1 The MACTEST/NCSITEST of the SLT Utility

The MACTEST.exe and the NCSITEST.exe can be found in the SLT utility. The newest version of the SLT utility can be downloaded from the web site: http://www.aspeedtech.com/support.php

The MACTEST.exe will send out testing frames and then compare the data value of those receiving frames, it need a loop-back stub to run the testing.

The NCSITEST.exe will only check the basic status of the PHY which support the NCSI protocol by sending request packets and receiving response packets. It will scan all packages ID from 0 to 7, and check the status of each channels. The NCSITEST.exe can work properly after the firmware of the PHY has correctly configured the PHY. Please make sure your PHY firmware is configured correctly.

1.1 Arguments of the NCSITEST

NCSITEST *mac* *package* *channel* *function* *margin*

Default value: *package* = 1, *channel* = 2, *function* = 0, *margin* = 5.

mac = 0: Using the MAC#1 to run the tool.
mac = 1: Using the MAC#2 to run the tool.

package: Total numbers of the shared NIC

channel: Total numbers of the channel of the shared NIC.

function = 0: Check the status of the shared NIC
function = 6: IO timing margin block qualification

margin = 5: Check SCU48 with a 5x1(RMII) solid margin block.
margin = 3: Check SCU48 with a 3x1(RMII) solid margin block.
margin = 0: Check SCU48 with a 1x1 solid margin block.

1.2 Arguments of the MACTEST

MACTEST *mac* *speed* *ctrl* *loop* *function* *phyadr* *margin*

Default value: *speed* = 3, *ctrl* = 8, *loop* = 1, *function* = 0, *phyadr* = 0, *margin* = 5.

mac = 0: Using the MAC#1 to run the tool.
mac = 1: Using the MAC#2 to run the tool.

speed = 0: 1G bps
speed = 1: 100M bps
speed = 2: 10M bps
speed = 3: 1G/100M/10M bps

ctrl[3] = 0: Disable PHY initiation
ctrl[3] = 1: Enable PHY initiation
ctrl[4] = 0: PHY external loopback
ctrl[4] = 1: PHY internal loopback

loop: loop number of the MACTEST.exe

function = 0: Tx/Rx frame checking
function = 1: Tx output 0xff frame
function = 2: Tx output 0x55 frame
function = 3: Tx output random frame
function = 6: IO timing margin block qualification

phyadr: The PHY Address of the dedicated PHY (0~31).

margin = 5: Check SCU48 with a 5x5/5x1 (RGMII/RMII) solid margin block.
margin = 3: Check SCU48 with a 3x3/3x1 (RGMII/RMII) solid margin block.
margin = 0: Check SCU48 with a 1x1 solid margin block.

2 The MACTEST/NCSITEST for the Manufacture Testing

After running the IO timing margin block qualification, please use the result to add a suitable command to the INI file of the SLT utility when running the manufacture testing.

2.1 Shared NIC with the RMII interface

NCSITEST *mac* 1 *channel* 0 *margin*

mac = 0: Using the MAC#1 to run the tool.
mac = 1: Using the MAC#2 to run the tool.
channel = 1: Total number of the channel of the shared NIC is 1.
channel = 2: Total number of the channel of the shared NIC is 2.
channel = 4: Total number of the channel of the shared NIC is 4.
margin = 5: Check SCU48 with a 5x1(RMII) solid margin block.
margin = 3: Check SCU48 with a 3x1(RMII) solid margin block.

NOTE: Any command outside this description only for the debug purpose.

2.2 Dedicated PHY with the RGMII/RMII interface

MACTEST *mac* 3 8 1000 0 *phyadr* *margin*

mac = 0: Using the MAC#1 to run the tool.
mac = 1: Using the MAC#2 to run the tool.
phyadr: The PHY Address of the dedicated PHY (0~31).
margin = 5: Check SCU48 with a 5x5/5x1(RGMII/RMII) solid margin block.
margin = 3: Check SCU48 with a 3x3/3x1(RGMII/RMII) solid margin block.

NOTE: Any command outside this description only for the debug purpose.

3 IO Timing Margin Block Qualification

The AST2300/AST2400/AST2500 has 2 independent MAC, each MAC can be either RGMII or RMII interface, and each MAC has independent register for programming timing delay value.

The SCU48 control the programming timing delay, and the default suggestion value of this register is 0x2255 (0x82208 for AST2500). If you found your SCU48 value is not 0x2255 (0x82208 for AST2500), please contact ASPEED for double check.

In addition, please execute MACTEST.exe or NCSITEST.exe on every of your project, no matter it is a brand new PCB or a revised PCB.
For every of your project, please select 5pcs mother boards randomly and make sure they can all pass MACTEST.exe or NCSITEST.exe.

3.1 The RGMII interface

When the MAC has been configured as the RGMII interface, it has programming timing delay at the Tx and the Rx clock signals. Each platforms need to pick up one of those settings to get a maximum margin block.

AST2300/AST2400

- SCU48[15:12]: MAC#2 RGMII RXCLK clock input delay
- SCU48[11:8]: MAC#1 RGMII RXCLK clock input delay
- SCU48[7:4]: MAC#2 RGMII TXCLK clock output delay
- SCU48[3:0]: MAC#1 RGMII TXCLK clock output delay

AST2500

- SCU48[23:18]: MAC#2 RGMII RXCLK clock input delay
- SCU48[17:12]: MAC#1 RGMII RXCLK clock input delay
- SCU48[11:6]: MAC#2 RGMII TXCLK clock output delay
- SCU48[5:0]: MAC#1 RGMII TXCLK clock output delay

NOTE: If your SCU48 is not 0x2255 (0x82208 for AST2500), please contact ASPEED for double check!

The margin block is suggested to be a 5x5 solid block, and the minimum requirement is a 3x3 solid block. The RGMII interface needs to pass the same solid margin block at the speed of 1G/100M/10M bps.

3.2 The RMII interface

When the MAC has been configured as the RMII interface, it has 16 stages of the programming timing delay at the Rx clock signal, and 2 stages of the programming timing delay at the Tx data signals. Each platforms need to pick up one of 32 settings to get a maximum margin block.

AST2300/AST2400

SCU48[25]: MAC#2 RMII transmit data at clock falling edge
SCU48[24]: MAC#1 RMII transmit data at clock falling edge
SCU48[15:12]: MAC#2 RMII RXCLK clock input delay
SCU48[11:8]: MAC#1 RMII RXCLK clock input delay

AST2500

SCU48[25]: MAC#2 RMII transmit data at clock falling edge
SCU48[24]: MAC#1 RMII transmit data at clock falling edge
SCU48[23:18]: MAC#2 RMII RXCLK clock input delay
SCU48[17:12]: MAC#1 RMII RXCLK clock input delay

NOTE: If your SCU48 is not 0x2255 (0x82208 for AST2500), please contact ASPEED for double check!

The margin block is suggested to be a 5x1 solid block, and the minimum requirement is a 3x1 solid block. The RMII interface need to pass the same solid margin block at the speed of 100M/10M bps.

3.3 Sequence of the IO timing margin block qualification

Step 1. Please plug in the loop-back stub (8 wires) to the RJ45 connector.

Step 2. Please help to make sure the BMC firmware of the testing platform is the newest version, and then the BMC firmware had booted up to the u-boot or the kernel.

Step 3a. [Shared NIC] Please help to run this command for the shard NIC.

NCSITEST *mac* 1 *channel* 6 5

mac = 0: Using the MAC#1 to run the tool.
mac = 1: Using the MAC#2 to run the tool.
channel = 1: Total number of the channel of the shared NIC is 1.
channel = 2: Total number of the channel of the shared NIC is 2.
channel = 4: Total number of the channel of the shared NIC is 4.

NOTE: Any command outside this description only for the debug purpose.

Step 3b. [Dedicated PHY] Please help to run this command for the dedicated PHY.

MACTEST *mac* 3 8 5000 6 *phyadr* 5

mac = 0: Using the MAC#1 to run the tool.
mac = 1: Using the MAC#2 to run the tool.
phyadr: The PHY Address of the dedicated PHY (0~31).

NOTE: Any command outside this description only for the debug purpose.

Step 4. For every of your project, please select 5pcs mother boards randomly and make sure they can all pass MACTEST.exe or NCSITEST.exe.

When the testing result is fail, please help to send back all "*.log" files to ASPEED, ASPEED will help to analyze your log file.

3.4 Example of the qualification with the RGMII interface

```
[1G  ]=======================================
Rx:SCU48[15:12]=    0 1 2 3 4 5 6 7 8 9 a b c d e f
                        |
Tx:SCU48[ 7: 4]=0:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=1:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=2:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=3:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=4:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=5:- o o 0 o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=6:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=7:  o o o x o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=8:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=9:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=a:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=b:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=c:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=d:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=e:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=f:  x x x x x x x x x x x x x x x x
```

This is the test result of the MAC#2 running the IO timing margin block qualification at the speed 1G bps. The IO timing margin block qualification test need to be all passed under the speed of 1G/100M/10M bps.

The symbol, “|” point out the setting of the SCU48[15:12] of the testing platform is the value 0x2, and the symbol, “-” point out the setting of the SCU48[7:4] of the testing platform is the value 0x5. The MACTEST.exe will use the setting of the SCU48 as the center point to qualify the margin block.

This qualification example will be failed when setting the MACTEST.exe to qualify a 5x5 solid margin block. This is because the 5x5 margin block has 24 pass points (“o”), and 1 fail point (“x”).

```
Tx:SCU48[ 7: 4]=3:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=4:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=5:- o o 0 o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=6:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=7:  o o o x o o o o x x x x x x x x
```

In this case, you can find out a maximum 8x6 full pass solid margin block which has 48 pass points (“o”), and the best setting of the SCU48 is one of those center points of the 8x6 full pass solid margin block.

```
Rx:SCU48[15:12]=    0 1 2 3 4 5 6 7 8 9 a b c d e f
                        |
Tx:SCU48[ 7: 4]=1:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=2:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=3:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=4:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=5:- o o 0 o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=6:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=7:  o o o x o o o o x x x x x x x x
```

NOTE: Any change of the SCU48 must be sent to ASPEED for double check.

1. (SCU48[15:12] = 0x3) && (SCU48[7:4] = 0x3)
2. (SCU48[15:12] = 0x3) && (SCU48[7:4] = 0x4)
3. (SCU48[15:12] = 0x4) && (SCU48[7:4] = 0x3)
4. (SCU48[15:12] = 0x4) && (SCU48[7:4] = 0x4)

Besides, those setting of the SCU48 also pass the 5x5 margin block qualification.

```
Rx:SCU48[15:12]=    0 1 2 3 4 5 6 7 8 9 a b c d e f
                        |
Tx:SCU48[ 7: 4]=1:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=2:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=3:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=4:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=5:- o o 0 o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=6:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=7:  o o o x o o o o x x x x x x x x
```

NOTE: Any change of the SCU48 must be sent to ASPEED for double check.

1. (SCU48[15:12] = 0x2) && (SCU48[7:4] = 0x3)
2. (SCU48[15:12] = 0x2) && (SCU48[7:4] = 0x4)
3. (SCU48[15:12] = 0x3) && (SCU48[7:4] = 0x3)
4. (SCU48[15:12] = 0x3) && (SCU48[7:4] = 0x4)
5. (SCU48[15:12] = 0x4) && (SCU48[7:4] = 0x3)
6. (SCU48[15:12] = 0x4) && (SCU48[7:4] = 0x4)
7. (SCU48[15:12] = 0x5) && (SCU48[7:4] = 0x3)
8. (SCU48[15:12] = 0x5) && (SCU48[7:4] = 0x4)

3.4.1 Warnning of the qualification with the RGMII interface

```
[1G  ]=======================================
Rx:SCU48[15:12]=    0 1 2 3 4 5 6 7 8 9 a b c d e f
                        |
Tx:SCU48[ 7: 4]=0:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=1:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=2:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=3:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=4:  o o o o o x o o x x x x x x x x
Tx:SCU48[ 7: 4]=5:- o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=6:  o o o o o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=7:  o o o x o o o o x x x x x x x x
Tx:SCU48[ 7: 4]=8:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=9:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=a:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=b:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=c:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=d:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=e:  x x x x x x x x x x x x x x x x
Tx:SCU48[ 7: 4]=f:  x x x x x x x x x x x x x x x x
```

This is the outcome of the MAC#2 running the IO timing margin block qualification at the speed 1G bps. The fail point(“x”) can’t be enclosed by pass points (“o”) in the vertical or the horizontal direction.

When this is happened, please inform the ASPEED for more help.

3.5 Example of the qualification with the RMII Interface

```
[100M]=======================================
Rx:SCU48[11: 8]=    0 1 2 3 4 5 6 7 8 9 a b c d e f
                        |
Tx:SCU48[   24]=0:- o o 0 o o o o o x x x x o o o o
Tx:SCU48[   24]=1:  x x x x x o o o o o o o o o o o
```

This is the test result of the MAC#1 running the IO timing margin block qualification at the speed 100M bps. The IO timing margin block qualification test need to be all passed under the speed of 100M/10M bps.

The symbol, "|" point out the setting of the SCU48[11:8] of the testing platform is the value 0x2, and the symbol, "-" point out the setting of the SCU48[24] of the testing platform is the value 0x0. The MACTEST.exe/NCSITEST.exe will use the setting of the SCU48 as the center point to qualify the margin block.

This qualification example will be passed when setting the MACTEST.exe/NCSITEST.exe to qualify a 5x1 solid margin block. This is because the 5x1 margin block has 5 pass points ("o").

```
Tx:SCU48[   24]=0:- o o 0 o o o o o x x x x o o o o
Tx:SCU48[   24]=1:  x x x x x o o o o o o o o o o o
```

In this case, you can find out a maximum 11x1 full pass solid margin block which has 11 pass points ("o"), and the best setting of the SCU48 is one of those center points of the 11x1 full pass solid margin block.

```
Rx:SCU48[11: 8]=    0 1 2 3 4 5 6 7 8 9 a b c d e f
                        |
Tx:SCU48[   24]=0:- o o 0 o o o o o x x x x o o o o
Tx:SCU48[   24]=1:  x x x x x o o o o o o o o o o o
```

NOTE: Any change of the SCU48 must be sent to ASPEED for double check.

1. (SCU48[11:8] = 0xa) && (SCU48[24] = 0x1)

Besides, those setting of the SCU48 also pass the 5x1 margin block qualification.

```
Rx:SCU48[11: 8]=     0 1 2 3 4 5 6 7 8 9 a b c d e f
                             |
Tx:SCU48[   24]=0:- o o 0 o o o o o x x x x o o o o
Tx:SCU48[   24]=1:  x x x x x o o o o o o o o o o o
```

NOTE: Any change of the SCU48 must be sent to ASPEED for double check.

1. (SCU48[11:8] = 0x2) && (SCU48[24] = 0x0)
2. (SCU48[11:8] = 0x3) && (SCU48[24] = 0x0)
3. (SCU48[11:8] = 0x4) && (SCU48[24] = 0x0)
4. (SCU48[11:8] = 0x5) && (SCU48[24] = 0x0)
5. (SCU48[11:8] = 0x7) && (SCU48[24] = 0x1)
6. (SCU48[11:8] = 0x8) && (SCU48[24] = 0x1)
7. (SCU48[11:8] = 0x9) && (SCU48[24] = 0x1)
8. (SCU48[11:8] = 0xa) && (SCU48[24] = 0x1)
9. (SCU48[11:8] = 0xb) && (SCU48[24] = 0x1)
10. (SCU48[11:8] = 0xc) && (SCU48[24] = 0x1)
11. (SCU48[11:8] = 0xd) && (SCU48[24] = 0x1)