SATA Host reference design on Altera V-series/10-series manual

Rev1.4 23-Aug-23

SATA Host reference design on Altera V-series/10-series manual

Rev1.4 23-Aug-23

1. Introduction

Serial ATA (SATA) is an evolutionary replacement for the Parallel ATA (PATA) physical storage interface. SATA interface increases speed transfer to be 1.5 Gbps for SATA-I, 3.0 Gbps for SATA-II, and 6.0 Gbps for SATA-III. To communication by SATA protocol, there are four layers in its architecture, i.e. Application, Transport, Link, and Phy.

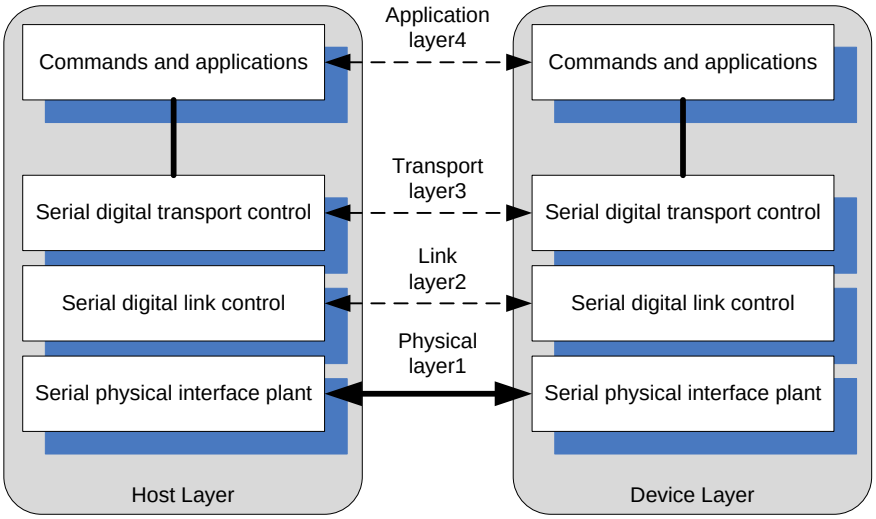

Figure 1-1 SATA Communitcation Layer

The Application layer is responsible for overall ATA command execution, including controlling Command Block Register accessed. The Transport layer is responsible for placing control information and data to be transferred between the host and device in a packet/frame, known as a Frame Information Structure (FIS). The Link layer is responsible for taking data from the constructed frames, encoding or decoding each byte using 8b/10b, and inserting control characters such that the 10-bit stream of data may be decoded correctly. The Physical layer is responsible for transmitting and receiving the encoded information as a serial data stream on the wire.

This reference design provides evaluation system which implements all SATA communication layers for Host side to transfer high speed data with SATA-II or SATA-III Device (HDD/SSD). SATA-IP core is designed to operate with Transceiver on V-series device and 10-series device such as CycloneV, ArriaV, StratixV, and Arria10. More details are described as follows.

2. Hardware description

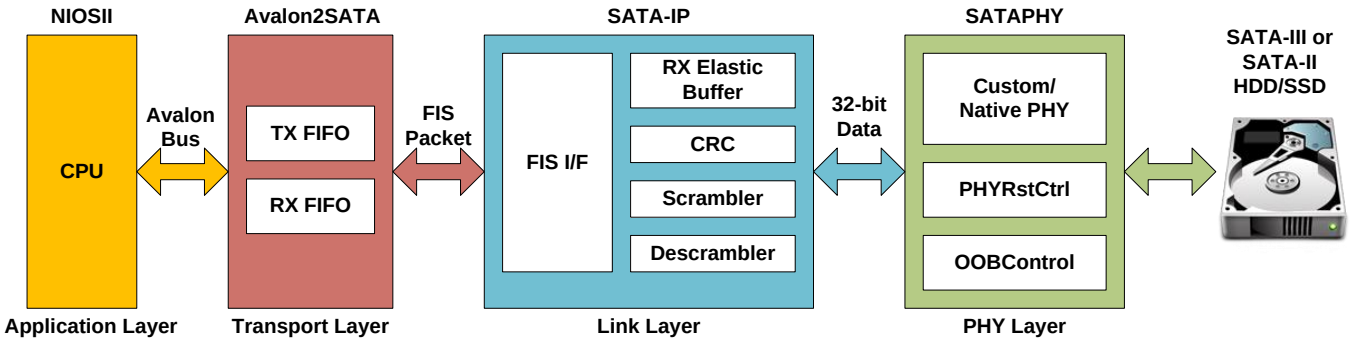

Figure 2-1 Hardware Architecture in reference design

As shown in Figure 2-1, hardware architecture can be divided into 4 modules to support each SATA layer protocol. SATA-IP implements Link Layer and some part of Transport Layer, so users need to prepare other Layer such as PHY Layer and Transport Layer by themselves. This reference design shows the example of Transport Layer and PHY Layer example based on Altera V-series development board.

- PHY Layer

This layer is designed by using built-in high speed serial component and use CustomPHY or NativePHY component from Altera IP wizard to set operation parameter. Internal logic is designed to control PHYIP component. State machine within OOBMainCtrl submodule controls OOB (Out-of-band) sequence of SATA protocol and co-operate with OOBComCtrl submodule which is used for generating/monitoring COMRESET, COMINIT, and COMWAKE signal. PHYRstCtrl controls reset sequence of transceiver characteristic.

PHY Layer is stored in “hdl/sata2phy.vhd or sata3phy.vhd” including “OOBComCtrl.vhd”, “PhyRstCtrl.vhd”, and “OOBMainCtrl.vhd”. All hdl source codes on PHY Layer will be provided in delivery items for customer.

- Link Layer by SATA-IP

The details of SATA IP core are described in“dg_sata3_ip_datasheet_altera5_en.pdf”.

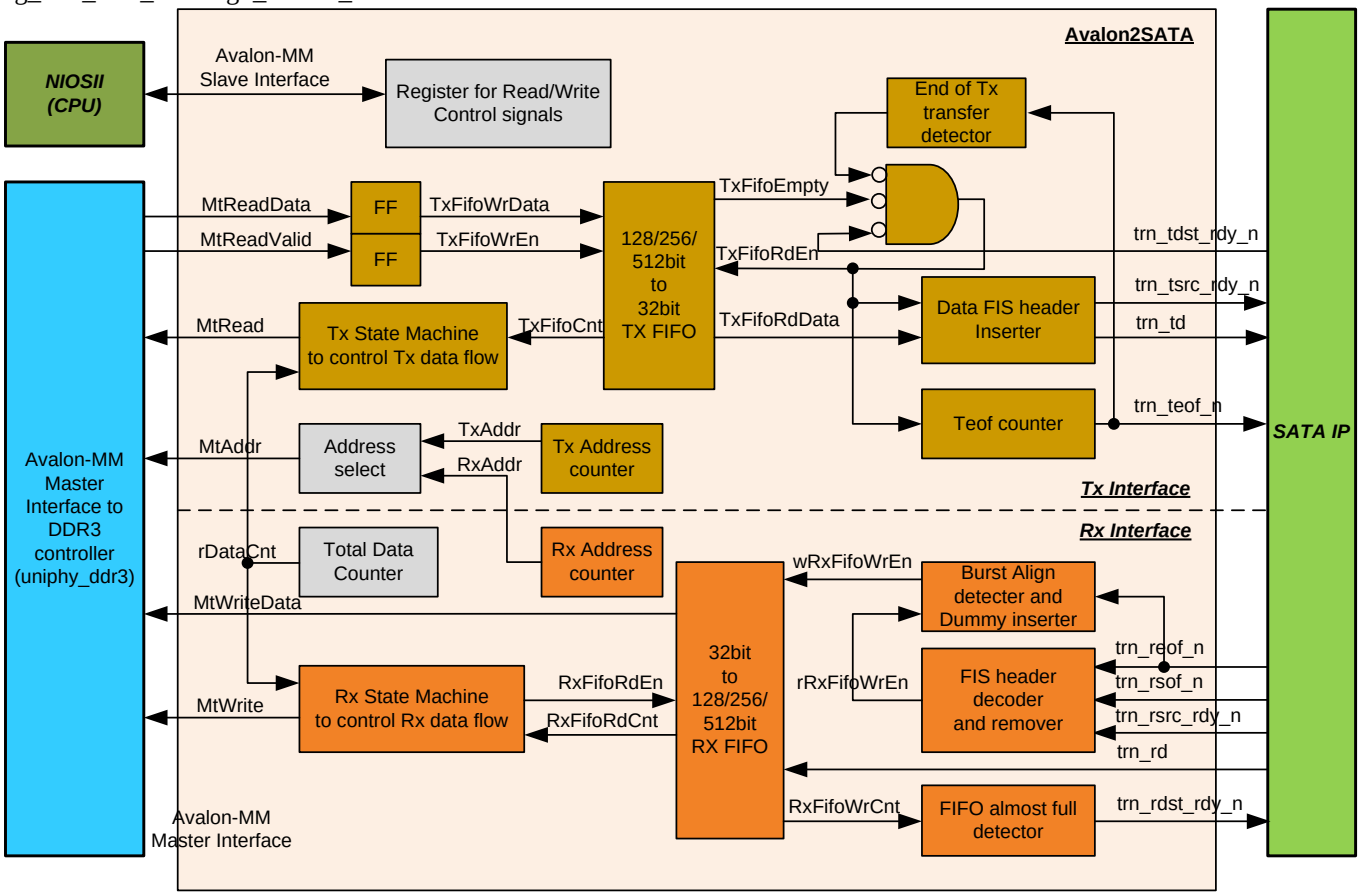

Figure 2-2 Avalon2SATA block diagram

- Transport Layer

Similar to typical SATA controller, CPU builds FIS data on Main memory (DDR3/DDR4) and send control signal to lower layer to start DMA mechanism with Main memory. To support DMA data transfer, Avalon-MM Master interface is designed within Avalon2SATA module, while control signal with CPU is designed by Avalon-MM Slave interface. For high performance transfer, Avalon-MM Master interface is designed to fixed 512-byte burst size. Bus size is different on each platform depending on DDR component on the platform. There are three styles in reference design, i.e. 8-beat of 512-bit data bus, 16-beat of 256-bit data bus, or 32-beat of 128-bit data bus for 512-byte transaction.

To write FIS packet to SATA-IP, each 512-byte data will be request and store in TXFIFO until Total Data Counter decrement to zero value (all data are transferred). If current transfer is Data FIS packet, Data FIS header will be auto-generated to SATA-IP. If total data size is aligned with 512-byte unit, all data in TXFIFO will be transferred to SATA-IP. In case of unaligned size, only request size will be transferred to SATA-IP. Dummy data from CPU still remain in TXFIFO and will be flush by reset signal.

To read FIS packet from SATA-IP, FIS header will be decoded and auto-removed for data FIS header, so only pure data will be stored to RX FIFO. If total data from SATA-IP is not aligned with 512-byte size, dummy data will be auto-added to align burst size. Then, data with dummy will be dumped to DDR. In Avalon2SATA module, two state machines are designed to control each transfer direction independently.

Transport Layer source code is stored in MMMt2SATA128, MMMt2SATA256, or MMMt2SATA512.vhd that also includes SATA IP Core and PHY Layer instance.

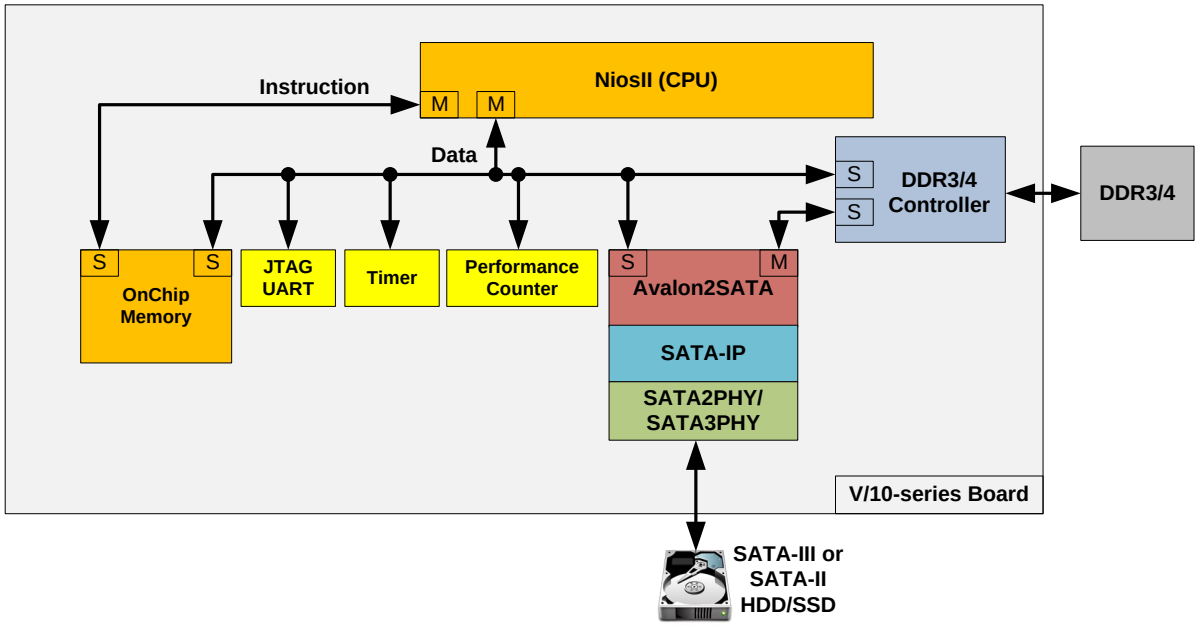

Figure 2-3 Block diagram of the reference design

- Application Layer by NiosII system

This reference design uses NiosII as a host processor, and uses DDR to be main memory which can be accessed by both NiosII and user logic connection for SATA-IP. DDR memory is split into four areas to store DATA FIS and other FIS of write and read operation, as shown in Figure 2-4. Program and data memory of NiosII is designed by using OnChip memory and user can send command and monitor result with NiosII by using JTAG UART. Since test result on test application shows transfer performance of each SATA device, Timer is required in NIOSII system.

Register map of control signals for this reference design is shown in Table 2-1. Software on NiosII along with DMA mechanism will transmit FIS data on the main memory to the Link Layer of SATA IP, or will receive data from the Link Layer to the main memory. Memory address and transfer size are controlled by Software. Data FIS header is added and removed by Avalon2SATA module, so Software do not need to mind about header information control for data management. After ending of each transfer, Finish flag will be set to alert NiosII and then start next data transfer until complete. More details about Software are described in next topic.

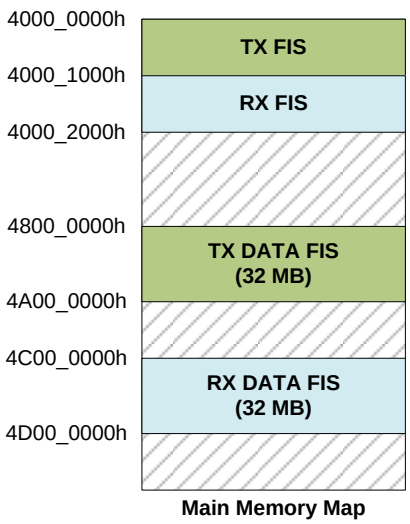

Figure 2-4 Memory map of Main Memory (DDR)

Address Rd/Wr	Register Name (Label in the “sata_host.c”)	Description (Bit order is little endian)
BA Rd	PHY Status Reg (PHY_STATUS)	[1] : SATA Speed (‘1’: SATA-III, ‘0’: SATA-II). The design on CycloneV SX SoC is fixed to SATA-II while other boards are fixed to SATA-III [0] : SATA PHY LINKUP (‘0’: No SATA available, ‘1’: PHY initialize complete)
BA+0x04 Rd	Error Code Reg. (ERROR_CODE)	SATA IP Error code after transmit/receive completion to detect CRC or FIS error.
BA+0x0C Rd	Receive Word Count Reg. (RX_COUNT)	[31] : Received FIS type in this interrupt (‘1’: Non-Data FIS, ‘0’: Data FIS). This bit is cleared by writing bit[29] of INT_CLEAR = ‘1’. [23:0] : Total Received word count of FIS data. Auto clear when next transfer start (CONTROL Reg is written).
BA+0x10 Rd	Interrupt Clear Reg (INT_CLEAR)	[31] : Received finish flag. Set after IP sending packet to host. [30] : Transmit finish flag. Set after host sending packet to IP.
BA+0x00 Wr	Transmit Data Address Reg. (TX_ADDR)	Set DDR start address of transmit FIS data area Bit[8:0] of this value needs to be equal to 0 for sector alignment.
BA+0x04 Wr	Received Data Address1 Reg. (RX_ADDR)	Set DDR start address of received other FIS area (except DATA FIS type). Bit[8:0] of this value needs to be equal to 0 for sector alignment.
BA+0x08 Wr	Control Reg. (CONTROL)	[31] : Hardware Reset [30] : Start Transmit data (‘1’: TX, ‘0’: RX) [29] : FIS type (‘1’: Data, ‘0’: Others) [15:0] : Total Transmit word count. RX_COUNT register is cleared by write operation to this register
BA+0x0C Wr	Received Data Address2 Reg. (RXD_ADDR)	Set DDR start address of received DATA FIS area Bit[8:0] of this value needs to be equal to 0.
BA+0x10 Wr	Interrupt Clear Reg (INT_CLEAR)	[31] : Set this bit to clear Received finish flag [30] : Set this bit to clear Transmit finish flag [29] : Set this bit to clear received FIS type (RX_COUNT bit 31)

Table 2-1 Register mapping from CPU side

(BA : Base Address = 0x01000000)

3. Software description

- SATA Device access via FIS

Communication between the Host and the Device via SATA is done by FIS (Frame Information Structure) data structure. NiosII in the Host design will build FIS data structure on its main memory space, and will send it to the Device by DMA controller that operates bus master. And FIS data sent from the Device is also transferred on the main memory by DMA controller.

Thus, NiosII will execute access to the SATA Device by the following sequence.
(1) Build FIS Data structure (First FIS command should be RegH2D FIS)
(2) Transmit FIS Data
(3) Wait to receive FIS Data
(4) Read received FIS Data
(5) Additional FIS data transmit/receive if necessary.

FIS transmit and receive counts are different according to the protocol type, but the brief sequence should be as above.

- Software of reference design

Software of this reference design implements three popular commands, i.e. IDENTIFY DEVICE, READ DMA EXT/READ DNA, and WRITE DMA EXT/WRITE DMA. This reference design can support both 48-bit and 28-bit LBA (Logical Block Address) mode to transfer data with SATA Device.
When SATA Device is powered on, it always sends Register – Device to Host FIS, so Host must wait this FIS from SATA Device before issue first command.

- IDENTIFY DEVICE

Table 3-1 shows FIS structure of IDENTIFY COMMAND that gets device information from SATA Device. This command requires parameter settings for its Command Opcode (ECH) and device number that is typically set to zero in SATA device. Device register value will be A0H because obsolete bit#7 and bit#5 are recommended to set. “C” bit should be set whenever SATA Host sends command to the SATA Device, and it is also the same for all other commands issue.
After finish parameter settings to Register – Host to Device FIS, Host sends it to Link Layer. SATA Device will firstly send PIO Setup FIS, and then send Data FIS that includes device information.
For detailed device information, please refer to the ATA Standard document that can be obtained from http://www.t13.org/. This reference design only shows device model number, 48bit LBA support information, and disk capacity information.

0	Features 00h	command ECh	C R R R PM Port 1 0 0 0 0h	FIS Type (27h)
1	Device A0h	LBA High 00h	LBA Mid 00h	LBA Low 00h
2	Features (exp) 00h	LBA High (exp) 00h	LBA Mid (exp) 00h	LBA Low (exp) 00h
3	Control 00h	Reserved (0)	sector Count (exp) 00h	Sector Count 00h
4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

Table 3-1 IDENTIFY COMMAND FIS structure

- READ DMA EXT

Table 3-2 shows FIS structure of READ DMA EXT that reads data from SATA Device in 48-bit LBA mode. READ DMA command will be used instead to read data in 28-bit LBA mode. There are two data transfer types, i.e. PIO and DMA, but their difference is insignificant for SATA case. In SATA Device, speed performance of PIO transfer and DMA transfer are also not so different. Because DMA Read process is easier than PIO, this reference design selects DMA transfer.

Host will set Opcode to 25H for 48-bit mode or C8H for 28-bit mode, LBA bit (bit#6) of Device register, LBA address, and read sector count to the Register – Host to Device FIS, and then transmit to SATA Device. Device will send read data equal with read sector count setting in Data FIS, and then send Register – Device to Host FIS to finish this command.

0	Features 00h	command 25h	C R R R PM Port 1 0 0 0 0h	FIS Type (27h)
1	Device E0h	LBA High LBA[23:16]	LBA Mid LBA[15:8]	LBA Low LBA[7:0]
2	Features (exp) 00h	LBA High (exp) LBA[47:40]	LBA Mid (exp) LBA[39:32]	LBA Low (exp) LBA[31:24]
3	Control 00h	Reserved (0)	sector Count (exp) sector_count[15:8]	Sector Count sector_count[7:0]
4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

Table 3-2 DMA READ EXT FIS structure

- WRITE DMA EXT

Table 3-3 shows FIS structure of WRITE DMA EXT that writes data to SATA Device in 48-bit LBA mode. WRITE DMA command will be used instead to write data in 28-bit LBA mode. FIS structure is almost identical to that of READ DMA EXT. Host will set Opcode to 35H for 48-bit mode or CAH for 28-bit mode, LBA bit, LBA address, write sector count. After sending this Host to Device FIS, Device will send DMA Activate FIS to the Host. Then, Host sends first Data FIS to the Device.

Host will repeat sending Data FIS to the Device until all the data transfer is completed. Finally, Device sends Register- Device to Host FIS to finish this command.

0	Features 00h	command 35h	C R R R PM Port 1 0 0 0 0h	FIS Type (27h)
1	Device E0h	LBA High LBA[23:16]	LBA Mid LBA[15:8]	LBA Low LBA[7:0]
2	Features (exp) 00h	LBA High (exp) LBA[47:40]	LBA Mid (exp) LBA[39:32]	LBA Low (exp) LBA[31:24]
3	Control 00h	Reserved (0)	sector Count (exp) sector_count[15:8]	Sector Count sector_count[7:0]
4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

Table 3-3 DMA WRITE EXT FIS structure

- Necessary consideration

Host software source code of this design is stored in “software/sata_host/sata_host.c”. Note that this reference design does not include error check or recovery from illegal/unexpected behavior. So user needs to add such consideration that software should check status or error check when Register – Device to Host FIS is received from the Device.

Figure 3-1 shows reference design operation result on serial terminal screen.

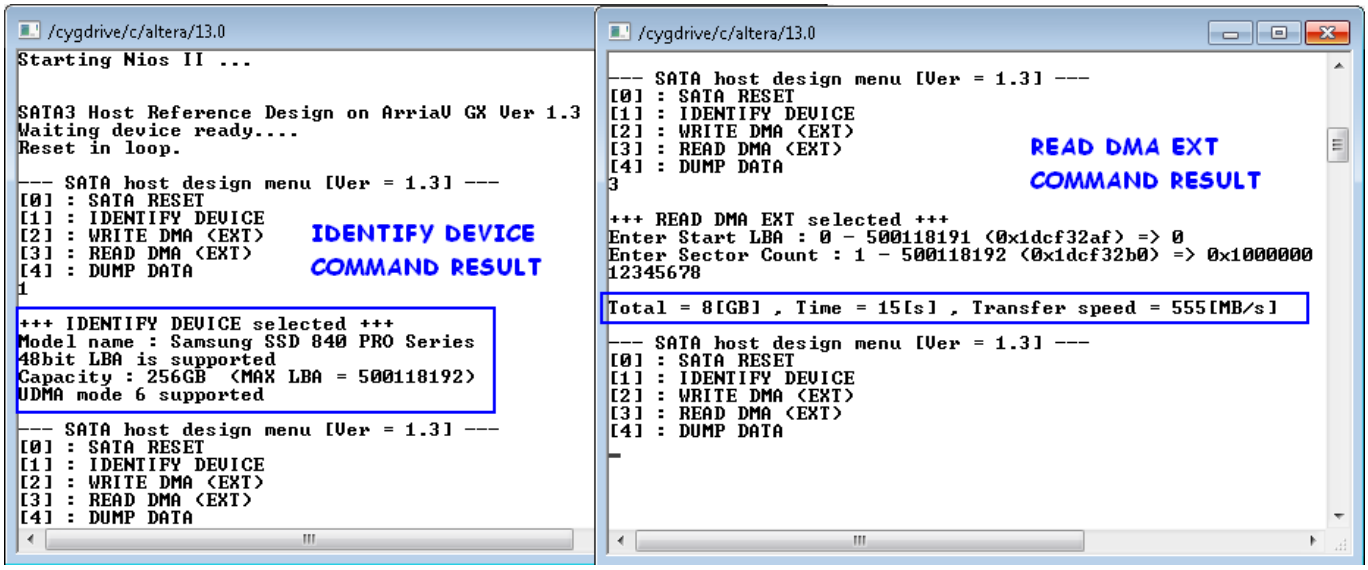

Figure 3-1 Operation result sample screen

4. Revision History

Revision	Date	Description
1.2	24-Nov-15	Merge all V-series to one document
1.3	3-Mar-16	Support Arria V ST SoC development board
1.4	27-Jun-16	Support 10-series device

Copyright: 2015 Design Gateway Co,Ltd.