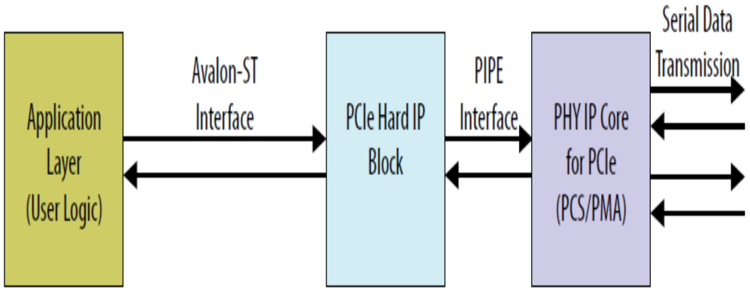
Altera PCIe reference design testing

CRU INDIA TEAM

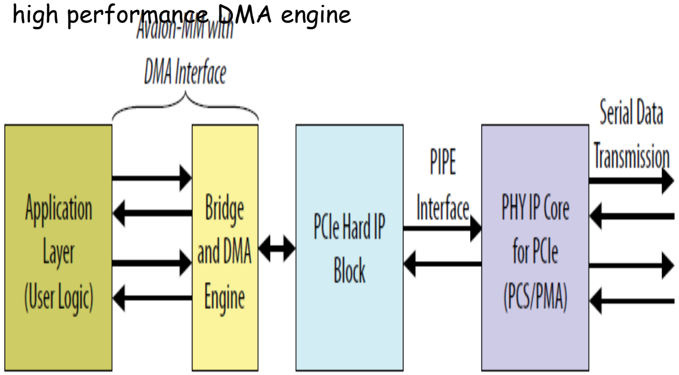
Example Design :Variation

We have found four example designs :

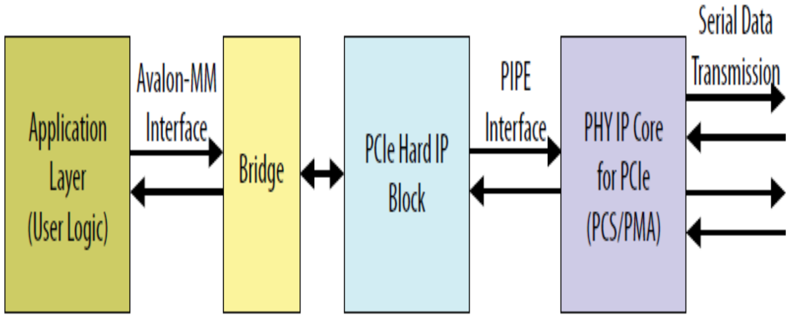
1. Stratix V Avalon-ST Interface for PCIe Solutions -- for better understanding of PCIe hip



3. V-Series Avalon-MM DMA Interface for PCIe Solutions – high performance DMA engine

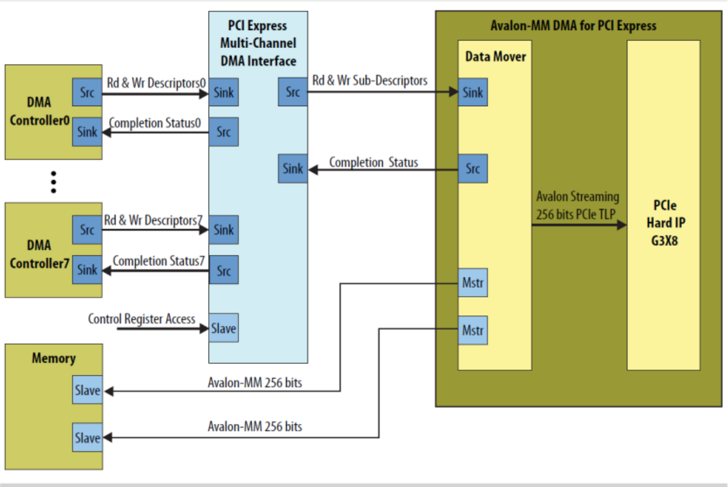

Avalon-MM with DMA Interface
Application Layer (User Logic)
Bridge and DMA Engine
PCIe Hard IP Block
PIPE Interface
PHY IP Core for PCIe (PCS/PMA)
Serial Data Transmission

2. Stratix V Avalon-MM Interface for PCIe Solutions— for quick design



4. PCI Express Multi-Channel DMA Interface--multichannel DMA ,support 8chs, one mem for 8 chs

Figure 1: PCI Express Multi-Channel DMA Interface Example Design System-Level Block Diagram



PCIe_Testing

- There are multiple PCIe reference design available in web . We have selected reference design based on 1st and 3rd example design:
  -- Stratix V Avalon-ST Interface for PCIe Solutions
  -- V-Series Avalon-MM DMA Interface for PCIe Solutions

- APPs

  --Both the reference design includes a Windows-based software application that sets up the DMA transfers. The software application also measures and displays the performance achieved for the transfers.

  --We have another jungo app that detect the presence of Altera board on PCI bus and fetch the details of reference design.

- Ref design selection:

  --Reference design1 based on 1st example design has multiple variation .We have selected the three of them
  --Gen 1 X8/Gen2 X8/Gen3 X4

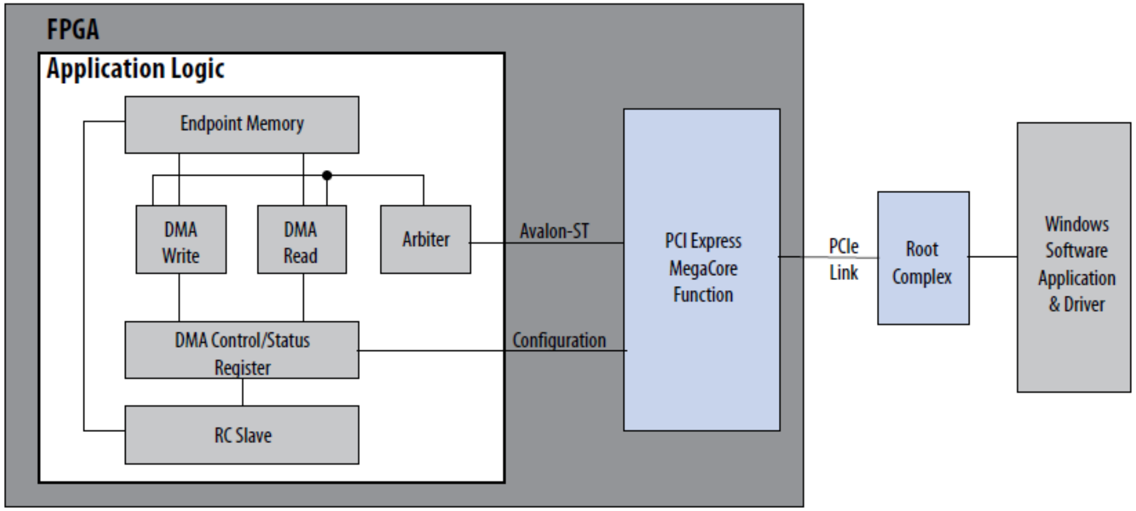
  --Reference design2 based on 3rd example design also has multiple variation .We have selected the one of them
  --Gen 3 X8

- There are two computers in VECC ,one support upto Gen2X8 and another upto Gen3 X8.

- We are not able to install the windows application for ref design 1 on computer that support upto Gen3X8.
  So,the result for DMA is for Gen2 X8

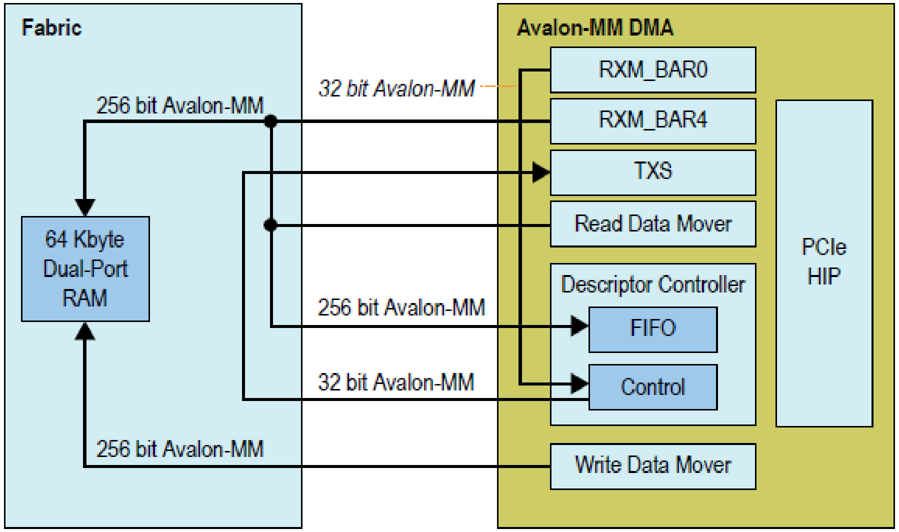
- We are able to install the drivers for ref design 2 on computer that support upto Gen3X8.Device manager has listed the Altera board. But, not able to run the application prog. till now.. So, right now we do not have results for DMA of reference design 2.

Reference design 1



User design contains the DMA engine

Reference design 2



User design contains the only the dual port RAM,
DMA is accessed through MM regs

Board and system setup

- Before prog. the FPGA,we have to change some DIP switches setting to create the testing environment and manage the trade off b/n configuration time and bus enumeration time

- **Problem**

  --Configuration time is much much greater than Bus enumeration time

  --FPGA will be still in configuration mode after bus enumeration, So, host will not detect PCIe based FPGA board

- **Reasons behind solution:**

  --FPGA needs power for configuration.

  --After configuration FPGA needs Power (either via PCI BUS or externally) to hold the configuration.

  --But when we shut down and restart or restart the host.,FPGA will not get power for some time in case of Power source=PCI BUS.

  --So, FPGA will lose its configuration.

  --So, We need to apply external power for the time being.

Solution:

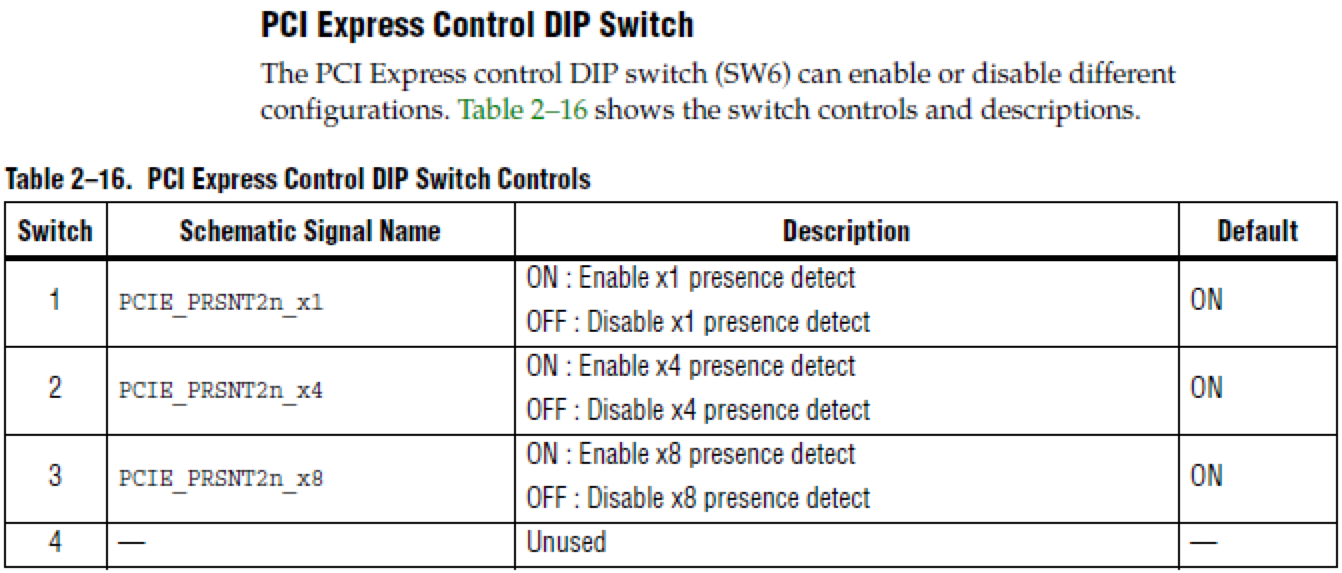
--apply external power to prog. the FPGA

--shut down and restart or restart the host i.e bus enumeration (FPGA remains configured ,since we have applied external power)

--so host will detect FPGA board on PCI bus

- **To manage the trade off b/n configuration time and bus enumeration time, there are other solutions like CvP and FPPX32**
- **But, CvP is not supported by Gen3 X8 mode**
- **We are trying to program the FLASH using PFL.**

PCIe Control DIP Switch



PCI Express Control DIP Switch

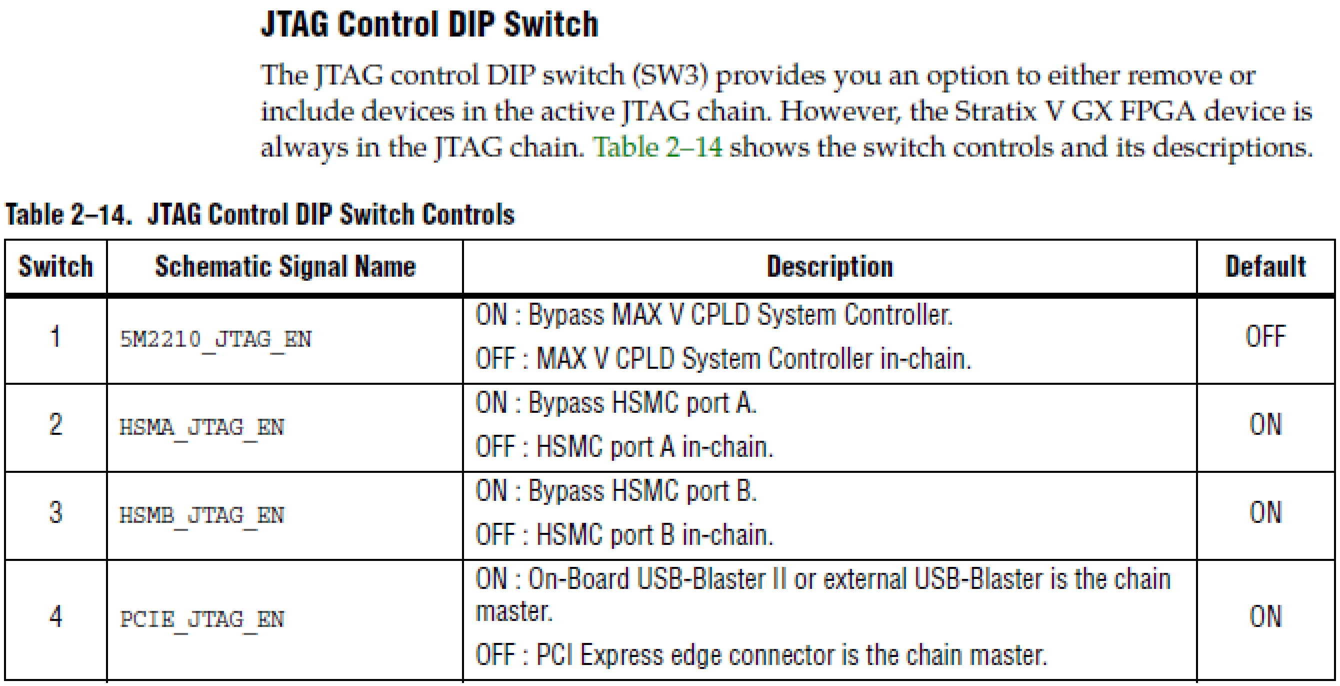
The PCI Express control DIP switch (SW6) can enable or disable different configurations. Table 2–16 shows the switch controls and descriptions.

Table 2–16. PCI Express Control DIP Switch Controls

Switch	Schematic Signal Name	Description	Default
1	`PCIE_PRSNT2n_x1`	ON : Enable x1 presence detect OFF : Disable x1 presence detect	ON
2	`PCIE_PRSNT2n_x4`	ON : Enable x4 presence detect OFF : Disable x4 presence detect	ON
3	`PCIE_PRSNT2n_x8`	ON : Enable x8 presence detect OFF : Disable x8 presence detect	ON
4	—	Unused	—

SW setting: OFF OFF ON ON

JTAG Control DIP Switch



JTAG Control DIP Switch

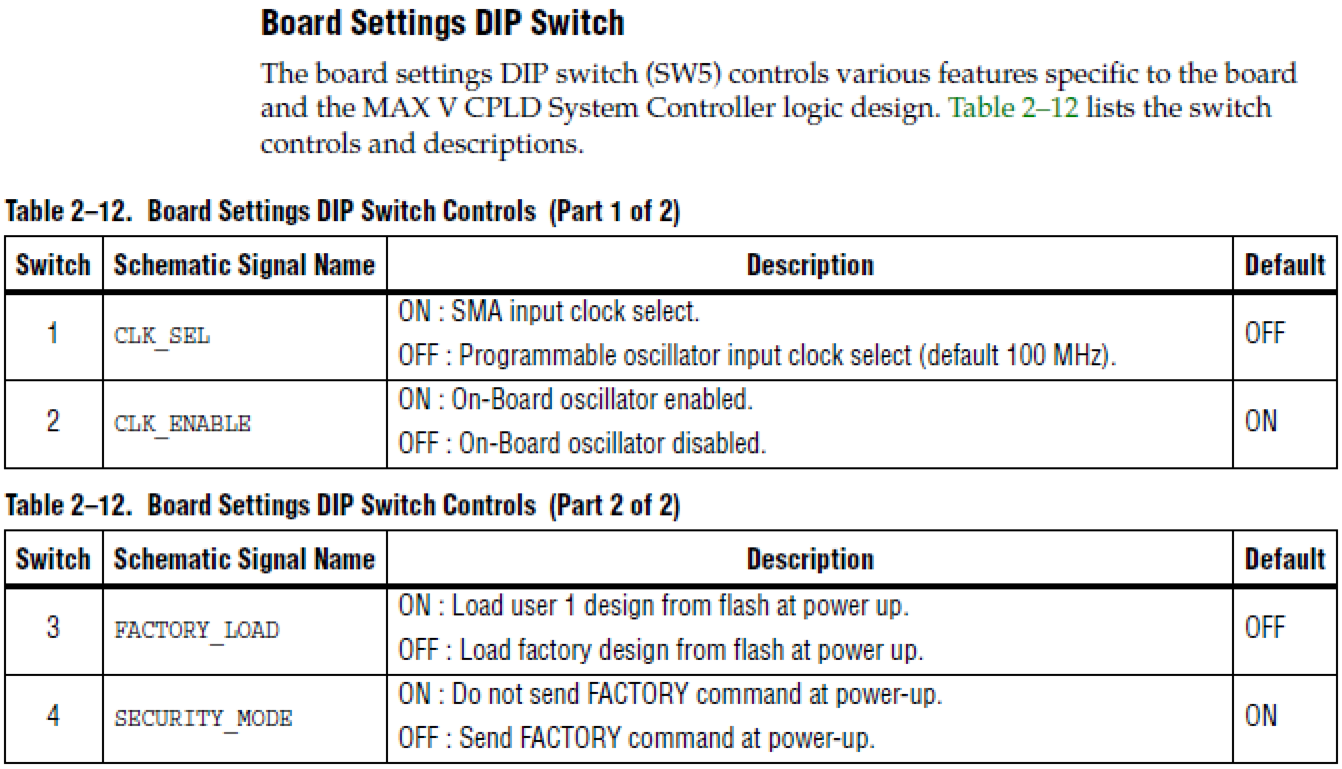
The JTAG control DIP switch (SW3) provides you an option to either remove or include devices in the active JTAG chain. However, the Stratix V GX FPGA device is always in the JTAG chain. Table 2–14 shows the switch controls and its descriptions.

Table 2–14. JTAG Control DIP Switch Controls

Switch	Schematic Signal Name	Description	Default
1	`5M2210_JTAG_EN`	ON : Bypass MAX V CPLD System Controller. OFF : MAX V CPLD System Controller in-chain.	OFF
2	`HSMA_JTAG_EN`	ON : Bypass HSMC port A. OFF : HSMC port A in-chain.	ON
3	`HSMB_JTAG_EN`	ON : Bypass HSMC port B. OFF : HSMC port B in-chain.	ON
4	`PCIE_JTAG_EN`	ON : On-Board USB-Blaster II or external USB-Blaster is the chain master. OFF : PCI Express edge connector is the chain master.	ON

SW setting: ON ON ON ON

Board setting DIP Switch



Board Settings DIP Switch

The board settings DIP switch (SW5) controls various features specific to the board and the MAX V CPLD System Controller logic design. Table 2–12 lists the switch controls and descriptions.

Table 2–12. Board Settings DIP Switch Controls (Part 1 of 2)

Switch	Schematic Signal Name	Description	Default
1	CLK_SEL	ON : SMA input clock select. OFF : Programmable oscillator input clock select (default 100 MHz).	OFF
2	CLK_ENABLE	ON : On-Board oscillator enabled. OFF : On-Board oscillator disabled.	ON

Table 2–12. Board Settings DIP Switch Controls (Part 2 of 2)

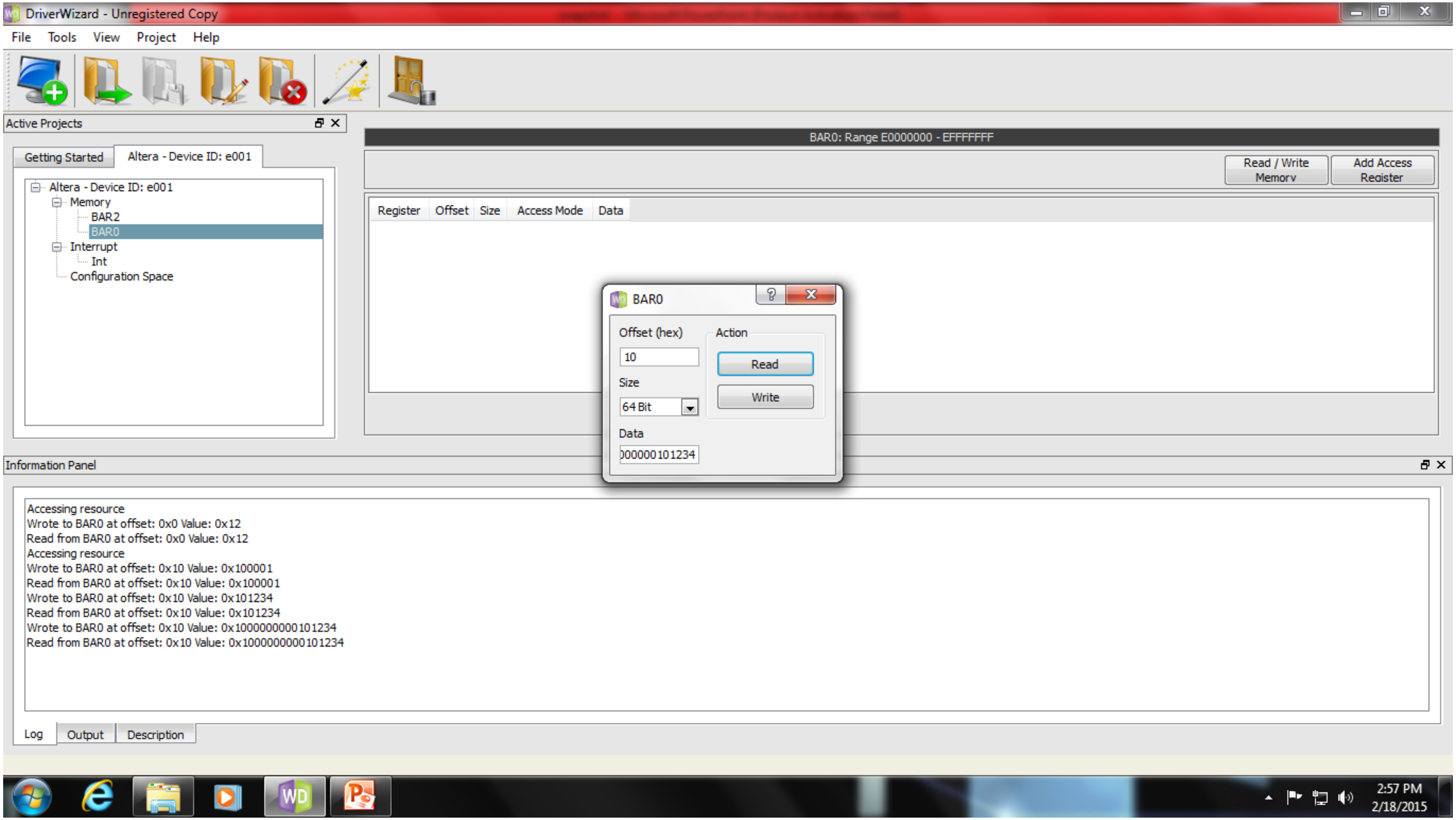
Switch	Schematic Signal Name	Description	Default
3	FACTORY_LOAD	ON : Load user 1 design from flash at power up. OFF : Load factory design from flash at power up.	OFF
4	SECURITY_MODE	ON : Do not send FACTORY command at power-up. OFF : Send FACTORY command at power-up.	ON

SW setting: OFF ON ON OFF

Jungo APP results

Gen1 x8 Avalon ST reference design1: Results

Memory Read/Write



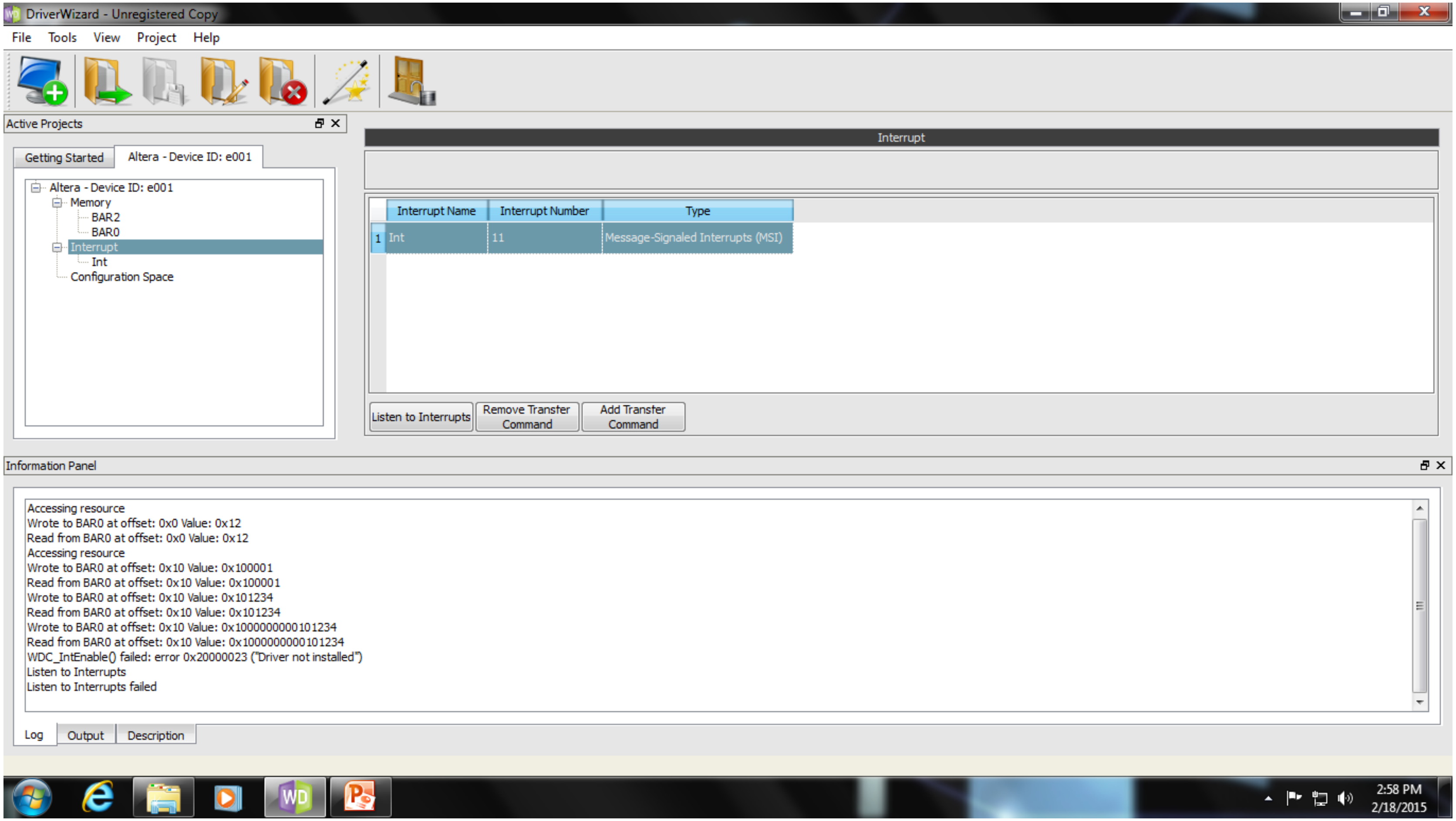
Offset (hex): 10

Size: 64 Bit

Data: 000000101234

```
Accessing resource
Wrote to BAR0 at offset: 0x0 Value: 0x12
Read from BAR0 at offset: 0x0 Value: 0x12
Accessing resource
Wrote to BAR0 at offset: 0x10 Value: 0x100001
Read from BAR0 at offset: 0x10 Value: 0x100001
Wrote to BAR0 at offset: 0x10 Value: 0x101234
Read from BAR0 at offset: 0x10 Value: 0x101234
Wrote to BAR0 at offset: 0x10 Value: 0x1000000000101234
Read from BAR0 at offset: 0x10 Value: 0x1000000000101234
```

Interrupt

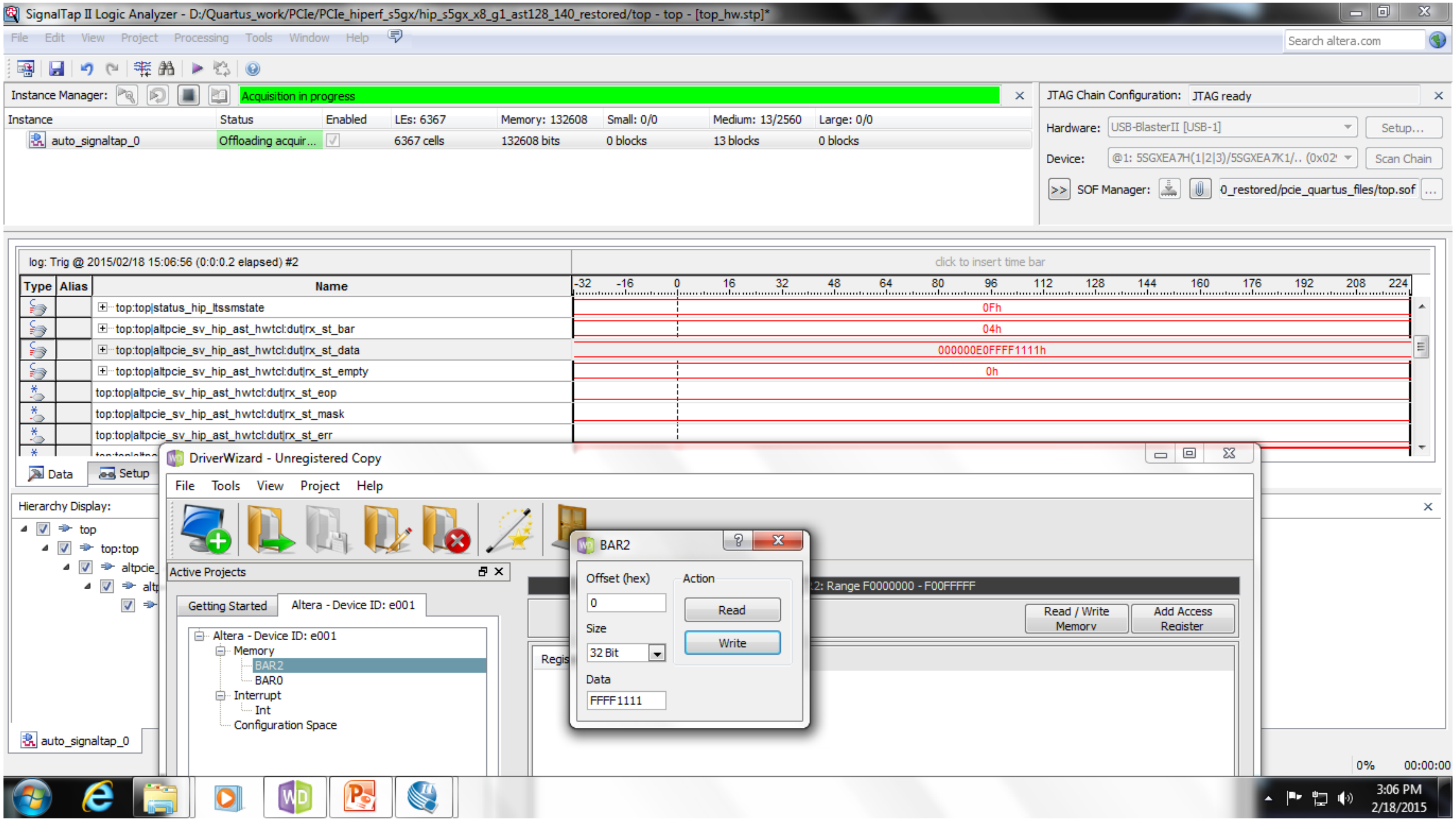


- Altera - Device ID: e001
  - Memory
    - BAR2
    - BAR0
  - Interrupt
    - Int
  - Configuration Space

	Interrupt Name	Interrupt Number	Type
1	Int	11	Message-Signaled Interrupts (MSI)

```
Accessing resource
Wrote to BAR0 at offset: 0x0 Value: 0x12
Read from BAR0 at offset: 0x0 Value: 0x12
Accessing resource
Wrote to BAR0 at offset: 0x10 Value: 0x100001
Read from BAR0 at offset: 0x10 Value: 0x100001
Wrote to BAR0 at offset: 0x10 Value: 0x101234
Read from BAR0 at offset: 0x10 Value: 0x101234
Wrote to BAR0 at offset: 0x10 Value: 0x1000000000101234
Read from BAR0 at offset: 0x10 Value: 0x1000000000101234
WDC_IntEnable() failed: error 0x20000023 ("Driver not installed")
Listen to Interrupts
Listen to Interrupts failed
```

Data Passed the protocol layer



SignalTap II Logic Analyzer - D:/Quartus_work/PCIe/PCIe_hiperf_s5gx/hip_s5gx_x8_g1_ast128_140_restored/top - top - [top_hw.stp]*

Instance Manager: Acquisition in progress

Instance	Status	Enabled	LEs: 6367	Memory: 132608	Small: 0/0	Medium: 13/2560	Large: 0/0
auto_signaltap_0	Offloading acquir...	☑	6367 cells	132608 bits	0 blocks	13 blocks	0 blocks

JTAG Chain Configuration: JTAG ready

Hardware: USB-BlasterII [USB-1]

Device: @1: 5SGXEA7H(1|2|3)/5SGXEA7K1/.. (0x02

SOF Manager: 0_restored/pcie_quartus_files/top.sof

log: Trig @ 2015/02/18 15:06:56 (0:0:0.2 elapsed) #2

Name	Value
top:top|status_hip_ltssmstate	0Fh
top:top|altpcie_sv_hip_ast_hwtcl:dut|rx_st_bar	04h
top:top|altpcie_sv_hip_ast_hwtcl:dut|rx_st_data	000000E0FFFF1111h
top:top|altpcie_sv_hip_ast_hwtcl:dut|rx_st_empty	0h
top:top|altpcie_sv_hip_ast_hwtcl:dut|rx_st_eop	
top:top|altpcie_sv_hip_ast_hwtcl:dut|rx_st_mask	
top:top|altpcie_sv_hip_ast_hwtcl:dut|rx_st_err	

DriverWizard - Unregistered Copy

Active Projects: Altera - Device ID: e001

- Altera - Device ID: e001
  - Memory
    - BAR2
    - BAR0
  - Interrupt
    - Int
  - Configuration Space

BAR2: Range F0000000 - F00FFFFF

BAR2

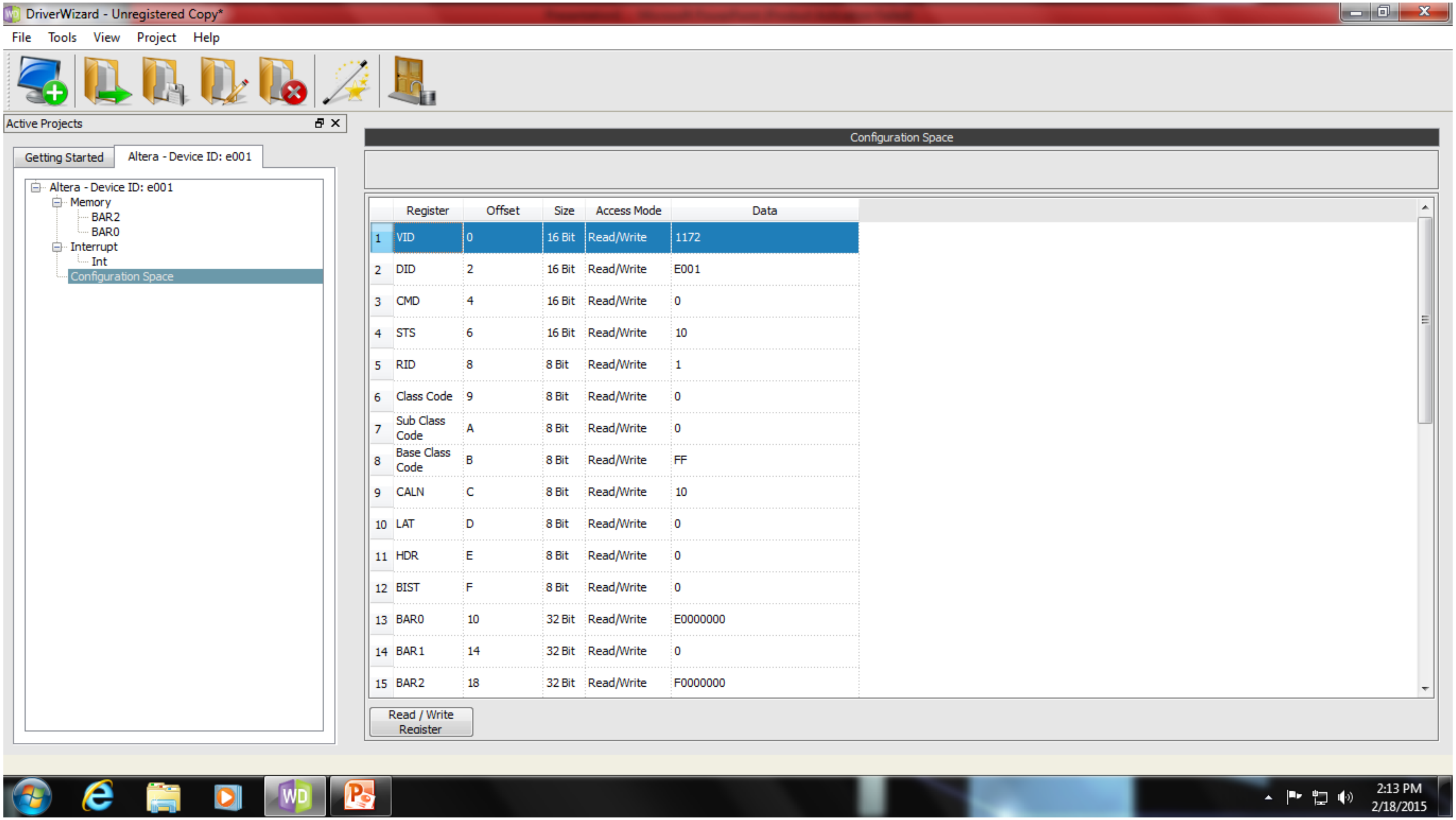
Offset (hex): 0

Size: 32 Bit

Data: FFFF1111

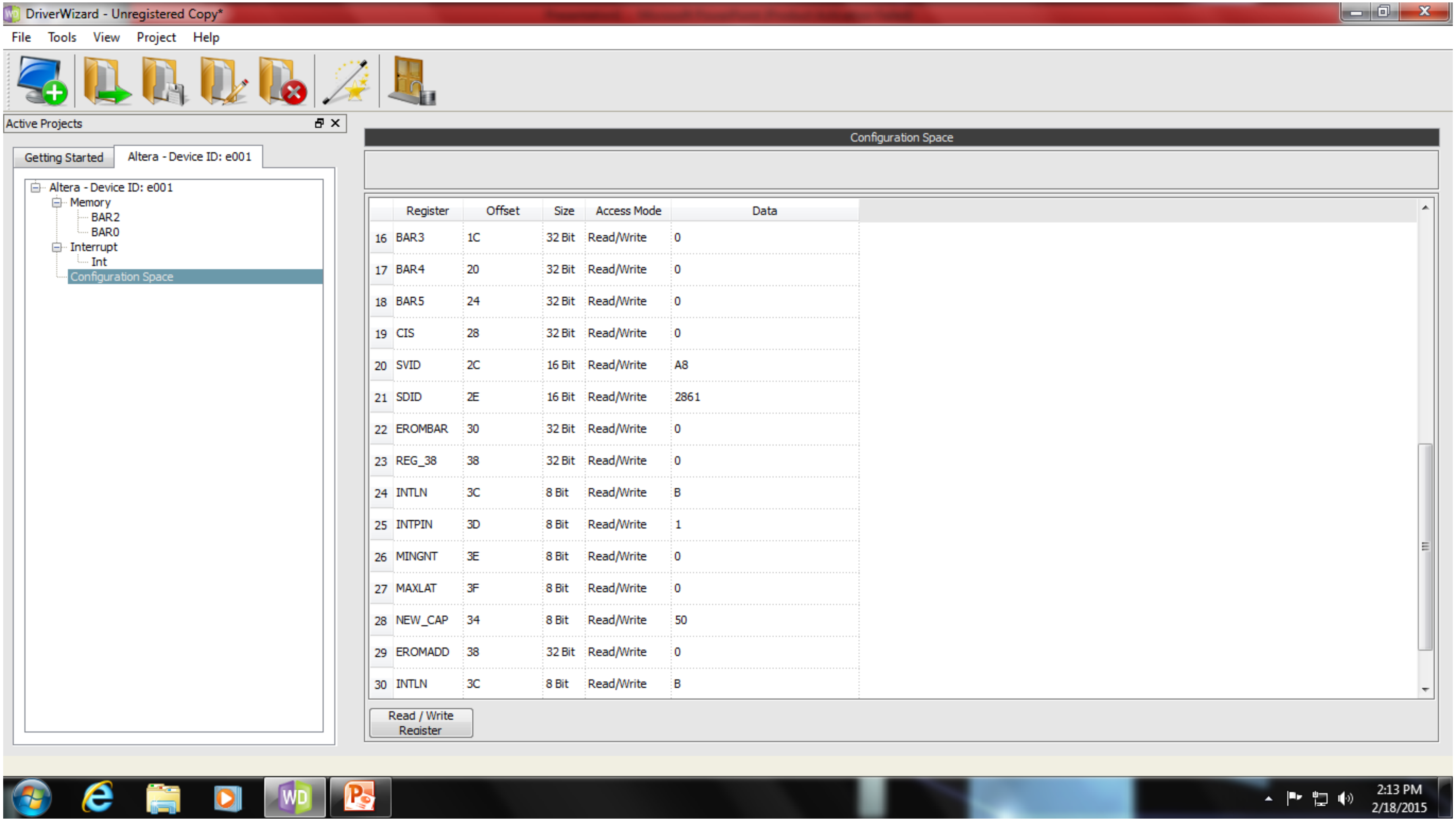
Gen2 x8 Avalon ST reference design1:Results

Configuration space header(registers)



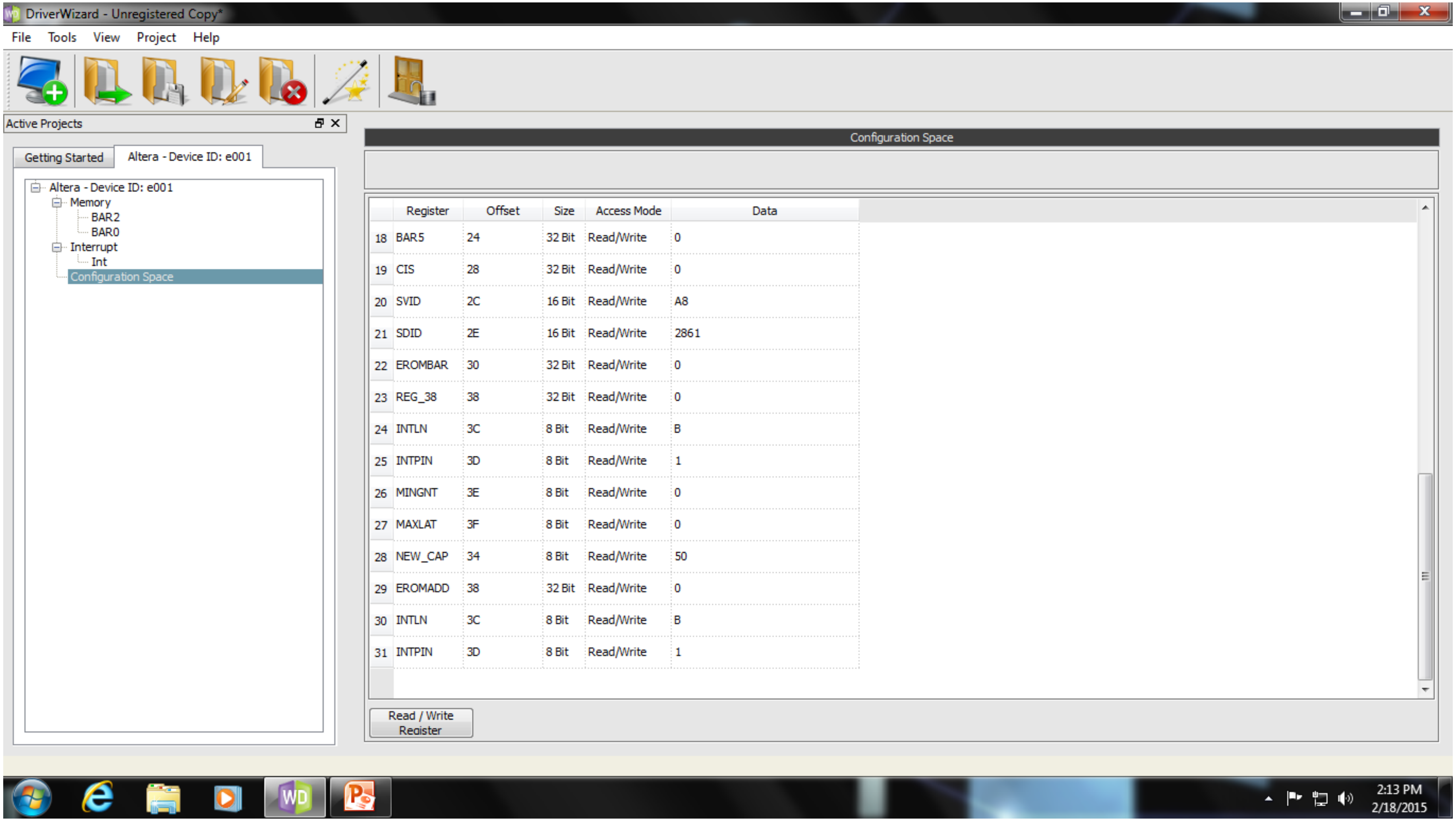
	Register	Offset	Size	Access Mode	Data
1	VID	0	16 Bit	Read/Write	1172
2	DID	2	16 Bit	Read/Write	E001
3	CMD	4	16 Bit	Read/Write	0
4	STS	6	16 Bit	Read/Write	10
5	RID	8	8 Bit	Read/Write	1
6	Class Code	9	8 Bit	Read/Write	0
7	Sub Class Code	A	8 Bit	Read/Write	0
8	Base Class Code	B	8 Bit	Read/Write	FF
9	CALN	C	8 Bit	Read/Write	10
10	LAT	D	8 Bit	Read/Write	0
11	HDR	E	8 Bit	Read/Write	0
12	BIST	F	8 Bit	Read/Write	0
13	BAR0	10	32 Bit	Read/Write	E0000000
14	BAR1	14	32 Bit	Read/Write	0
15	BAR2	18	32 Bit	Read/Write	F0000000

Configuration space header(registers)



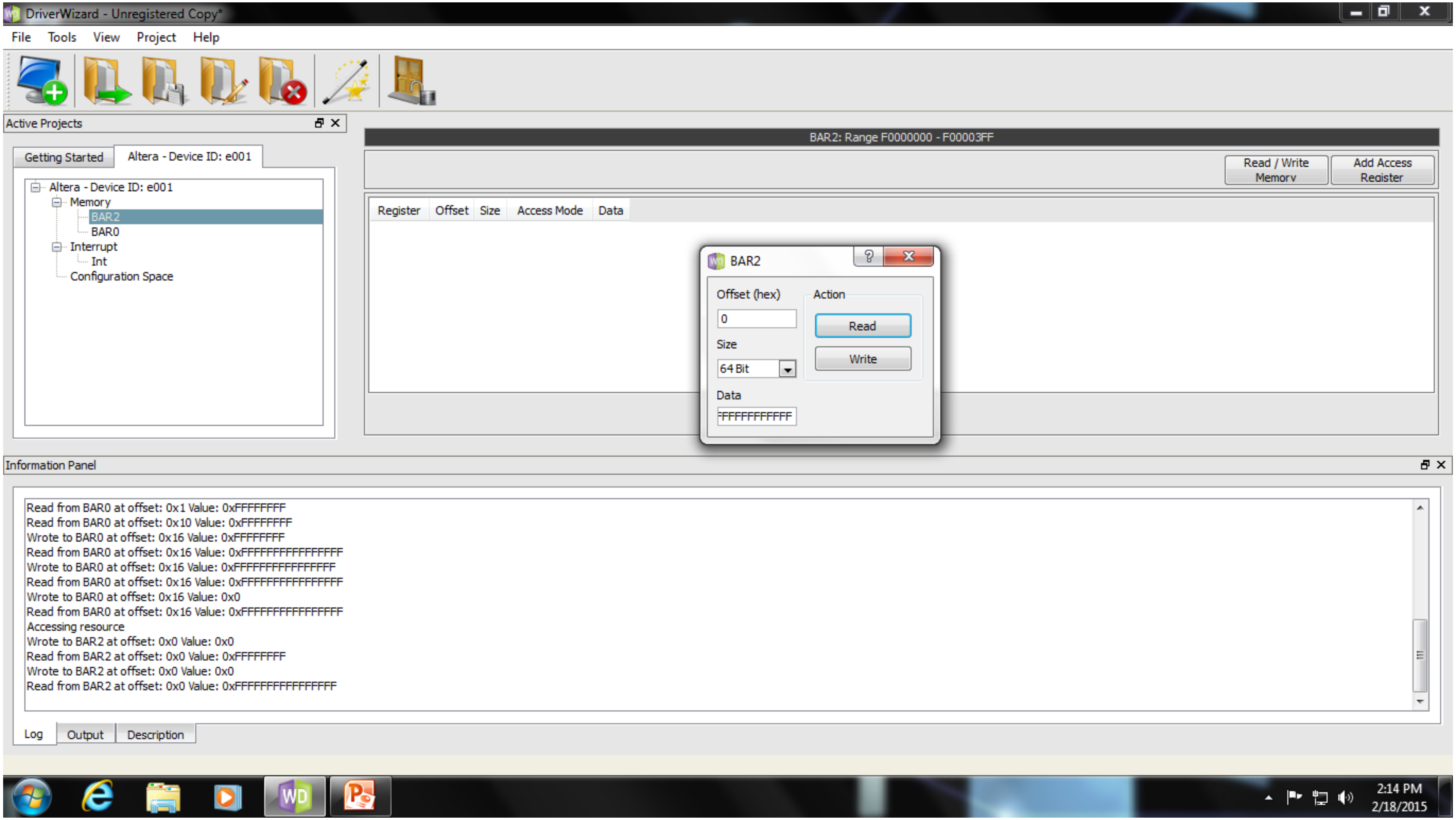
	Register	Offset	Size	Access Mode	Data
16	BAR3	1C	32 Bit	Read/Write	0
17	BAR4	20	32 Bit	Read/Write	0
18	BAR5	24	32 Bit	Read/Write	0
19	CIS	28	32 Bit	Read/Write	0
20	SVID	2C	16 Bit	Read/Write	A8
21	SDID	2E	16 Bit	Read/Write	2861
22	EROMBAR	30	32 Bit	Read/Write	0
23	REG_38	38	32 Bit	Read/Write	0
24	INTLN	3C	8 Bit	Read/Write	B
25	INTPIN	3D	8 Bit	Read/Write	1
26	MINGNT	3E	8 Bit	Read/Write	0
27	MAXLAT	3F	8 Bit	Read/Write	0
28	NEW_CAP	34	8 Bit	Read/Write	50
29	EROMADD	38	32 Bit	Read/Write	0
30	INTLN	3C	8 Bit	Read/Write	B

Configuration space header(registers)



	Register	Offset	Size	Access Mode	Data
18	BAR5	24	32 Bit	Read/Write	0
19	CIS	28	32 Bit	Read/Write	0
20	SVID	2C	16 Bit	Read/Write	A8
21	SDID	2E	16 Bit	Read/Write	2861
22	EROMBAR	30	32 Bit	Read/Write	0
23	REG_38	38	32 Bit	Read/Write	0
24	INTLN	3C	8 Bit	Read/Write	B
25	INTPIN	3D	8 Bit	Read/Write	1
26	MINGNT	3E	8 Bit	Read/Write	0
27	MAXLAT	3F	8 Bit	Read/Write	0
28	NEW_CAP	34	8 Bit	Read/Write	50
29	EROMADD	38	32 Bit	Read/Write	0
30	INTLN	3C	8 Bit	Read/Write	B
31	INTPIN	3D	8 Bit	Read/Write	1

Memory Read/Write(error)



BAR2: Range F0000000 - F00003FF

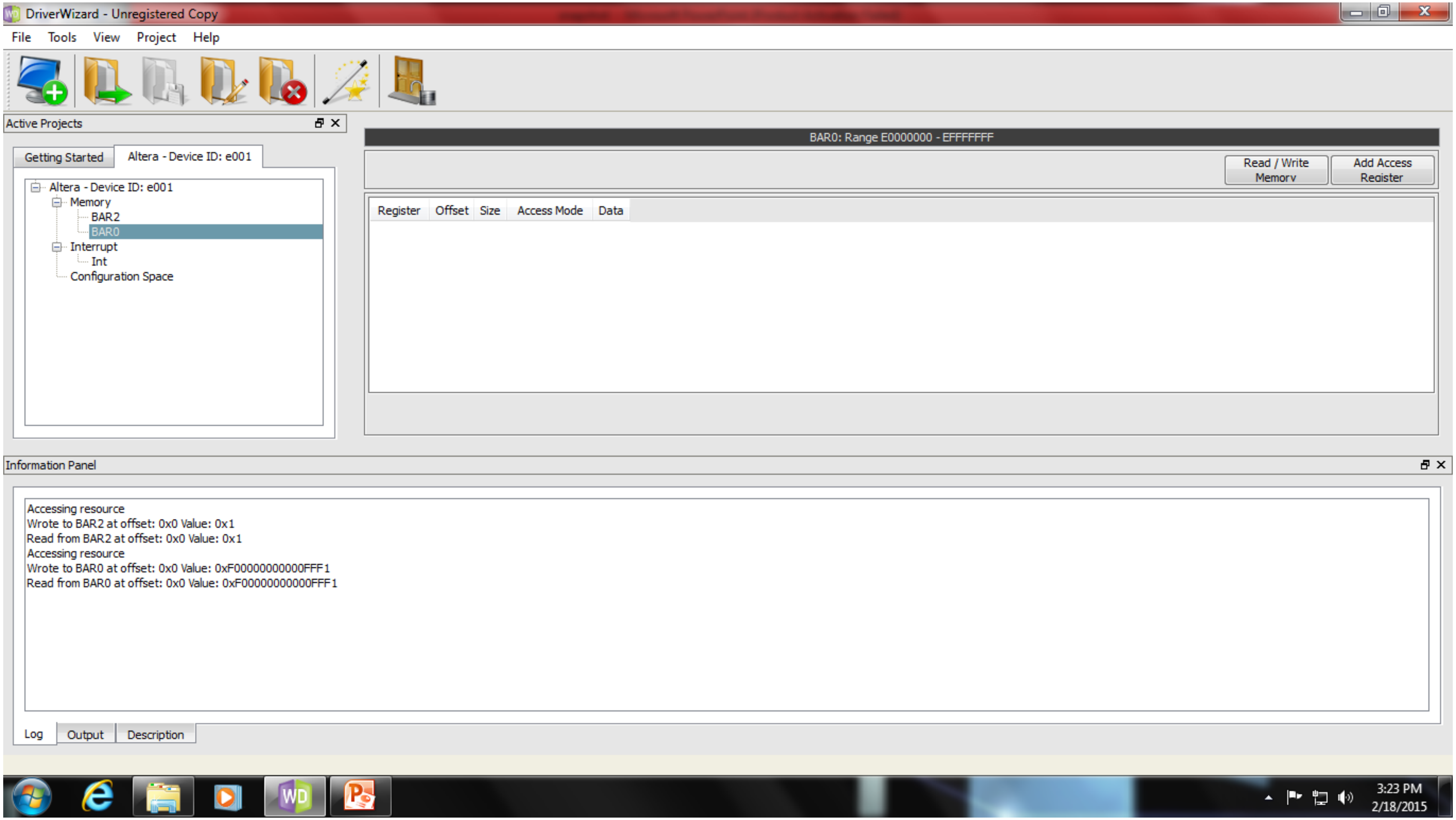
Offset (hex): 0

Size: 64 Bit

Data: FFFFFFFFFFF

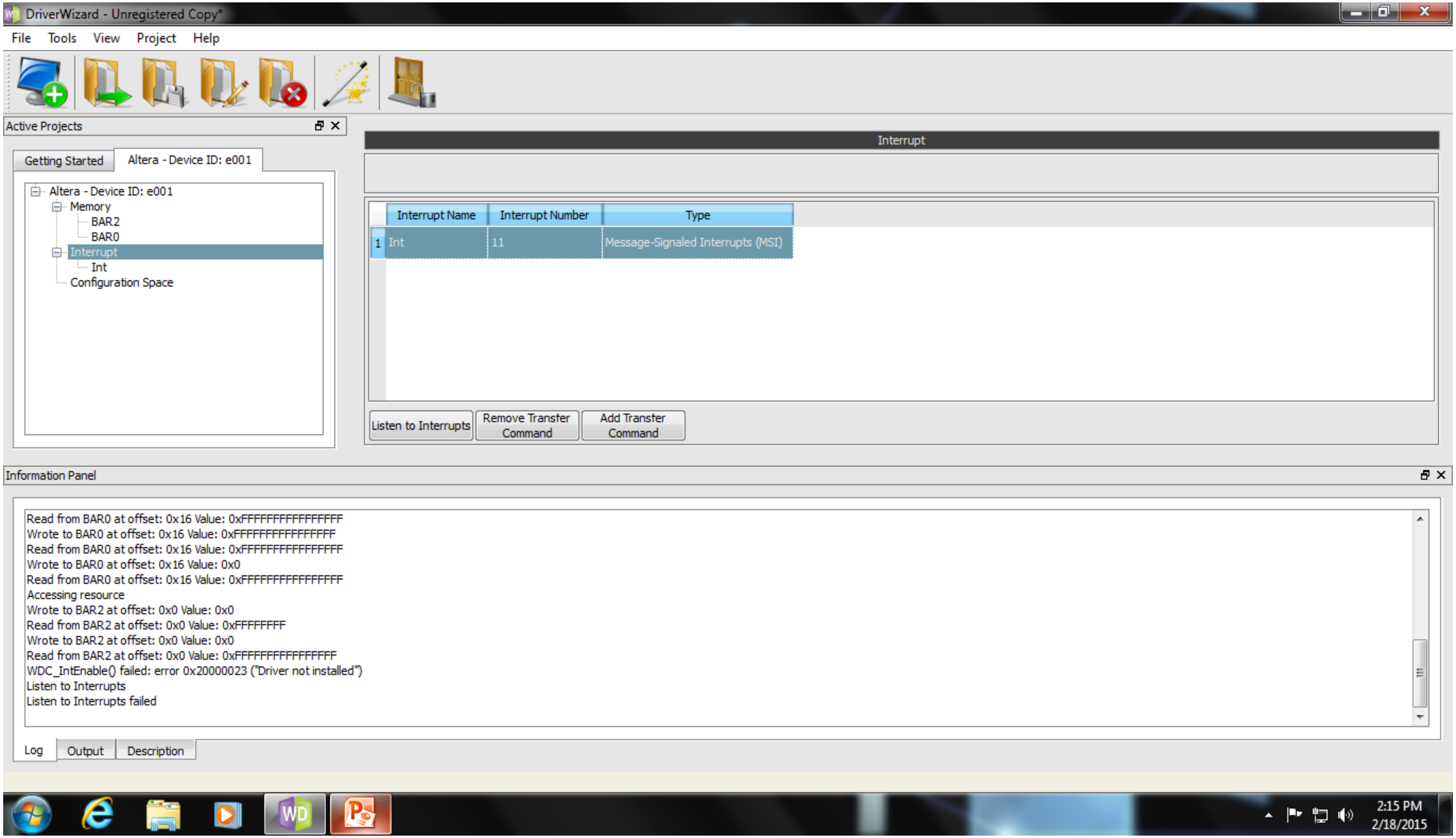
```
Read from BAR0 at offset: 0x1 Value: 0xFFFFFFFF
Read from BAR0 at offset: 0x10 Value: 0xFFFFFFFF
Wrote to BAR0 at offset: 0x16 Value: 0xFFFFFFFF
Read from BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Wrote to BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Read from BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Wrote to BAR0 at offset: 0x16 Value: 0x0
Read from BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Accessing resource
Wrote to BAR2 at offset: 0x0 Value: 0x0
Read from BAR2 at offset: 0x0 Value: 0xFFFFFFFF
Wrote to BAR2 at offset: 0x0 Value: 0x0
Read from BAR2 at offset: 0x0 Value: 0xFFFFFFFFFFFFFFFF
```

Memory Read/Write



```
Accessing resource
Wrote to BAR2 at offset: 0x0 Value: 0x1
Read from BAR2 at offset: 0x0 Value: 0x1
Accessing resource
Wrote to BAR0 at offset: 0x0 Value: 0xF00000000000FFF1
Read from BAR0 at offset: 0x0 Value: 0xF00000000000FFF1
```

Interrupt

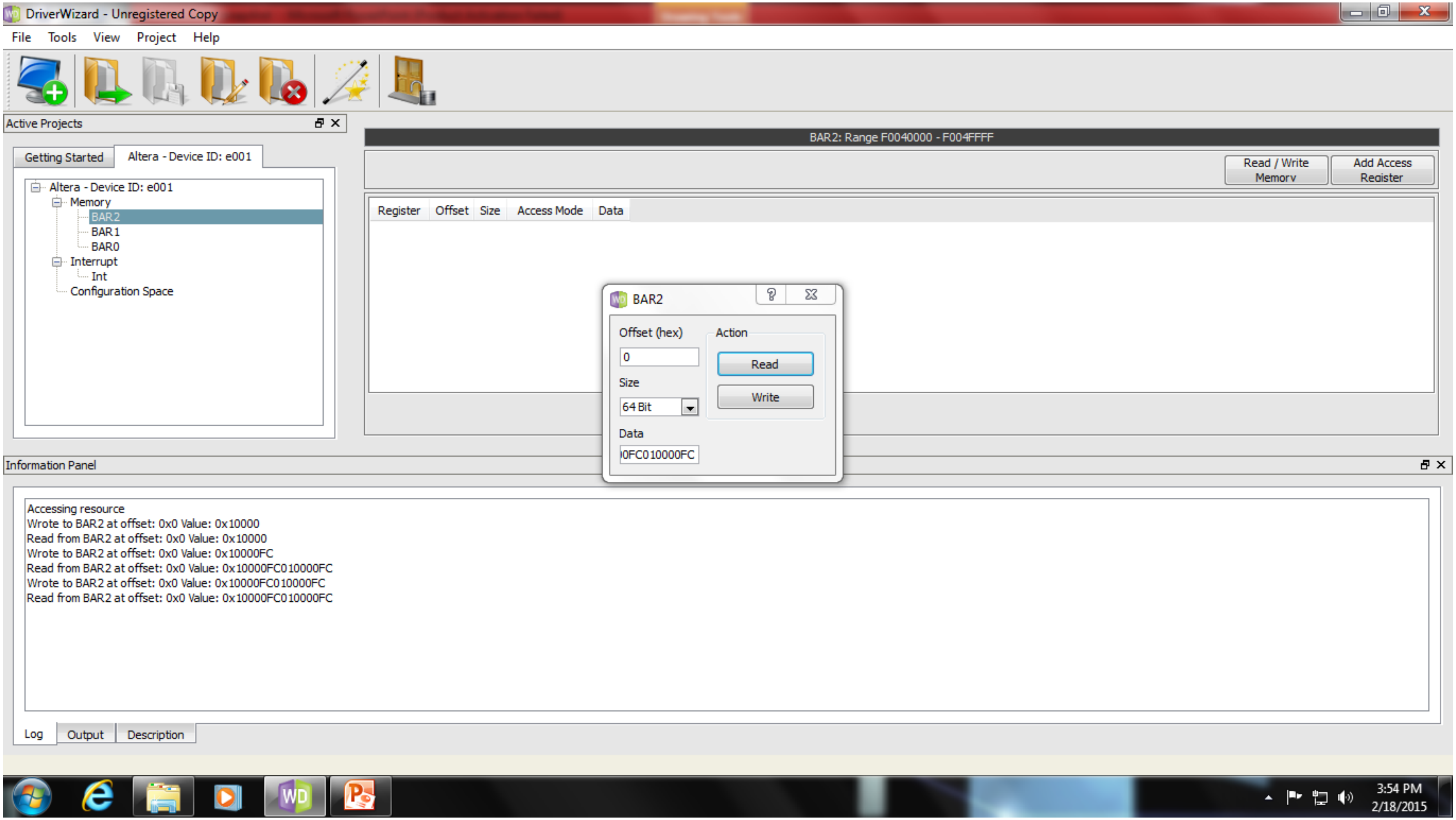


	Interrupt Name	Interrupt Number	Type
1	Int	11	Message-Signaled Interrupts (MSI)

```
Read from BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Wrote to BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Read from BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Wrote to BAR0 at offset: 0x16 Value: 0x0
Read from BAR0 at offset: 0x16 Value: 0xFFFFFFFFFFFFFFFF
Accessing resource
Wrote to BAR2 at offset: 0x0 Value: 0x0
Read from BAR2 at offset: 0x0 Value: 0xFFFFFFFF
Wrote to BAR2 at offset: 0x0 Value: 0x0
Read from BAR2 at offset: 0x0 Value: 0xFFFFFFFFFFFFFFFF
WDC_IntEnable() failed: error 0x20000023 ("Driver not installed")
Listen to Interrupts
Listen to Interrupts failed
```

Gen3 x4 Avalon ST reference design1:Results

Memory Read/Write



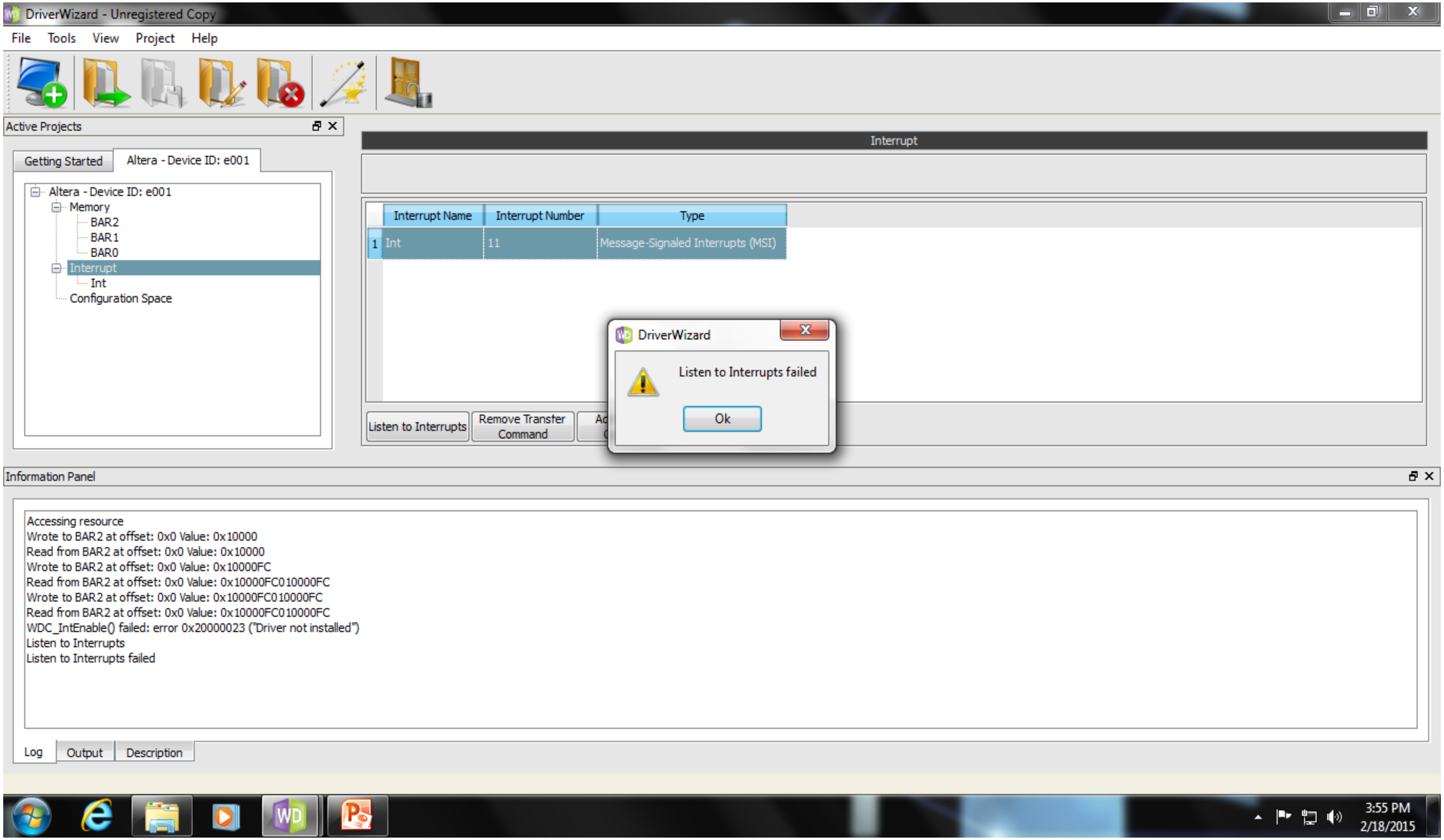
Offset (hex): 0

Size: 64 Bit

Data: 0FC010000FC

```
Accessing resource
Wrote to BAR2 at offset: 0x0 Value: 0x10000
Read from BAR2 at offset: 0x0 Value: 0x10000
Wrote to BAR2 at offset: 0x0 Value: 0x10000FC
Read from BAR2 at offset: 0x0 Value: 0x10000FC010000FC
Wrote to BAR2 at offset: 0x0 Value: 0x10000FC010000FC
Read from BAR2 at offset: 0x0 Value: 0x10000FC010000FC
```

Interrupt



	Interrupt Name	Interrupt Number	Type
1	Int	11	Message-Signaled Interrupts (MSI)

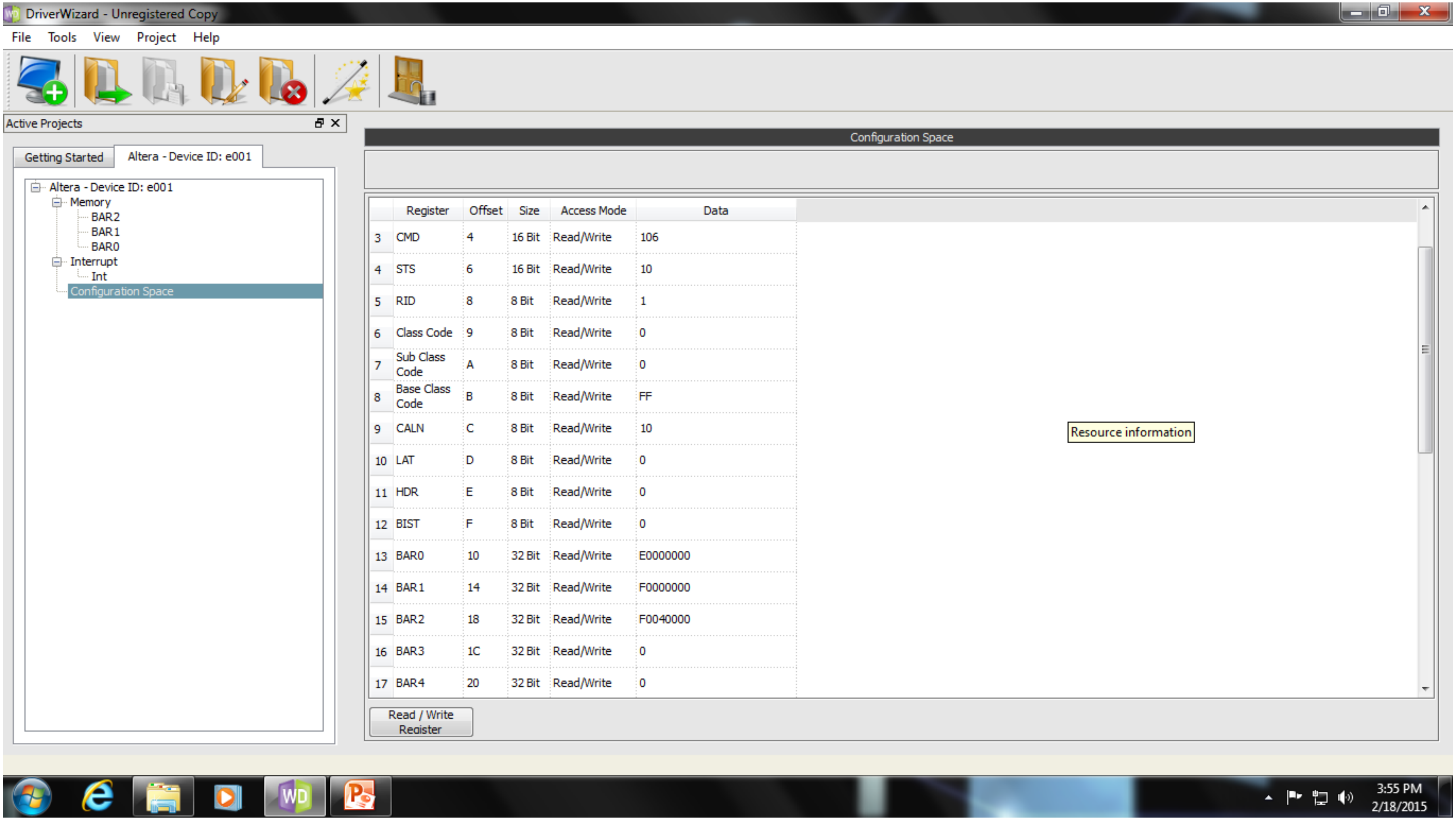
DriverWizard

Listen to Interrupts failed

Ok

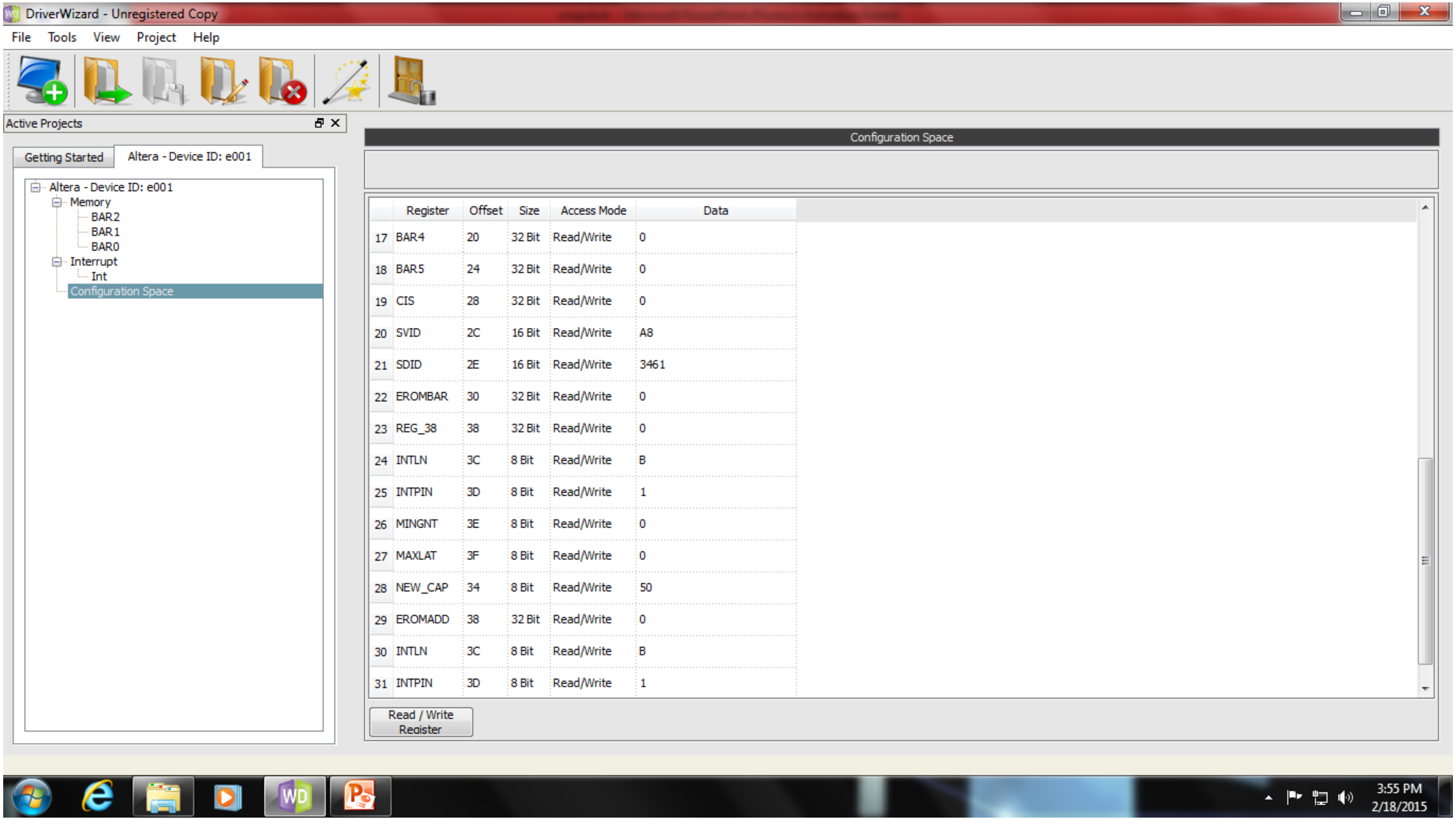
```
Accessing resource
Wrote to BAR2 at offset: 0x0 Value: 0x10000
Read from BAR2 at offset: 0x0 Value: 0x10000
Wrote to BAR2 at offset: 0x0 Value: 0x10000FC
Read from BAR2 at offset: 0x0 Value: 0x10000FC010000FC
Wrote to BAR2 at offset: 0x0 Value: 0x10000FC010000FC
Read from BAR2 at offset: 0x0 Value: 0x10000FC010000FC
WDC_IntEnable() failed: error 0x20000023 ("Driver not installed")
Listen to Interrupts
Listen to Interrupts failed
```

Configuration space header(registers)



	Register	Offset	Size	Access Mode	Data
3	CMD	4	16 Bit	Read/Write	106
4	STS	6	16 Bit	Read/Write	10
5	RID	8	8 Bit	Read/Write	1
6	Class Code	9	8 Bit	Read/Write	0
7	Sub Class Code	A	8 Bit	Read/Write	0
8	Base Class Code	B	8 Bit	Read/Write	FF
9	CALN	C	8 Bit	Read/Write	10
10	LAT	D	8 Bit	Read/Write	0
11	HDR	E	8 Bit	Read/Write	0
12	BIST	F	8 Bit	Read/Write	0
13	BAR0	10	32 Bit	Read/Write	E0000000
14	BAR1	14	32 Bit	Read/Write	F0000000
15	BAR2	18	32 Bit	Read/Write	F0040000
16	BAR3	1C	32 Bit	Read/Write	0
17	BAR4	20	32 Bit	Read/Write	0

Configuration space header(registers)



	Register	Offset	Size	Access Mode	Data
17	BAR4	20	32 Bit	Read/Write	0
18	BAR5	24	32 Bit	Read/Write	0
19	CIS	28	32 Bit	Read/Write	0
20	SVID	2C	16 Bit	Read/Write	A8
21	SDID	2E	16 Bit	Read/Write	3461
22	EROMBAR	30	32 Bit	Read/Write	0
23	REG_38	38	32 Bit	Read/Write	0
24	INTLN	3C	8 Bit	Read/Write	B
25	INTPIN	3D	8 Bit	Read/Write	1
26	MINGNT	3E	8 Bit	Read/Write	0
27	MAXLAT	3F	8 Bit	Read/Write	0
28	NEW_CAP	34	8 Bit	Read/Write	50
29	EROMADD	38	32 Bit	Read/Write	0
30	INTLN	3C	8 Bit	Read/Write	B
31	INTPIN	3D	8 Bit	Read/Write	1

Results

Result of reference design1 using Jungo app(see jungo_app_results section)

	Memory RD/WR(32/64 bit)	**Interrupt**	**Configuration Header**
Gen1 X8	Yes, both 32 and 64 bit RD &WR successful	Listen to Interrupt failed	Shows the header
Gen2 X8	Yes, both 32 and 64 bit RD &WR successful	Listen to Interrupt failed	Shows the header
Gen3 X4	Yes, both 32 and 64 bit RD &WR successful	Listen to Interrupt failed	Shows the header

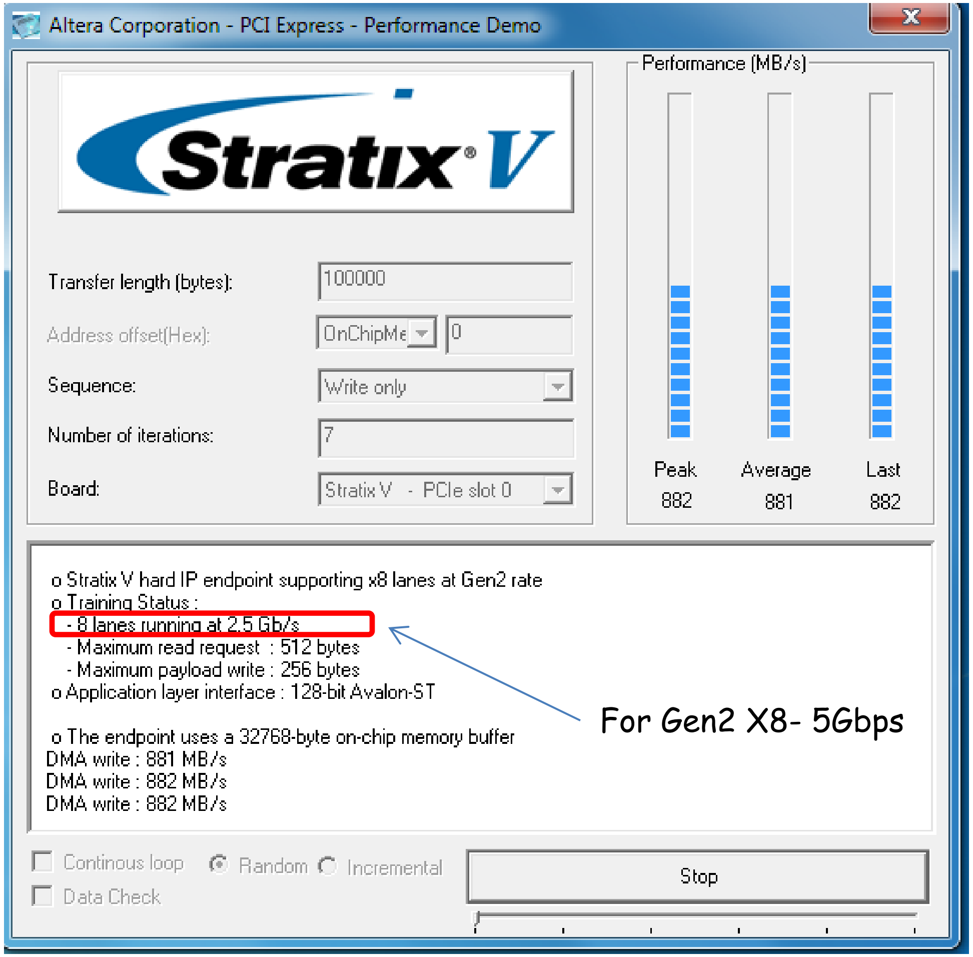
1. We have tested the memory read and write functionality(32 bit and 64 bit) for each variation .Memory read and write is successful except in case of Gen2 X8,where initially there are wrong memory read and write. The problem resolved when we have re-tested (done everything from step1) it later.

2. Listen to interrupt failed for each variation. **Evaluating interrupt**….
   --**Implement MSI-X option is off in Capabilities register > MSI-X Capabilities**

3. Show the Configuration space register (header) for each variation.

PCIE DMA APPLICATION

Design summary and Link training status



Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Sequence: Write only

Number of iterations: 7

Board: Stratix V - PCIe slot 0

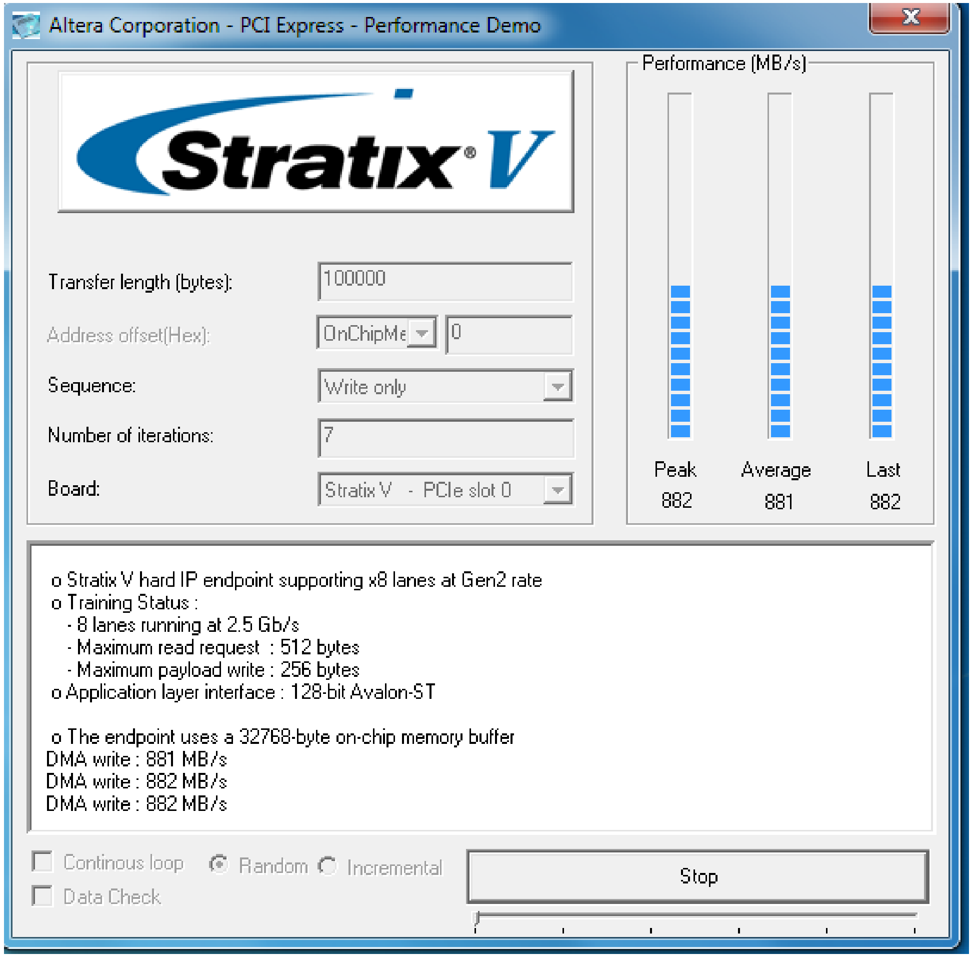
```
o Stratix V hard IP endpoint supporting x8 lanes at Gen2 rate
o Training Status :
  - 8 lanes running at 2.5 Gb/s
  - Maximum read request  : 512 bytes
  - Maximum payload write : 256 bytes
o Application layer interface : 128-bit Avalon-ST

o The endpoint uses a 32768-byte on-chip memory buffer
DMA write : 881 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
```

□ Continous loop ● Random ○ Incremental

□ Data Check

For Gen2 X8- 5Gbps

Write only/7 iterations/transfer length=100000 bytes
look the performance



Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Sequence: Write only

Number of iterations: 7

Board: Stratix V - PCIe slot 0

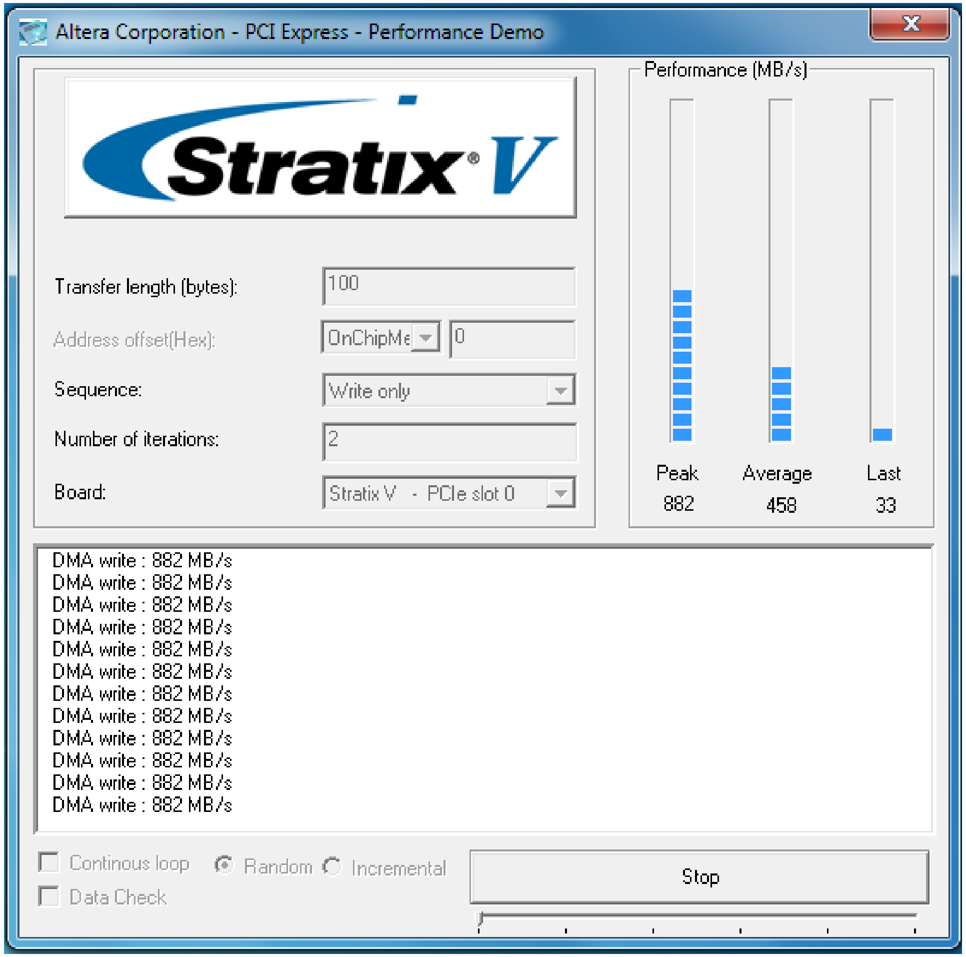
Peak	Average	Last
882	881	882

```
o Stratix V hard IP endpoint supporting x8 lanes at Gen2 rate
o Training Status :
  - 8 lanes running at 2.5 Gb/s
  - Maximum read request  : 512 bytes
  - Maximum payload write : 256 bytes
o Application layer interface : 128-bit Avalon-ST

o The endpoint uses a 32768-byte on-chip memory buffer
DMA write : 881 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
```

□ Continous loop ● Random ○ Incremental

□ Data Check

Write only/2 iterations/transfer length=100 bytes
look the performance**



Transfer length (bytes): 100

Address offset(Hex): OnChipMe 0

Sequence: Write only

Number of iterations: 2

Board: Stratix V - PCIe slot 0

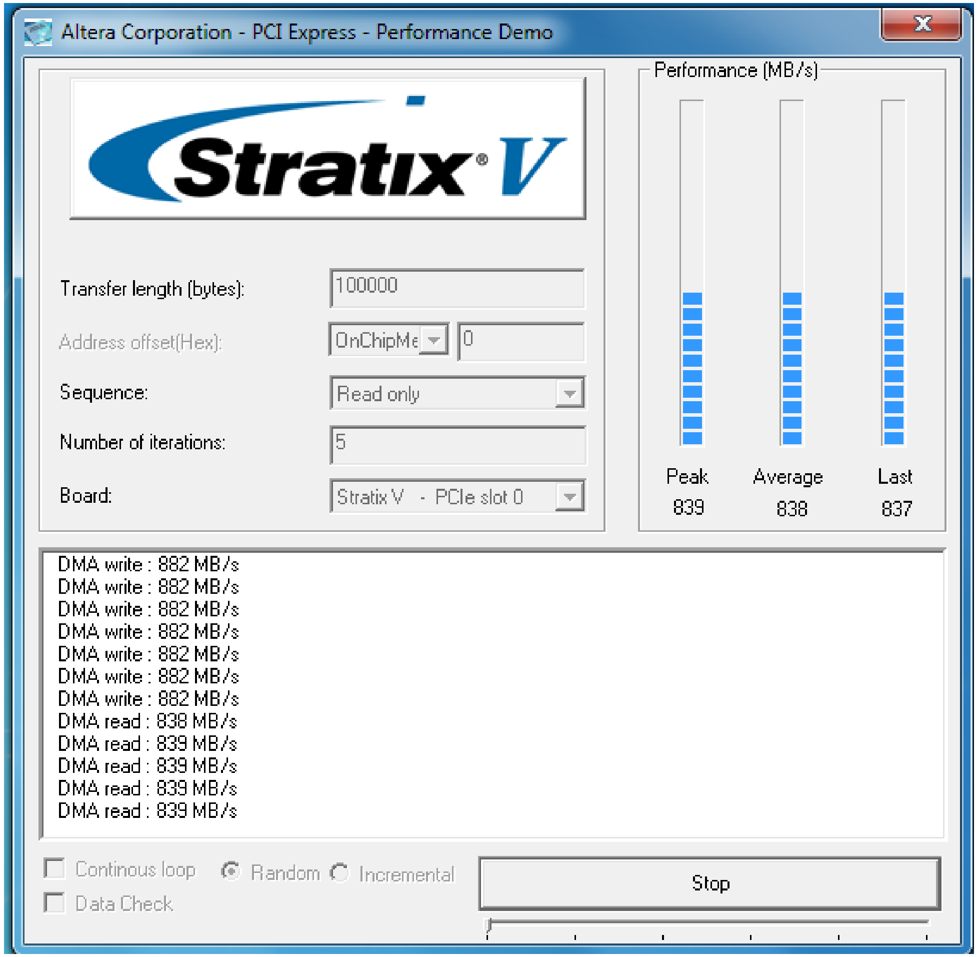
Peak 882 Average 458 Last 33

```
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
```

□ Continous loop ● Random ○ Incremental

□ Data Check

Read only/5 iterations/transfer length=100000 bytes
look the performance



Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Sequence: Read only

Number of iterations: 5

Board: Stratix V - PCIe slot 0

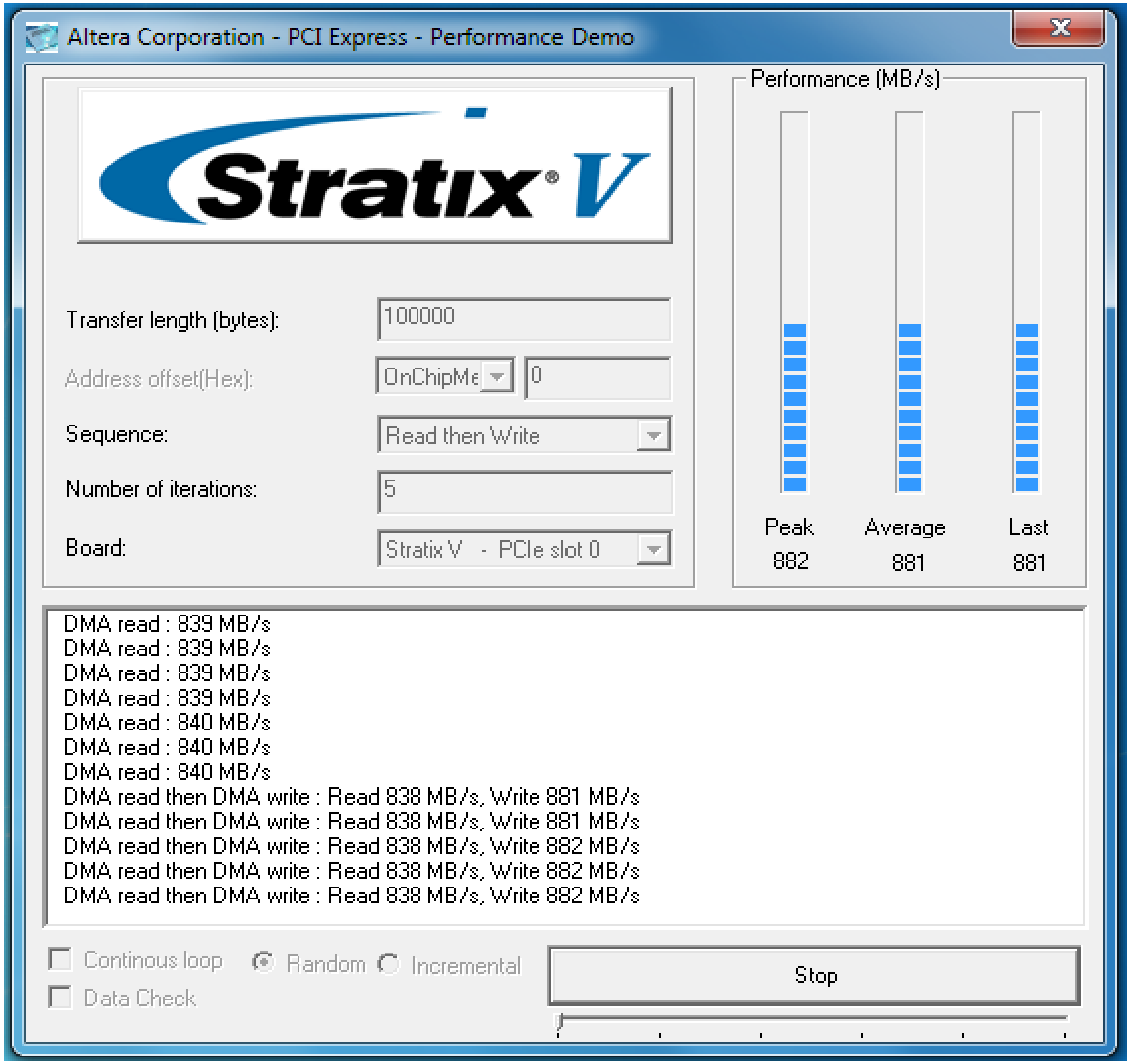
Peak	Average	Last
839	838	837

```
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA write : 882 MB/s
DMA read : 838 MB/s
DMA read : 839 MB/s
DMA read : 839 MB/s
DMA read : 839 MB/s
DMA read : 839 MB/s
```

□ Continous loop ● Random ○ Incremental

□ Data Check

Read then write/5 iterations/transfer length=100000 bytes
look the performance



Altera Corporation - PCI Express - Performance Demo

Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Sequence: Read then Write

Number of iterations: 5

Board: Stratix V - PCIe slot 0

Performance (MB/s)

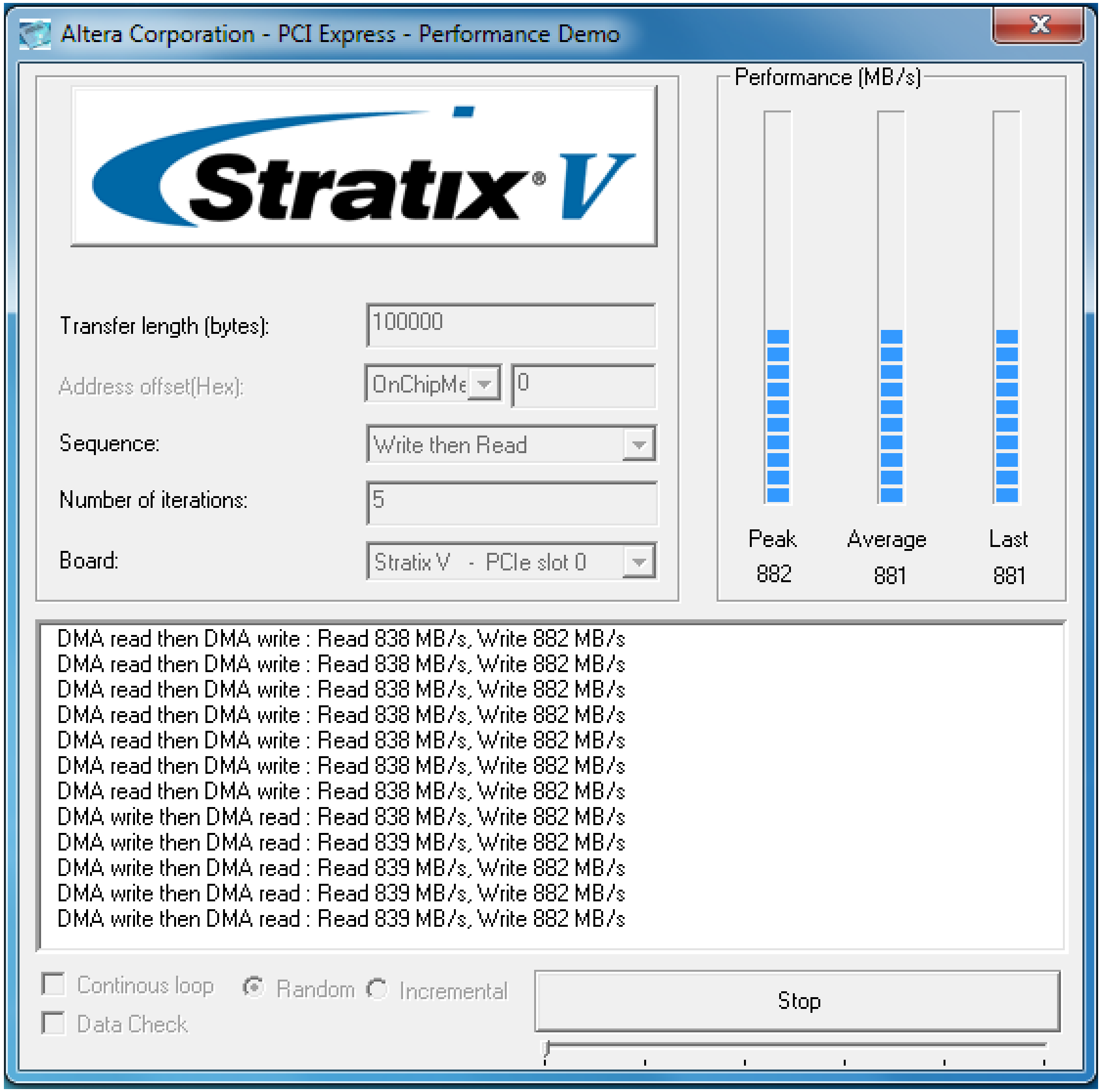
Peak	Average	Last
882	881	881

```
DMA read : 839 MB/s
DMA read : 839 MB/s
DMA read : 839 MB/s
DMA read : 839 MB/s
DMA read : 840 MB/s
DMA read : 840 MB/s
DMA read : 840 MB/s
DMA read then DMA write : Read 838 MB/s, Write 881 MB/s
DMA read then DMA write : Read 838 MB/s, Write 881 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
```

□ Continous loop ● Random ○ Incremental

□ Data Check

Write then Read/5 iterations/transfer length=100000 bytes
look the performance



Altera Corporation - PCI Express - Performance Demo

Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Sequence: Write then Read

Number of iterations: 5

Board: Stratix V - PCIe slot 0

Performance (MB/s)

Peak 882

Average 881

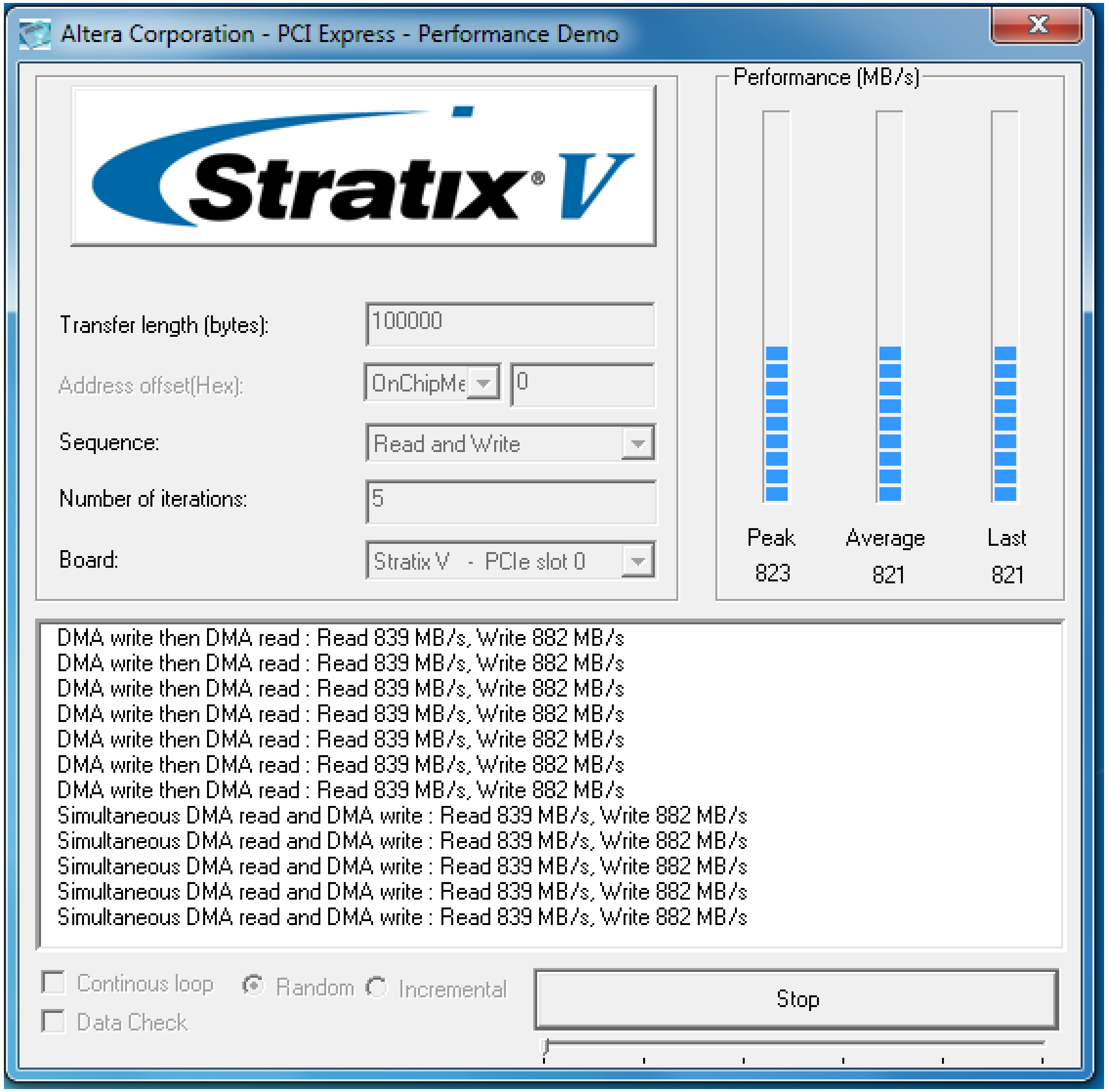
Last 881

```
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA read then DMA write : Read 838 MB/s, Write 882 MB/s
DMA write then DMA read : Read 838 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
```

☐ Continous loop ● Random ○ Incremental

☐ Data Check

Read and write /5 iterations/transfer length=100000 bytes
look the performance



Altera Corporation - PCI Express - Performance Demo

Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Sequence: Read and Write

Number of iterations: 5

Board: Stratix V - PCIe slot 0

Performance (MB/s)

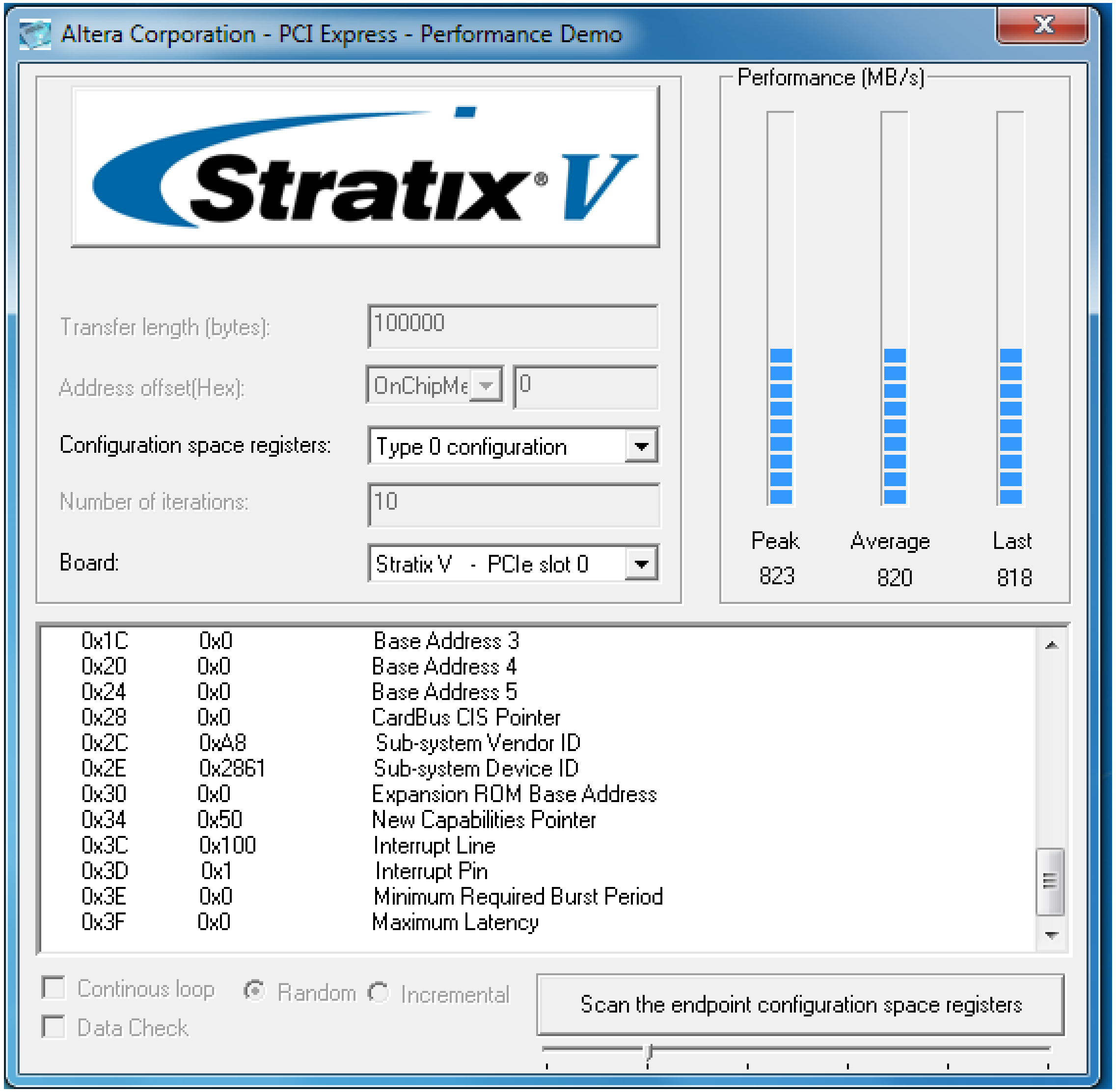
Peak	Average	Last
823	821	821

```
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
DMA write then DMA read : Read 839 MB/s, Write 882 MB/s
Simultaneous DMA read and DMA write : Read 839 MB/s, Write 882 MB/s
Simultaneous DMA read and DMA write : Read 839 MB/s, Write 882 MB/s
Simultaneous DMA read and DMA write : Read 839 MB/s, Write 882 MB/s
Simultaneous DMA read and DMA write : Read 839 MB/s, Write 882 MB/s
Simultaneous DMA read and DMA write : Read 839 MB/s, Write 882 MB/s
```

☐ Continous loop ● Random ○ Incremental

☐ Data Check

EP configuration space registers



Altera Corporation - PCI Express - Performance Demo

Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Configuration space registers: Type 0 configuration

Number of iterations: 10

Board: Stratix V - PCIe slot 0

Performance (MB/s)

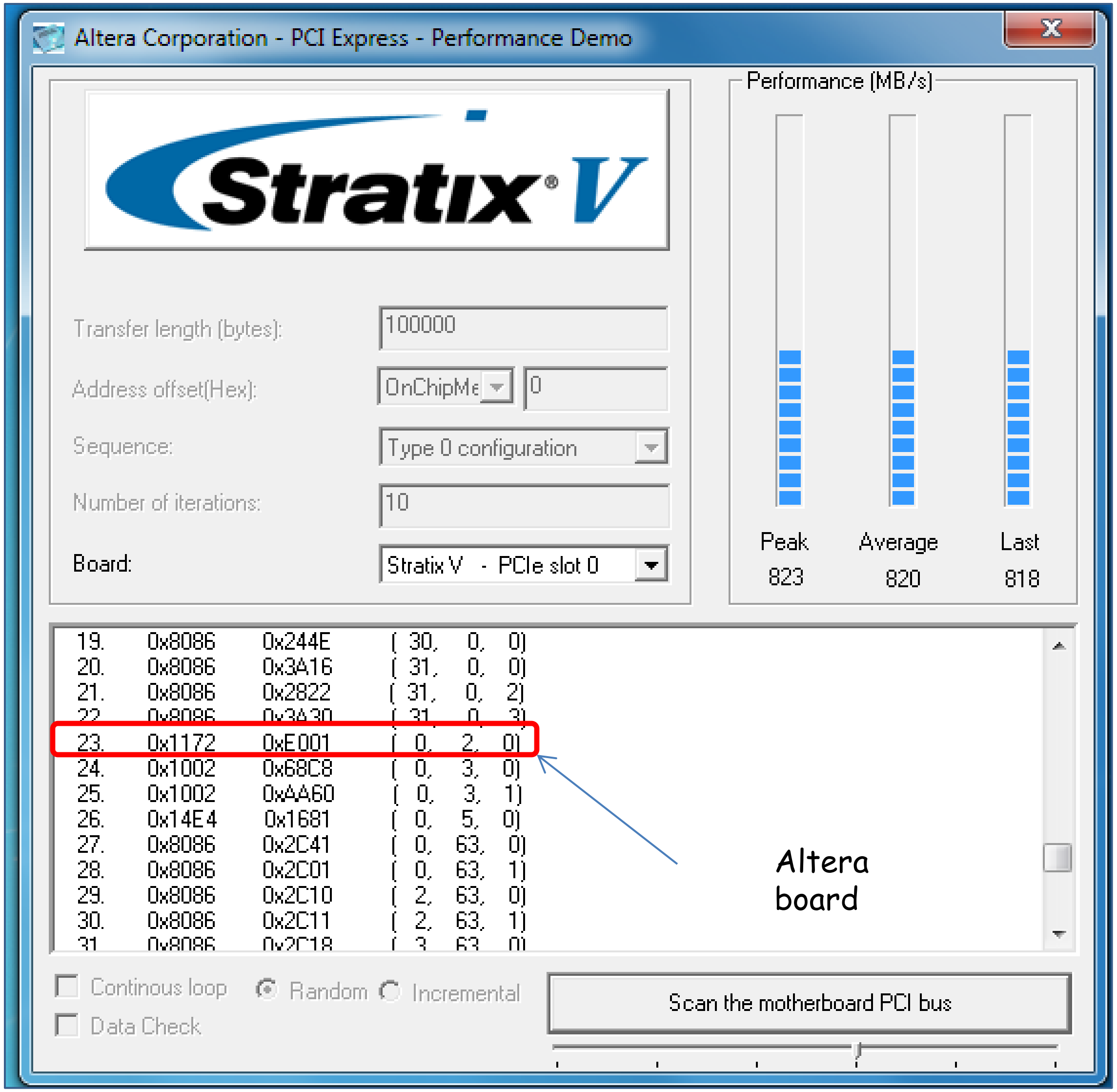
Peak	Average	Last
823	820	818

```
0x1C    0x0        Base Address 3
0x20    0x0        Base Address 4
0x24    0x0        Base Address 5
0x28    0x0        CardBus CIS Pointer
0x2C    0xA8       Sub-system Vendor ID
0x2E    0x2861     Sub-system Device ID
0x30    0x0        Expansion ROM Base Address
0x34    0x50       New Capabilities Pointer
0x3C    0x100      Interrupt Line
0x3D    0x1        Interrupt Pin
0x3E    0x0        Minimum Required Burst Period
0x3F    0x0        Maximum Latency
```

□ Continous loop ● Random ○ Incremental

□ Data Check

Scan the Motherboard PCI BUS



Altera Corporation - PCI Express - Performance Demo

Transfer length (bytes): 100000

Address offset(Hex): OnChipMe 0

Sequence: Type 0 configuration

Number of iterations: 10

Board: Stratix V - PCIe slot 0

Performance (MB/s)

Peak	Average	Last
823	820	818

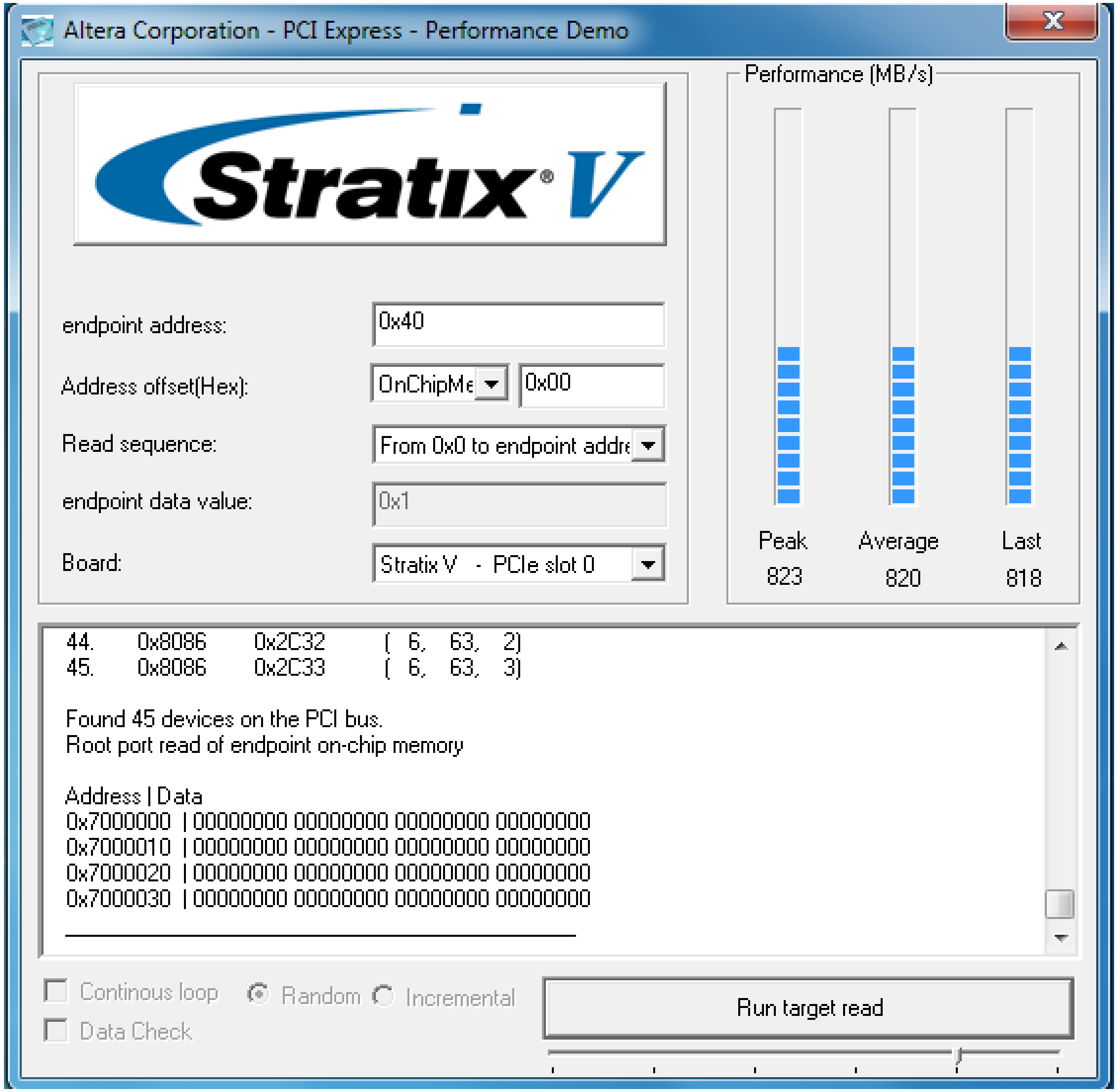
```
19.   0x8086   0x244E   ( 30,  0,  0)
20.   0x8086   0x3A16   ( 31,  0,  0)
21.   0x8086   0x2822   ( 31,  0,  2)
22.   0x8086   0x3A30   ( 31,  0,  3)
23.   0x1172   0xE001   (  0,  2,  0)
24.   0x1002   0x68C8   (  0,  3,  0)
25.   0x1002   0xAA60   (  0,  3,  1)
26.   0x14E4   0x1681   (  0,  5,  0)
27.   0x8086   0x2C41   (  0, 63,  0)
28.   0x8086   0x2C01   (  0, 63,  1)
29.   0x8086   0x2C10   (  2, 63,  0)
30.   0x8086   0x2C11   (  2, 63,  1)
31    0x8086   0x2C18   (  3, 63,  0)
```

Altera board

□ Continous loop ● Random ○ Incremental

□ Data Check

EP memory write



endpoint address: 0x40

Address offset(Hex): OnChipMe 0x00

Read sequence: From 0x0 to endpoint addre

endpoint data value: 0x1

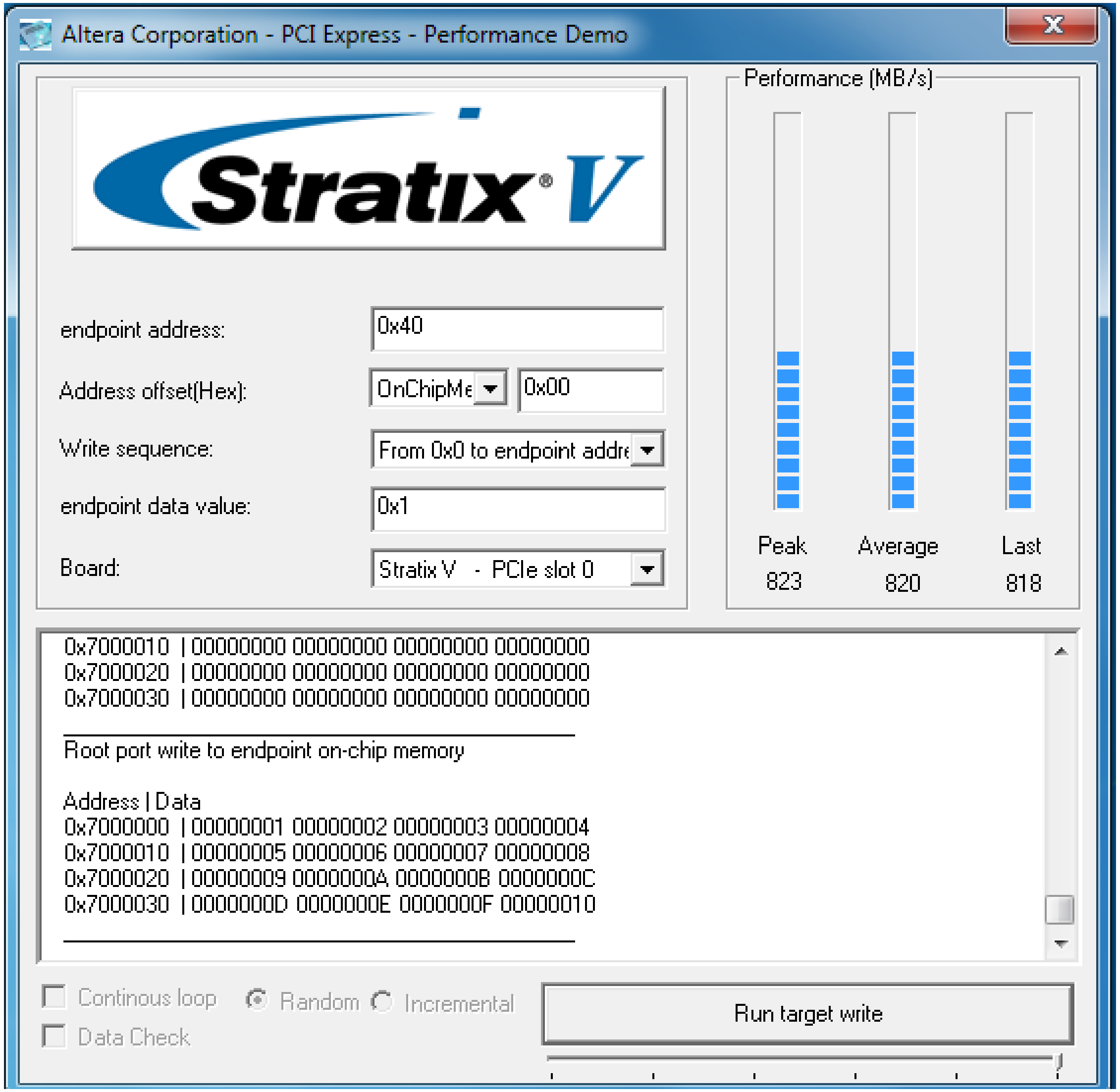
Board: Stratix V - PCIe slot 0

```
 44.     0x8086      0x2C32       (  6,   63,   2)
 45.     0x8086      0x2C33       (  6,   63,   3)

Found 45 devices on the PCI bus.
Root port read of endpoint on-chip memory

Address | Data
0x7000000  | 00000000 00000000 00000000 00000000
0x7000010  | 00000000 00000000 00000000 00000000
0x7000020  | 00000000 00000000 00000000 00000000
0x7000030  | 00000000 00000000 00000000 00000000
```

☐ Continous loop ● Random ○ Incremental

☐ Data Check

EP memory write



endpoint address: 0x40

Address offset(Hex): OnChipMe 0x00

Write sequence: From 0x0 to endpoint addre

endpoint data value: 0x1

Board: Stratix V - PCIe slot 0

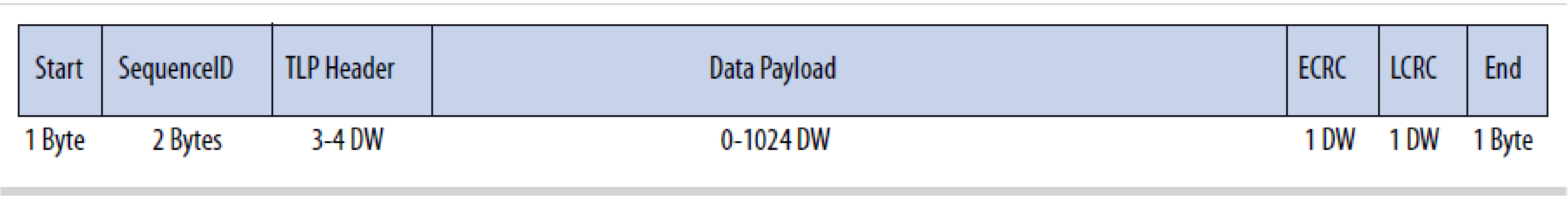
```
0x7000010  | 00000000 00000000 00000000 00000000
0x7000020  | 00000000 00000000 00000000 00000000
0x7000030  | 00000000 00000000 00000000 00000000

________________________________________
Root port write to endpoint on-chip memory

Address | Data
0x7000000  | 00000001 00000002 00000003 00000004
0x7000010  | 00000005 00000006 00000007 00000008
0x7000020  | 00000009 0000000A 0000000B 0000000C
0x7000030  | 0000000D 0000000E 0000000F 00000010

________________________________________
```

Results

Result of reference design1 (gen2 X8) using DMA app

1. Performance is same for write only, read only, write then read, read then write ,write and read
2. Performance degraded when the transfer length reduces from 100000 byte to 100 bytes
   --- due to the increasing ratio of header versus payload data and partly filled PCIe packets
3. App fetches board settings, PCI bus of motherboard ,EP configuration space register information
4. App reads and writes to the target memory.
5. Link training status gives some other indication.

Figure 1: TLP Format



Start	SequenceID	TLP Header	Data Payload	ECRC	LCRC	End
1 Byte	2 Bytes	3-4 DW	0-1024 DW	1 DW	1 DW	1 Byte

The theoretical maximum throughput is calculated using the following formula:

Throughput % = payload size / (payload size + overhead)

For a 256-byte maximum payload size and a three dword TLP header (or five dword overhead), the maximum possible throughput is (256/(256+20)), or 92%.

Benchmark results 1

- The following tables list the performance of x1, x4, and x8 operations with the Stratix V GX FPGA development board for the Intel i7-3930K 3.8 GHz Sandy Bridge-E processor using reference design1.
- The table shows the average throughput with the following parameters:
  - 100 KByte transfer
  - 20 iterations
  - A 256-byte payload
  - Maximum 512-byte read request
  - 256-byte read completion

In CRU,input data stream for single DMA ch is 3200 MB/s

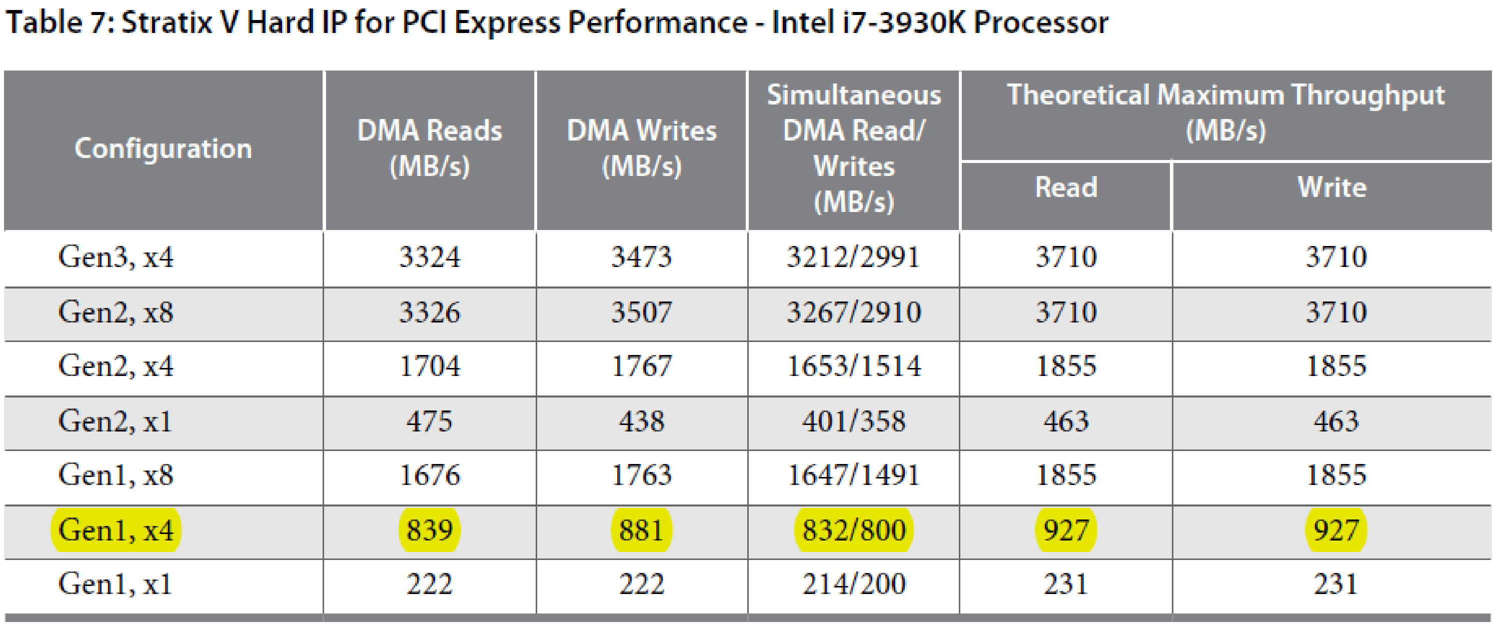


Table 7: Stratix V Hard IP for PCI Express Performance - Intel i7-3930K Processor

Configuration	DMA Reads (MB/s)	DMA Writes (MB/s)	Simultaneous DMA Read/ Writes (MB/s)	Theoretical Maximum Throughput (MB/s)	
				Read	Write
Gen3, x4	3324	3473	3212/2991	3710	3710
Gen2, x8	3326	3507	3267/2910	3710	3710
Gen2, x4	1704	1767	1653/1514	1855	1855
Gen2, x1	475	438	401/358	463	463
Gen1, x8	1676	1763	1647/1491	1855	1855
Gen1, x4	839	881	832/800	927	927
Gen1, x1	222	222	214/200	231	231

DMA results of RUN2
(Thanks to Budapest Team)

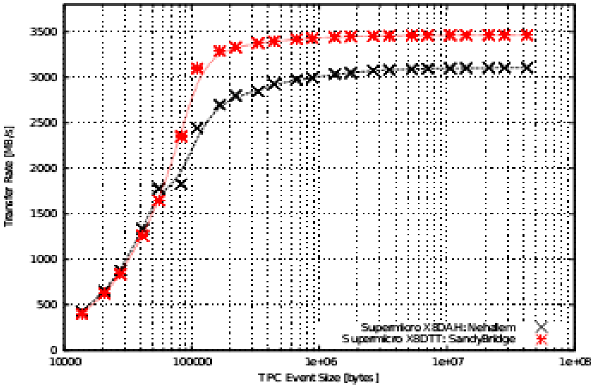


Figure 3. PCIe payload throughput of C-RORC to host RAM transfer as a sum of 12 independent DMA channels for a range of TPC event sizes tested on two different machines.

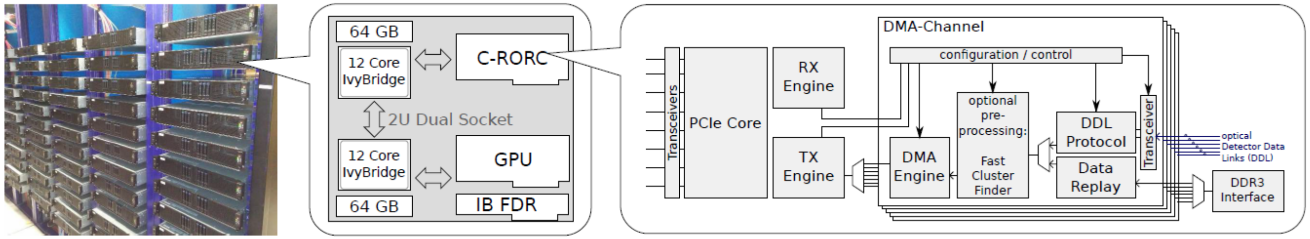


Figure 5: C-RORC Installation in the ALICE HLT for Run 2 and schematic drawing of the dataflow in the firmware.

--SandyBridge architecture having the PCIe endpoint implemented directly in the CPU against a Ne-halem architecture where PCIe is connected using an IO-Hub.

--Both architectures show decreased transfer rates for small event sizes due to the increased overhead into the ReportBuffer, the increas-ing ratio of header versus payload data and only partly filled PCIe packets.

--is sum equal to 12*DMA transfer rate of one ch ?
--independent chs ??

PCIE & CRU

Gathered basic idea of PCIe w.r.t. our requirement based on Erno's view

1. PCIe is a BUS protocol, so multiple components can share it
   --multiple GBT links can share the PCIe BUS

2. Two types of BUS protocol-EP and RP
   --CRU will use EP BUS and O2 will use RP BUS protocol

3. O2 will see CRU (PCI40)as a piece of HW on PCI BUS
   --multiple FW blocks of CRU will be accessed via BAR registers
   --CSRs(contains BARS) will be mapped to system memory
   --Thus, we can access the FW blocks (registers) of CRU via application program (running on computers) by simple system memory RD and WR

4. To speed up data transfer, we need DMA(minimum processor intervention)
   --processor handover the memory transfer request to DMA and after successful transfer, DMA sends an ack. to processor
   --multiple GBT links send the data to DMA controllers
   --DMA controllers manage the data transfer b/n CRU and system memory through PCI BUS
   --DMA controller is master on that PCI BUS, so, multichannel bus master DMA term is used (from CRU development status slide,18th feb,2015)

5. PCIE configuration space is 256 byte space

Future plans

- More precise requirement analysis of MC BM DMA
  ---data flow,interfaces,architecture
- Comparison table –module required and modules available with Altera
- More DMA performance test with available reference design
- Study the DMA controller