

Top-level Rules:

1.a) An Initialization Process may be required for correct device operation in Boundary-Scan testing. INIT_SETUP and/or INIT_RUN instructions shall be provided for a device that requires an orderly initialization process. The Initialization will leave the device in a controlled setup state of the input and ~~/output~~s pins for operation compatible with Boundary-Scan testing of the board under test. INIT_SETUP and INIT_RUN are both optional and independent. One instruction can exist without the other. If both instructions are present, then INIT_SETUP provides the parameters for use by INIT_RUN.

1.b) If the Initialization Process requires parameterization, the INIT_SETUP instruction is required to provide the parameters with the INIT_DATA TDR. If the INIT_RUN instruction requires externally supplied data (parameters), these data shall be supplied by the INIT_SETUP instruction.

1.c) If the Initialization Process requires a specified time between when normal operation is disturbed and before any ~~EXTTEST-type~~ instruction in 1.e can be used, then the INIT_RUN instruction is required.

~~1.d) If the INIT_RUN instruction makes use of externally supplied data (parameters) to select available options that govern the end state of its Initialization process, these data shall be supplied by an optional INIT_SETUP instruction.~~

1.d) If the Initialization Process has not been correctly completed on a device, then the only Standard instructions which can be presumed to operate correctly are the following: BYPASS, IDCODE, USERCODE, HIGHZ, PRELOAD, RUNBIST (in case of HIGHZ-type behavior), INTEST (in case of HIGHZ-type behavior).

Permission

1.e) The following instructions may be not operate correctly without Initialization Process successful completion: CLAMP, EXTEST, RUNBIST (in case of CLAMP-type behavior), INTEST (in case of CLAMP-type behavior). ~~1.e) If INIT_SETUP is provided, then INIT_RUN shall also be provided.~~

INIT_SETUP Rules:

2.a) The optional INIT_SETUP instruction shall place an INIT_DATA register between TDI and TDO. This register shall obey all rules for Test Data Registers.

2.b) The optional INIT_SETUP instruction shall not affect normal operation of the device. ~~_____~~ Does not effect either electrical or data values of the chip or the pins.

~~Note: INIT_SETUP is a normal mode instruction as is PRELOAD or BYPASS. INIT_SETUP does not effect either the electrical parameters or data state of the device and its I/O.~~

2.c) While INIT_SETUP is in effect, the INIT_DATA register shall shift data bits in from TDI on each rising edge of TCK while in the Shift-DR TAP controller state. ~~"~~

2.d) While INIT_SETUP is in effect, the INIT_DATA register shall shift data bits out onto TDO on each falling edge of TCK while in the Shift-DR TAP controller state. ~~(Note, we should decide if there is a static capture pattern associated with INIT_DATA.)~~

~~2.e) While INIT_SETUP is in effect, the INIT_DATA register content shall be updated into internal memory where it is subsequently available for use by the INIT instruction, on the falling edge of TCK while in the Update-DR TAP controller state.~~

2.~~ef~~) The values and encodings of the INIT_DATA register shall be documented in the device's system behavior description to enable board designers s to specify values correct for their board designs s.

INIT_SETUP Recommendations:

~~2.g) The INIT_SETUP TDR may capture at least a few bits of defined static capture. A leading 2 bits of "10" (1 for first bit out TDO)~~

INIT_SETUP Permissions:

2.g~~h~~) The binary value(s) for the INIT_SETUP instruction may be selected by the device designer.

2.h) The INIT_DATA register may capture a deterministic data pattern and may optional specify this in the BSDL.

Note: Needs to check if Update stages are allowed for TDRs. INIT_SETUP may update into memory the INIT_DATA at UpdatedR. Will need to discuss in general text. No need for this as a requirement, but it is certainly allowed.

INIT_RUN Rules:

3.a) The optional INIT_RUN instruction shall place either an INIT_STATUS register, or the BYPASS register between TDI and TDO.

3.b) If the optional INIT_RUN instruction utilizes the BYPASS register, said register shall behave, with respect to Capture-DR and Shift-DR, same as it does the BYPASS instruction.

3.c) If the optional INIT_RUN instruction utilizes an INIT_STATUS register, then current status data shall be captured in INIT_STATUS on the rising edge of TCK in the Capture-DR TAP controller state.

3.d) The minimum required length for the INIT_STATUS register is 2 bits. Otherwise, the length is determined by the device designer. The minimum two bits shall the following defined meanings:

1) Bit 0 shall indicate either a "Completed" Status if set to 1 or a "In-Progress" status if set to 0.

2) Bit 1 shall indicate either a "Init Pass" Status if set to 1 or a "Init Fail" Status if set to a 0. If Bit 0 is indicating an "In-progress" status, then Bit 1 must indicate "Init Fail" until the Initialization process has successfully completed. A case of "In-Progress" and "Pass" is cto be considered an error condition.

3) If a "Completed" and "Init Pass" status value is scanned out during Shift-DR, then the device is ready Boundary-Scan Testing to occur.

4) If an "Init Fail" and "Completed" status is provided then Rule 1.d) applies.

5) Known "In-Progress" status value(s) shall be captured in INIT_STATUS on the rising edge of TCK in the Capture-DR TAP controller state while the INIT_RUN instruction is still in execution.

~~1) If a "completed" status value is scanned out during Shift-DR, then the device is ready Boundary-Scan Testing to occur.~~

NOTE for working group: BSDL should not specify the values for either INIT_SETUP or INIT_STATUS. This should be done in a side file.

~~2) A known "in progress" status value(s) shall be captured in INIT_STATUS on the rising edge of TCK in the Capture-DR TAP controller state while the INIT_RUN instruction is still in execution.~~

3.d) The device designer shall document the encodings of the INIT_STATUS in device system documentation to enable ~~board designers~~ to complete board level Boundary-Scan testing.

3.e) The content of the INIT_STATUS register shall be shifted out on TDO on falling edges of TCK while in the Shift-DR TAP controller state.

3.f) Data on TDI shall be shifted in to the INIT_STATUS register on the rising edge of TCK in the Shift-DR TAP controller state.

3.g) Any data shifted into INIT_STATUS, or updated by a falling edge of TCK in the Update-DR TAP controller state, shall have no effect on the device.

3.h) The INIT_RUN instruction, based on the optional INIT_DATA register set to a value appropriate for the current board design and netlist, shall enable creation of an end state which is appropriate for the board level testing.

3.i) The creation of the end state shall occur after one or more of these events:

- 1) a specified maximum number of TCK falling edges in any state besides Test-Logic-Reset, and/or,
- 2) a specified minimum amount of elapsed time, and/or,
- 3) a specified minimum number of system clocks, and/or,
- 4) the presentation of a "Completed" -status value in the INIT_STATUS register during Shift-DR.

Note to group: Complex INIT completion dependent on a system clock is not recommended as system clk are burdensome to board test engineers. Recommend system clock usage should be avoided. Recommend using only one of TCK cycles, elapsed time or system clocks. Recommend only specifying the case(s) which are needed.

3.j) If a minimum number of system clock is specified in 3.i.3) then the system clock shall not be required once the INIT_RUN instruction is no longer the active instruction.

3.kj) The end state shall ~~at least~~ persist until one of these events occurs:

- 1) Power is removed from the device,
- 2) The device enters a 1149.1/~~1149.6~~ nNon-compliant state (due to change in compliance enable pins)

3) The device enters Test-Logic-Reset state. After each exit from Test-Logic-Reset, the INIT_SETUP and INIT_RUN sequence shall be required before boundary scan.

4) A new INIT_SETUP and INIT_RUN sequence is run with a different setting.

5) If any other instruction is set by UpdateIR while the INIT_RUN has not not completed, the chip is left in unknown Initialize state and this action should be avoided. The chip is assumed to be not yet ready for EXTEST type operation.

NOTE: Since the "end state" is a defined system state, a "reset" pin pulse can remove said state per the system definition of that end state. However, once EXTEST is entered, the device is in test mode and cannot respond to said reset pulse.

3.1*) Re-loading INIT_RUN UpdateIR while INIT_RUN was the previous instruction (is active) does not cause a state change (i.e. Re-execution of the INIT_RUN UpdateIR state does not restart the initialization process.)

INIT_RUN Permissions:

3.m±) The binary value(s) for the INIT_RUN instruction may be selected by the device designer.

3.nm) INIT_DATA register and INIT_STATUS register may be the same TDR. INIT_SETUP would treat this register as an input register (CaptureDR) and INIT_RUN treats the register as an output register (ShiftDR).

Strawman BSDL snippet for the new INIT_SETUP/INIT_RUN instructions and the new INIT_DATA / INIT_STATUS TDRs. This is currently just an example to promote discussions. Some existing structures are reused (e.g. Inst opcodes; and some are based on modified structures used for RUNBIST (e.g. WAIT_DURATION → INIT_RUN_DURATION, EXPECT_DATA → STATUS_DATA) since we can may not want the same sematic meanings. New portions are highlighted in blue

...

```
attribute INSTRUCTION_LENGTH of SoC1 : entity is 4;
```

```
attribute INSTRUCTION_OPCODE of SoC1 : entity is
```

```
"EXTEST          (0000)," &      -- Hex 0
"EXTEST_PULSE    (0101)," &      -- Hex 5
"EXTEST_TRAIN    (0110)," &      -- Hex 6
"SAMPLE          (0111), "&      -- Hex 7
"PRELOAD         (0100), "&      -- Hex 4
"CLAMP           (0001), "&      -- Hex 1
"HIGHZ           (0010), "&      -- Hex 2
"IDCODE          (0011), "&      -- Hex 3
"INIT_SETUP      (1101), "&      -- Hex C
"INIT_RUN        (1110), "&      -- Hex E
"BYPASS          (1111), ";      -- Hex F
```

...

```
attribute REGISTER_ACCESS of SoC1 : entity is
```

```
"BYPASS(BYPASS),"&
"BOUNDARY (EXTEST_PULSE, EXTEST_TRAIN, SAMPLE, PRELOAD)," &
"INIT_DATA[100] (INIT_SETUP)," &      -- 100 bit long INIT_DATA TDR
"INIT_STATUS[5] (INIT_RUN)," &      -- 5 bit long INIT_STATUS TDR
"DEVICE_ID (IDCODE)";
```

...

```
attribute INIT_RUN_EXECUTION of SoC1 : entity is
```

```
"INIT_RUN_DURATION (TCK 25), &      -- Max Completion requires 25 TCKs
"STATUS_DATA 00011";                -- in any non-TLR TAP State
-- INIT_STATUS for successful completion is 00011
```