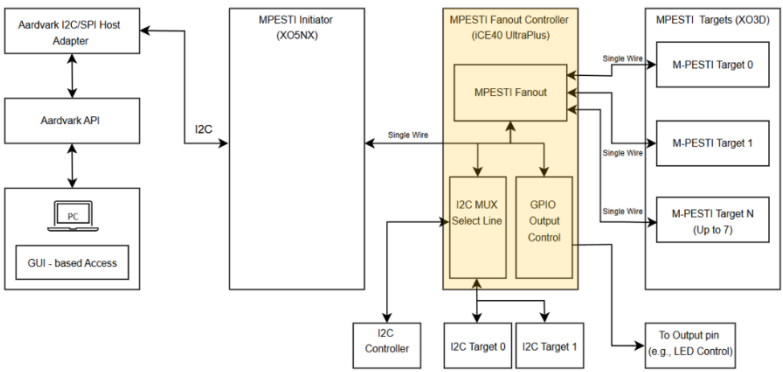
Applications exist where the ability to fan out a PESTI bus to multiple targets exist. One such example is a motherboard or Host Processor Module (HPM) connection to a Power Distribution Board (PDB) with N number of target subsystems such as backplanes or risers. Since the number N is not a priori known by the motherboard, and pre-plumbing for a maximum quantity requires additional interconnects, PESTI fan out support helps scale the ability to support PESTI type features. This document shows a quick demonstration of M-PESTI Fanout Controller using MachXO5-NX as Initiator, iCE40 UltraPlus as Fanout Controller, MachXO3D as Target and Aardvark I2C/SPI Host Adapter acting as the I2C Host.



1 Requirements

The table below shows the summary of software and hardware that are required to run the demonstration.

Software Requirements	Software tool and version	• Lattice Radiant™ Software 2025.2 or higher • Lattice Diamond™ 3.14 or higher • Lattice Diamond™ Programmer version 3.14 or higher • Lattice Propel™ 2025.2 or higher
Hardware Requirements	Board	• **M-PESTI Initiator:** MachXO5-NX Development Board (LFMXO5-25-EVN) • **M-PESTI Target:** MachXO3D Breakout Board (LCMXO3D-9400HC-BEVN) • **M-PESTI Fanout Controller:** iCE40 UltraPlus Breakout Board (iCE40UP5K-B-EVN) • Aardvark I2C/SPI Host Adapter • One (1) I2C Controller and two (2) I2C Target/Sensor (Optional for I2C MUX Feature)

2 Collaterals

The following will be used to run this demonstration.

- **XO5NX_Initiator.jed** – configuration bitstream file for MPESTI Initiator (LFMXO5-25-EVN). This can be found in the Bitstreams folder.
- **iCE40UP_Fanout.bin** – configuration bitstream file for MPESTI Fanout Controller (iCE40UP5K-B-EVN). This can be found in the Bitstreams folder.
- **XO3D_Target.jed** – configuration bitstream file for MPESTI Target (LCMXO3D-9400HC-BEVN). This can be found in the Bitstreams folder.
- **mpesti_demo_gui_source_fanout.py** – Aardvark I2C python driver code for GUI-based access. This can be found in the Aardvark_Source_Code/python folder.

3 Pinout

The table below shows the pinouts related to the MPESTI Initiator (LFMXO5-25-EVN)

Name	MachXO5-NX Ball	MachXO5-NX Development Board Component	Description
rstn_i	G20	SW5	Active low reset signal.
mpesti_io[0]	E4	J9 (4)	M-PESTI line connected to Fanout Controller's mpesti_fanout_io
SCL	E11	J16(1)	I2C Interface Serial Clock Line connected to Aardvark I2C/SPI Host
SDA	D13	J16(2)	I2C Interface Serial Data Line connected to Aardvark I2C/SPI Host

The table below shows the pinouts related to the MPESTI Target (LCMXO3D-9400HC-BEVN).

Name	MachXO3D Ball	MachXO3D Breakout Board Component	Description
reset_n	B3	SW1	Active low reset signal
mpesti_t[0]	D1	J8 (30)	MPESTI Fanout Target [0] line connected to Fanout Controller's mpesti_dst_io[0]
mpesti_t[1]	E1	J8 (28)	MPESTI Fanout Target [1] line connected to Fanout Controller's mpesti_dst_io[1]

The table below shows the pinouts related to the MPESTI Fanout Controller (iCE40UP5K-B-EVN).

Name	iCE40UP5K Ball	iCE40 UltraPlus Breakout Board Component	Description
CRESETB	CRESET_B	SW1	Active low reset signal
mpesti_fanout_io	44	IOB_3B_G6	MPESTI line connection connected to Initiator's mpesti_io[0]
mpesti_dst_io[0]	43	IOT_49A	MPESTI Fanout Target [0] line connected to MPESTI target's mpesti_t[0]
mpesti_dst_io[1]	34	IOT_44B	MPESTI Fanout Target [1] line connected to MPESTI target's mpesti_t[1]
gpio_o[0]	45	IOB_5B	Bit[0] output of GPIO output control. General purpose.
gpio_o[1]	20	IOB_25B_G3	Bit[1] output of GPIO output control. General purpose.
gpio_o[2]	48	IOB_4A	Bit[2] output of GPIO output control. General purpose.
gpio_o[3]	13	IOB_24A	Bit[3] output of GPIO output control. General purpose.
gpio_o[4]	3	IOB_9B	Bit[4] output of GPIO output control
gpio_o[5]	21	IOB_23B	Bit[5] output of GPIO output control. General purpose.
gpio_o[6]	4	IOB_8A	Bit[6] output of GPIO output control. General purpose.
gpio_o[7]	12	IOB_22A	Bit[7] output of GPIO output control. General purpose.

Name	iCE40UP5K Ball	iCE40 UltraPlus Breakout Board Component	Description
scl_controller_io	46	IOB_0A	For I2C MUX feature, connected to external I2C Controller Serial Clock Line
sda_controller_io	2	IOB_6A	For I2C MUX feature, connected to external I2C Controller Serial Data Line
scl_target_io[0]	36	IOT_48B	For I2C MUX feature, connected to external I2C Target Bus[0] Serial Clock Line
sda_target_io[0]	42	IOT_51A	For I2C MUX feature, connected to external I2C Target Bus[0] Serial Data Line
scl_target_io[1]	38	IOT_50B	For I2C MUX feature, connected to external I2C Target Bus[1] Serial Clock Line
sda_target_io[1]	28	IOT_41A	For I2C MUX feature, connected to external I2C Target Bus[1] Serial Data Line

4 Board Jumper Settings

The following board jumper settings for MachXO5-NX Development Board (LFMXO5-25-EVN) were used in this demo. Any jumpers not mentioned are considered open connections.

- JP3 – closed
- J22 – Pin 1 to Pin 2
- J21 – Pin 1 to Pin 2
- J23 – Pin 1 to Pin 2
- J25 – Pin 1 to Pin 2
- J29 – Pin 2 to Pin 3
- J20 – Pin 1 to Pin 2
- J26 – Pin 1 to Pin 2

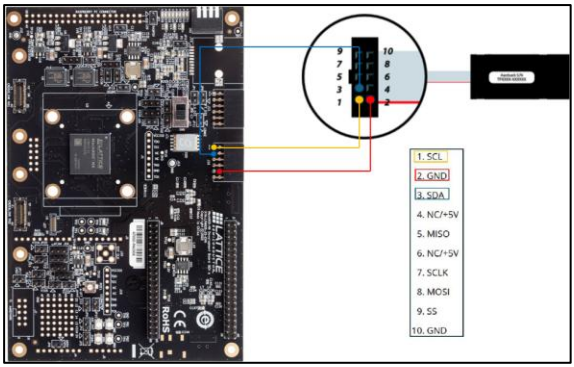
The following board jumper settings for iCE40 UltraPlus Breakout Board (iCE40UP5K-B-EVN). Any jumpers not mentioned are considered open connections.

- J28 – closed
- J51 – closed
- J7 – closed
- J6 – PROG FLASH

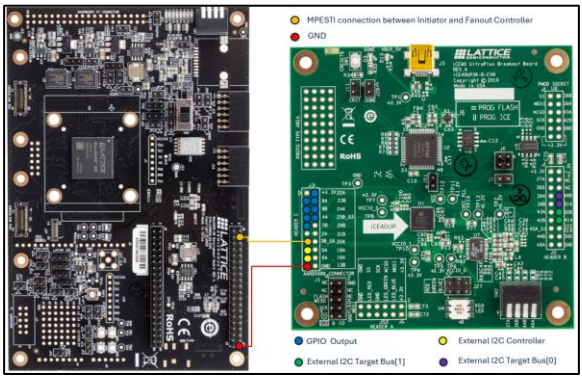
5 Hardware Connections

The following figures show the overall demonstration connections, supplementing the pinout descriptions in Section 3.

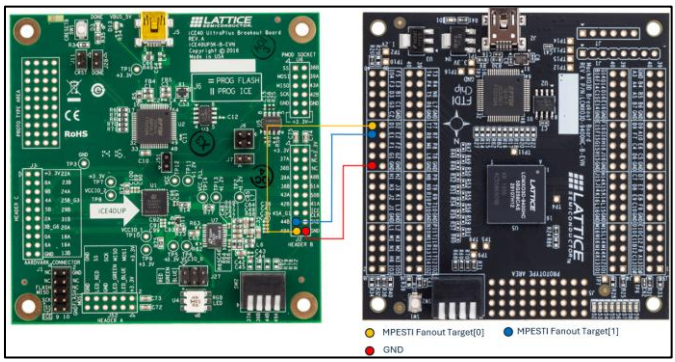
The figure below describes the connections between MPESTI Initiator (LFMXO5-25-EVN) and Aardvark I2C/SPI Host Adapter.



The figure below describes the connections between MPESTI Initiator (LFMXO5-25-EVN) and MPESTI Fanout Controller (iCE40UP5K-B-EVN). The miscellaneous signals are also described.



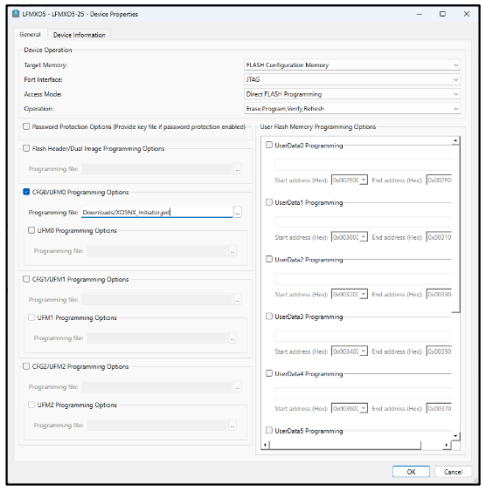
The figure below describes the connections between the MPESTI Fanout Controller (iCE40UP5K-B-EVN) and MPESTI Target (LCMXO3D-9400HC-BEVN).



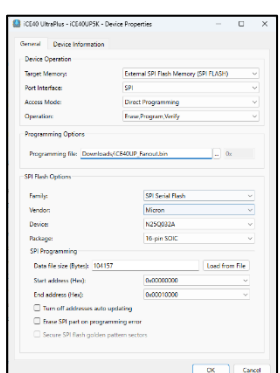
6 Running the Demonstration

The following steps assume that you have already properly set up the hardware connections as described in Sections 3, 4, and 5.

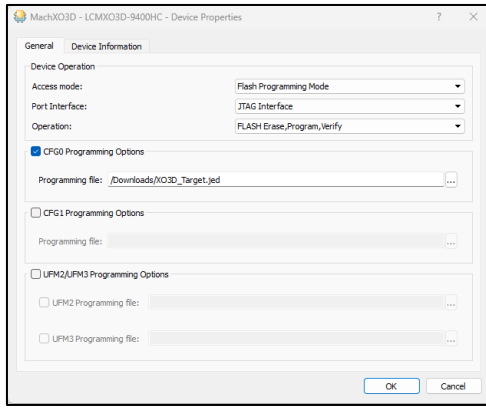
1.) Program the MPESTI Initiator (LFMXO5-25-EVN) board using XO5NX_Initiator.jed. Upon successful programming, LEDs **D1–D8** should **blink**.



2.) Program the MPESTI Fanout Controller (iCE40UP5K-B-EVN) board using iCE40UP_Fanout.bin. Upon successful programming, the **DONE LED should light up**.

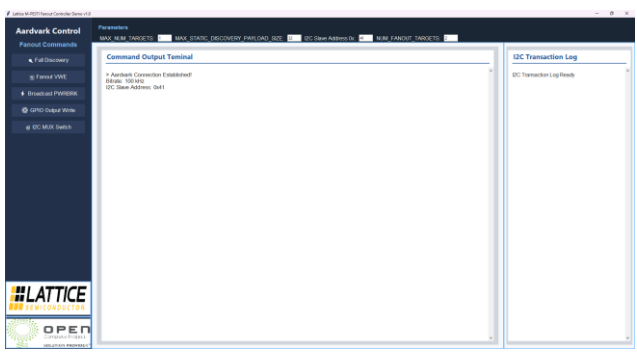


3.) Program the MPESTI Target (LCMXO3D-9400HC-BEVN) board using XO3D_Target.jed. Upon successful programming, LED **D9** should **blink**.

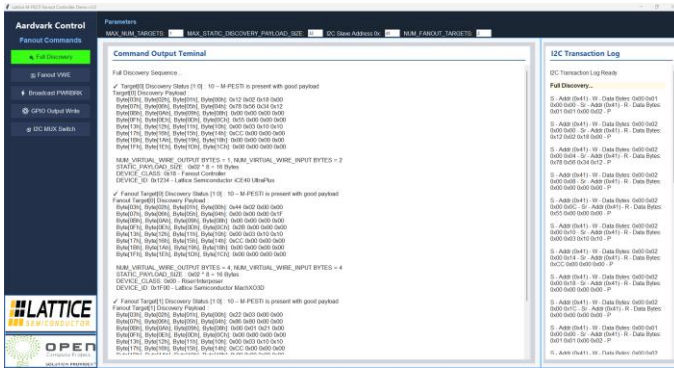


4.) After successfully programming all boards, perform a power cycle. Power-up the MPESTI Initiator Board first then wait for LEDs D1-D8 to blink. Next, power up the MPESTI Fanout Controller Board and wait for DONE LED to light up. Finally, power-up MPESTI Target Board and wait for LED D9 to blink.

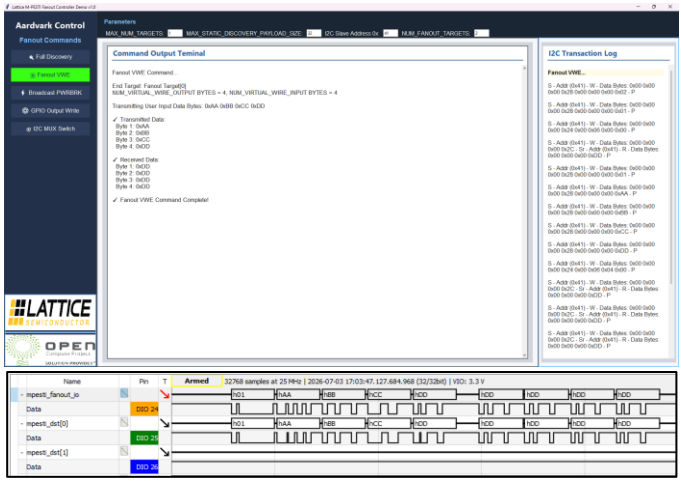
5.) Run the sample python code (**mpesti_demo_gui_source_fanout.py**). Upon successful execution, the GUI should be seen as shown below. For instructions on how to bring up the sample python I2C driver code you may refer to FPGA-RD-02341-1.0 Section 6.1.2 Running the Reference Python Code.

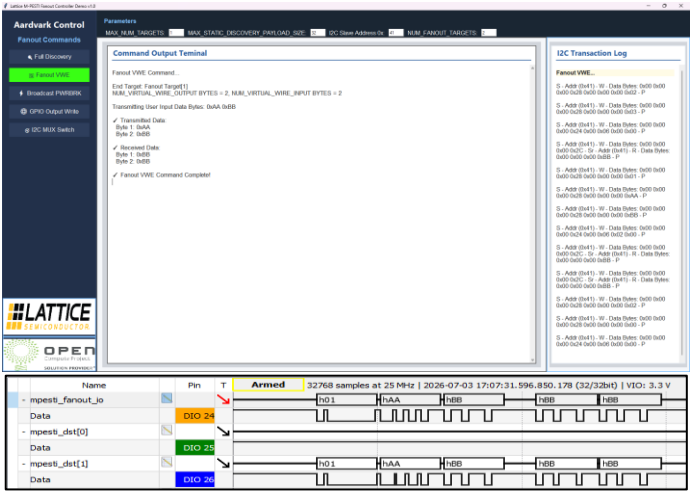


6.) Run the Full Discovery command. This command allows the Initiator to check the discovery status of the Fanout Controller and the connected MPESTI Targets. If the status is detected as "Good Payload Received," the discovery payload is automatically fetched and displayed in the GUI. Note that the demonstration continues to follow the Discovery Payload Format of Specification 1.0, because information in Specification 1.2 regarding the number of supported Virtual Wires is not yet fully defined.

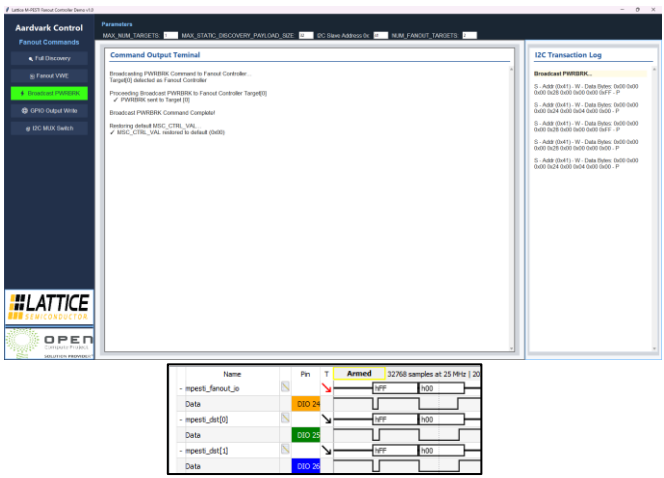


7.) Run Fanout VWE command. This command allows Virtual Wire Exchange between the Initiator and the MPESTI Targets connected to the downstream of the Fanout Controller. This command automatically uses the NUM_VIRTUAL_WIRE_OUTPUT/INPUT BYTES attribute of the MPESTI target that was fetched during Full Discovery.

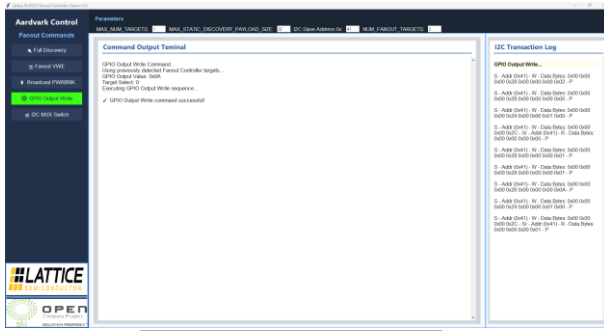




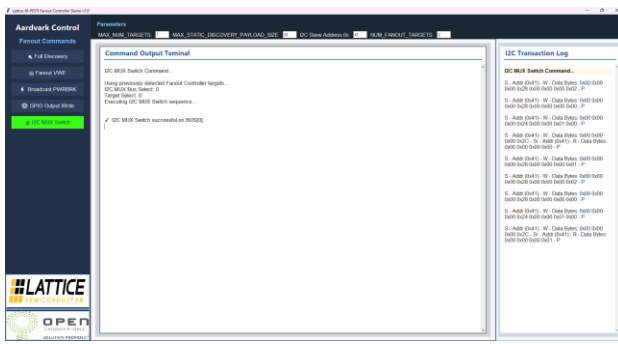
8.) Run Broadcast PWRBRK command. This command allows the Initiator to broadcast the PWBRK command (0x00) to all MPESTI targets connected to the downstream of the Fanout Controller simultaneously.

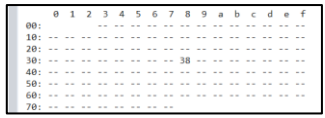


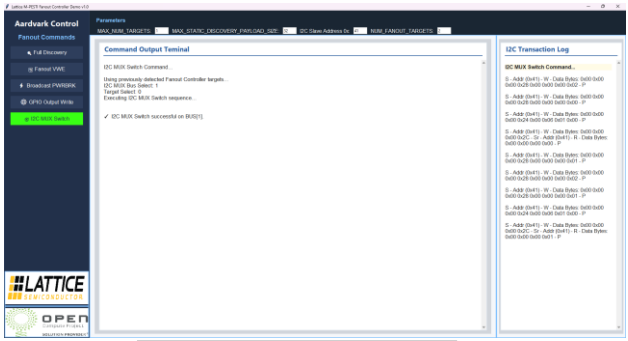
9.) Run GPIO Output Write command. This command allows the Initiator to control the GPIO output of the Fanout Controller through MPESTI.

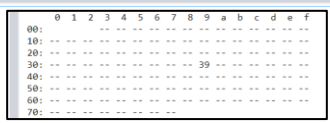


10.) Run the I2C MUX Switch. This command allows the Initiator to control the I2C Target Bus Selection through MPESTI. By default, the I2C Target Bus[0] is selected. In this demo, a target with an I2C address of 0x38 is connected to the Bus[0] while a target with an I2C address of 0x39 is connected to the Bus[1]. The additional external I2C controller will then be able to communicate with the selected I2C Target Bus even if the target devices have the same I2C address. The I2C MUX has been tested to run at 100 kHz.









11.) You have now successfully completed the demonstration.

7 References

- Modular-Peripheral Sideband Tunneling Interface (M-PESTI) Base Specification Version 1.0 Release Candidate 2
- Modular-Peripheral Sideband Tunneling Interface (M-PESTI) Base Specification Version 1.2 Release Candidate 2

Additional Terms and Conditions Applicable to Lattice Programming and Development Hardware

Lattice device programmers, programming cables, socket adapters, and other hardware sold for use in conjunction with Lattice software ("Programming Hardware") and Lattice evaluation boards and development kits sold for use in conjunction with evaluating Lattice products ("Development Hardware") are designed and intended for use solely with semiconductor components manufactured by Lattice Semiconductor Corporation. Programming and Development Hardware is warranted to meet Lattice specifications only for a period of ninety (90) days; in all other respects the terms and conditions of sale of Programming and Development Hardware shall be Lattice's standard terms and conditions set forth in Lattice's Sales Order Acknowledgment. Additionally, Lattice specifications for Programming and Development Hardware limit their use to low-volume engineering applications only, and not for volume production use. The warranty for Programming and Development Hardware will not apply to any Programming or Development Hardware used in production, used with worn or improperly installed hardware, or used with incompatible systems or components.

Technical Support

www.latticesemi.com/support