-5 Output Enable Bits of the Boundary-scan Register*

The most-significant bit, **OE3** (bit 319), is the JTAG output enable bit for the **SysAD**, **SysADC**, **SysCmd**, and **SysCmdP** buses. Output is enabled when this bit is set to 1.

**OE2** (bit 318) is the JTAG output enable for the **SCData** and **SCDChk** buses. Output is enabled when this bit is set to 1.

**OE1** (bit 317) is the JTAG output enable for the **SCTag** and **SCTChk** buses.

The remaining 316 bits correspond to 316 signal pads of the processor. Output is enabled when this bit is set to 1.

At the end of this chapter, Table 14-2 lists the scan order of these 316 scan bits, starting from **JTDI** and ending with **JTDO**.

## Test Access Port (TAP)

The Test Access Port (TAP) consists of the four signal pins: **JTDI**, **JTDO**, **JTMS**, and **JTCK**. Serial test data and instructions are communicated over these four signal pins, along with control of the test to be executed.

As Figure 14-6 shows, data is serially scanned into one of the three registers (*Instruction* register, *Bypass* register, or the *Boundary-scan* register) from the **JTDI** pin, or it is scanned from one of these three registers onto the **JTDO** pin.

The **JTDI** input feeds the least-significant bit (LSB) of the selected register, whereas the most-significant bit (MSB) of the selected register appears on the **JTDO** output.

The **JTMS** input controls the state transitions of the main TAP controller state machine.

The **JTCK** input is a dedicated test clock that allows serial JTAG data to be shifted synchronously, independent of any chip-specific or system clocks.

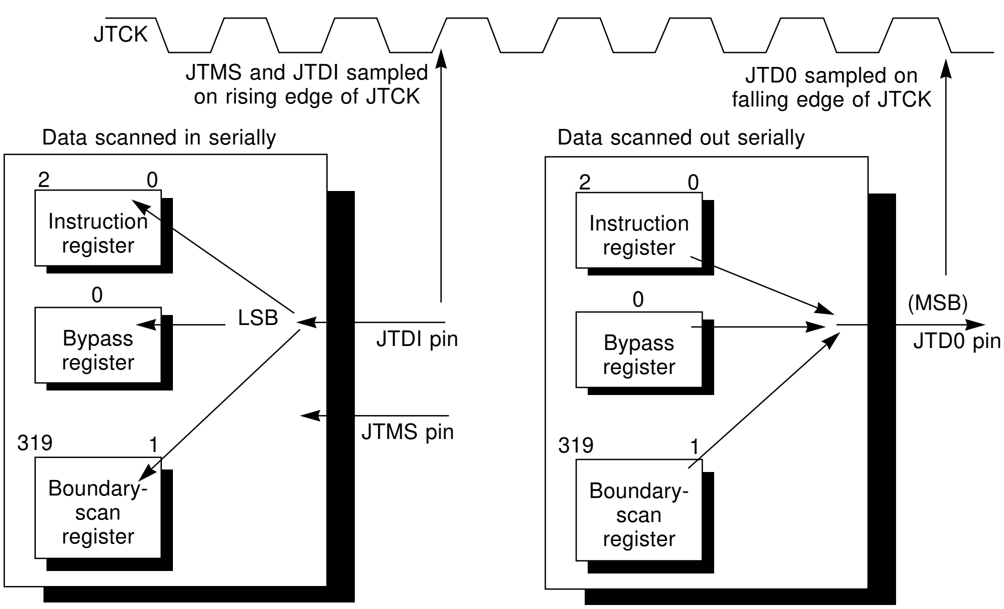


*Figure 14-6 JTAG Test Access Port*

Data on the **JTDI** and **JTMS** pins is sampled on the rising edge of the **JTCK** input clock signal. Data on the **JTDO** pin changes on the falling edge of the **JTCK** clock signal.

### TAP Controller

The processor implements the 16-state TAP controller as defined in the IEEE JTAG specification.

### Controller Reset

The TAP controller state machine can be put into Reset state by one of the following:

- deassertion of the **VCCOk** input resets the TAP controller
- keeping the **JTMS** input signal asserted through five consecutive rising edges of **JTCK** input sends the TAP controller state machine into its Reset state.

In either case, keeping **JTMS** asserted maintains the Reset state.

### Controller States

The TAP controller has four states: Reset, Capture, Shift, and Update. They can reflect either instructions (as in the Shift-IR state) or data (as in the Capture-DR state).

- When the TAP controller is in the Reset state, the value 0x7 is loaded into the parallel output latch, selecting the *Bypass* register as default. The three most significant bits of the *Boundary-scan* register are cleared to 0, disabling the outputs.
- When the TAP controller is in the Capture-IR state, the value 0x4 is loaded into the shift register stage.
- When the TAP controller is in the Capture-DR (Boundary-scan) state, the data currently on the processor input and I/O pins is latched into the *Boundary-scan* register. In this state, the *Boundary-scan* register bits corresponding to output pins are arbitrary and cannot be checked during the scan out process.
- When the TAP controller is in the Shift-IR state, data is loaded serially into the shift register stage of the *Instruction* register from the **JTDI** input pin, and the MSB of the *Instruction* register's shift register stage is shifted onto the **JTDO** pin.

- When the TAP controller is in the Shift-DR (Boundary-scan) state, data is serially shifted into the *Boundary-scan* register from the **JTDI** pin, and the contents of the *Boundary-scan* register are serially shifted onto the **JTDO** pin.
- When the TAP controller is in the Update-IR state, the current data in the shift register stage is loaded into the parallel output latch.
- When the TAP controller is in the Update-DR (Boundary-scan) state, data in the *Boundary-scan* register is latched into the register parallel output latch. Bits corresponding to output pins, and those I/O pins whose outputs are enabled (by the three MSBs of the *Boundary-scan* register), are loaded onto the processor pins.

Table 14-2 shows the boundary scan order of the processor signals.

*Table 14-2 JTAG Scan Order of R4000 Processor Pins*

| Pin # | Signal Name | Pin # | Signal Name | Pin # | Signal Name | Pin # | Signal Name |
|---|---|---|---|---|---|---|---|
| 1. | SCDChk(13) | 2. | SysADC(1) | 3. | SCDChk(1) | 4. | SysADC(5) |
| 5. | SCDChk(5) | 6. | Status(0) | 7. | Status(1) | 8. | Status(2) |
| 9. | Status(3) | 10. | IvdErr* | 11. | Status(4) | 12. | IvdAck* |
| 13. | Status(5) | 14. | Status(6) | 15. | Status(7) | 16. | SCDChk(7) |
| 17. | SysADC(7) | 18. | SCDChk(3) | 19. | SysADC(3) | 20. | SCDChk(15) |
| 21. | VCCOk | 22. | SCTag(16) | 23. | SCDChk(11) | 24. | SCData(63) |
| 25. | SysAD(63) | 26. | SCData(31) | 27. | SysAD(31) | 28. | SCData(127) |
| 29. | SCTag(17) | 30. | SCData(95) | 31. | SCData(62) | 32. | SysAD(62) |
| 33. | SCData(30) | 34. | SysAD(30) | 35. | SCData(126) | 36. | SCTag(18) |
| 37. | SCData(94) | 38. | RClock(1:0) (share the same JTAG bit) | 39. | SCTag(19) | 40. | SCData(61) |
| 41. | SysAD(61) | 42. | SCData(29) | 43. | SysAD(29) | 44. | SCData(125) |
| 45. | Reset* | 46. | SCTag(20) | 47. | SCData(93) | 48. | SCData(60) |
| 49. | SysAD(60) | 50. | SCData(28) | 51. | SysAD(28) | 52. | SCData(124) |
| 53. | ColdReset* | 54. | SCTag(21) | 55. | SCData(92) | 56. | SCData(59) |
| 57. | SysAD(59) | 58. | SCData(27) | 59. | SysAD(27) | 60. | SCData(123) |
| 61. | IOIn | 62. | SCTag(22) | 63. | SCData(91) | 64. | SCData(58) |
| 65. | SysAD(58) | 66. | SCData(26) | 67. | SysAD(26) | 68. | SCData(122) |
| 69. | IOOut | 70. | SCTag(23) | 71. | SCData(90) | 72. | SCData(57) |
| 73. | SysAD(57) | 74. | SCData(25) | 75. | SysAD(25) | 76. | SCData(121) |
| 77. | GrpRun* | 78. | SCTag(24) | 79. | SCData(89) | 80. | SCData(56) |
| 81. | SysAD(56) | 82. | SCData(24) | 83. | SysAD(24) | 84. | SCData(120) |
| 85. | GrpStall* | 86. | SCTChk(0) | 87. | SCData(88) | 88. | SCDChk(6) |
| 89. | SysADC(6) | 90. | SCDChk(2) | 91. | SysADC(2) | 92. | SCDChk(14) |
| 93. | NMI* | 94. | SCTChk(1) | 95. | SCDChk(10) | 96. | SCData(55) |

*Table 14-2 (cont.) JTAG Scan Order of R4000 Processor Pins*

| Pin # Signal Name | Pin # Signal Name | Pin # Signal Name | Pin # Signal Name |
|---|---|---|---|
| 97. SysAD(55) | 98. SCData(23) | 99. SysAD(23) | 100. SCData(119) |
| 101. Release* | 102. SCTChk(2) | 103. SCData(87) | 104. SCData(54) |
| 105. SysAD(54) | 106. SysAD(22) | 107. ModeIn | 108. SCData(22) |
| 109. RdRdy* | 110. SCData(118) | 111. SCData(86) | 112. SCData(53) |
| 113. SysAD(53) | 114. SCData(21) | 115. SysAD(21) | 116. SCData(117) |
| 117. ExtRqst* | 118. SCTChk(3) | 119. SCData(85) | 120. SCData(52) |
| 121. SysAD(52) | 122. SCData(20) | 123. SysAD(20) | 124. SCData(116) |
| 125. ValidOut* | 126. SCTChk(4) | 127. SCData(84) | 128. SCData(51) |
| 129. SysAD(51) | 130. SCData(19) | 131. SysAD(19) | 132. SCData(115) |
| 133. ValidIn* | 134. SCTChk(5) | 135. SCData(83) | 136. SCAddr0W,X (share the same JTAG bit) |
| 137. SCAddr0Y,Z (share the same JTAG bit) | 138. SCAddr(1) | 139. SCData(50) | 140. SysAD(50) |
| 141. SCData(18) | 142. SysAD(18) | 143. SCData(114) | 144. Int*(0) |
| 145. SCTChk(6) | 146. SCData(82) | 147. SCData(49) | 148. SysAD(49) |
| 149. SCData(17) | 150. SysAD(17) | 151. SCData(113) | 152. SCAddr(2)/Int*(1) |
| 153. SCAddr(3) | 154. SCData(81) | 155. SCData(48) | 156. SysAD(48) |
| 157. SCData(16) | 158. SysAD(16) | 159. SCData(112) | 160. SCAddr(4)/Int*(2) |
| 161. SCAddr(5) | 162. SCData(80) | 163. SCAddr(6) | 164. SCAddr(7) |
| 165. SCAddr(8) | 166. SCAddr(9) | 167. SCAddr(10) | 168. SCAddr(11) |
| 169. SC64Addr | 170. SCAddr(12) | 171. SCAddr(13) | 172. SCAddr(14) |
| 173. SCAddr(15) | 174. SCAddr(16) | 175. SCAddr(17) | 176. SCData(64) |
| 177. SCAPar(0) | 178. SCAPar(1)/Int*(3) | 179. SCData(96) | 180. SysAD(0) |
| 181. SCData(0) | 182. SysAD(32) | 183. SCData(32) | 184. SCData(65) |
| 185. SCAPar(2) | 186. SCOE*/Int*(4) | 187. SCData(97) | 188. SysAD(1) |
| 189. SCData(1) | 190. SysAD(33) | 191. SCData(33) | 192. SCData(66) |
| 193. SCDCS* | 194. SCTCS*/Int*(5) | 195. SCData(98) | 196. SysAD(2) |
| 197. SCData(2) | 198. SysAD(34) | 199. SCData(34) | 200. SCTag(0) |
| 201. SCWrW,X* (share the same JTAG bit) | 202. SCWrY,Z* (share the same JTAG bit) | 203. SCData(67) | 204. SCTag(1) |
| 205. SysCmd(0) | 206. SCData(99) | 207. SysAD(3) | 208. SCData(3) |
| 209. SysAD(35) | 210. SCData(35) | 211. SCData(68) | 212. SCTag(2) |
| 213. SysCmd(1) | 214. SCData(100) | 215. SysAD(4) | 216. SCData(4) |
| 217. SysAD(36) | 218. SCData(36) | 219. SCData(69) | 220. SCTag(3) |
| 221. SysCmd(2) | 222. SCData(101) | 223. SysAD(5) | 224. SCData(5) |

*Table 14-2 (cont.) JTAG Scan Order of R4000 Processor Pins*

| Pin # Signal Name | Pin # Signal Name | Pin # Signal Name | Pin # Signal Name |
|---|---|---|---|
| 225. SysAD(37) | 226. SCData(37) | 227. SCData(70) | 228. WrRdy* |
| 229. ModeClock | 230. SCData(102) | 231. SysAD(6) | 232. SCData(6) |
| 233. SysAD(38) | 234. SCData(38) | 235. SCData(71) | 236. SCTag(4) |
| 237. SysCmd(3) | 238. SCData(103) | 239. SysAD(7) | 240. SCData(7) |
| 241. SysAD(39) | 242. SCData(39) | 243. SCDChk(8) | 244. SCTag(5) |
| 245. SysCmd(4) | 246. SCDChk(12) | 247. SysADC(0) | 248. SCDChk(0) |
| 249. SysADC(4) | 250. SCDChk(4) | 251. SCData(72) | 252. SCTag(6) |
| 253. SysCmd(5) | 254. SCData(104) | 255. SysAD(8) | 256. SCData(8) |
| 257. SysAD(40) | 258. SCData(40) | 259. SCData(73) | 260. SCTag(7) |
| 261. SysCmd(6) | 262. SCData(105) | 263. SysAD(9) | 264. SCData(9) |
| 265. SysAD(41) | 266. SCData(41) | 267. SCData(74) | 268. SCTag(8) |
| 269. SysCmd(7) | 270. SCData(106) | 271. SysAD(10) | 272. SCData(10) |
| 273. SysAD(42) | 274. SCData(42) | 275. SCData(75) | 276. SCTag(9) |
| 277. SysCmd(8) | 278. SCData(107) | 279. SysAD(11) | 280. SCData(11) |
| 281. SysAD(43) | 282. SCData(43) | 283. SCData(76) | 284. SCTag(10) |
| 285. SysCmdP | 286. SCData(108) | 287. SysAD(12) | 288. SCData(12) |
| 289. SysAD(44) | 290. SCData(44) | 291. SCData(77) | 292. SCTag(11) |
| 293. Fault* | 294. SCData(109) | 295. SysAD(13) | 296. SCData(13) |
| 297. SysAD(45) | 298. SCData(45) | 299. SCTag(12) | 300. TClock(1:0) (share the same JTAG bit) |
| 301. SCData(78) | 302. SCTag(13) | 303. SCData(110) | 304. SysAD(14) |
| 305. SCData(14) | 306. SysAD(46) | 307. SCData(46) | 308. SCData(79) |
| 309. SCTag(14) | 310. SCData(111) | 311. SysAD(15) | 312. SCData(15) |
| 313. SysAD(47) | 314. SCData(47) | 315. SCDChk(9) | 316. SCTag(15)† |

†See the section titled Boundary-Scan Register earlier in this chapter, for a description of the last three output enable bits, 319:317.

## 14.4 Implementation-Specific Details

This section describes details of JTAG boundary-scan operation that are specific to the processor.

- The **MasterClock**, **MasterOut**, **SyncIn**, and **SyncOut** signal pads do not support JTAG.
- The following pairs of output pads share a single JTAG bit:

  **SCAddr0W** *and* **SCAddr0X**
  **SCAddr0Y** *and* **SCAddr0Z**
  **SCWrW*** *and* **SCWrX***
  **SCWrY*** *and* **SCWrZ***
  **TClock(0)** *and* **TClock(1)**
  **RClock(0)** *and* **RClock(1)**

- All input pads data are first latched into a processor clock-based register in the pad cell before they are captured into the Boundary-scan register in the Capture-DR (Boundary-scan) state. When the phase-locked loop is disabled, the processor clock is half the frequency of **MasterClock**. Therefore, when the TAP controller is in the Capture-DR (Boundary-scan) state, the data setup required at the input pads is more than two **MasterClock** periods before the rising edge of the **JTCK**.
- The output enable controls generated from the three most-significant bits of the *Boundary-scan* register are latched into a Processor Clock-based register before they actually enable the data onto the pads. Therefore, the delay from the rising edge of **JTCK** in the Update-DR (Boundary-scan) state to data valid at the output pins of the chip is greater than two **MasterClock** periods.