
Purpose

This demo is for CoreVectorBlox neural network acceleration on the PolarFire® field-programmable gate array (FPGA) devices. This document provides instructions on how to use the corresponding reference design.

Intended Audience

This demo guide is intended for:

- FPGA designers
- Firmware designers
- System level designers
- Data scientists

References

The following documents are referred in this demo guide.

- [CoreVectorBlox SDK Programmer's Guide](#)
- [CoreVectorBlox IP Handbook](#)

Table of Contents

Purpose.....	1
Intended Audience.....	1
References.....	1
1. Introduction.....	3
1.1. Known Issue.....	3
2. Design Requirements.....	4
3. Development Kit for Demo.....	5
4. Demo Design Description.....	7
4.1. System Design.....	7
5. Setting Up the Demo.....	8
5.1. Setting Up the Hardware.....	8
5.2. Programming and Operating the PolarFire SoC Video Kit.....	8
6. Adding Additional Models.....	9
6.1. Obtaining Model File.....	9
7. Error Debugging.....	10
8. Revision History.....	11
Microchip Information.....	12
Trademarks.....	12
Legal Notice.....	12
Microchip Devices Code Protection Feature.....	12

1. Introduction

This document describes how to run the CoreVectorBlox Neural Network using the PolarFire SoC Video Kit, the Dual Camera sensor module, an HDMI monitor, and optionally a HDMI source such as a laptop. The demo design features a fully integrated solution developed using Microchip Libero[®] SoC software to help customers evaluate PolarFire FPGA in Neural Network Vision applications and to build prototypes quickly. For more information, see [Smart Embedded Vision](#).

The demo demonstrates the following functions:

- HDMI display controller
- VectorBlox CNN acceleration of object detection, classification, recognition, and pose estimation demos
- MIPI CSI-2 RX to read one of the cameras

The PolarFire SoC Video Kit (MPFS250TS) includes the following components:

- A PolarFire SoC FPGA (MPFS250TS - 1FCG1152I)
- HDMI 2.0 with rail clamps, ReDrivers, and corresponding connectors
- Dual camera sensor featuring IMX334 Sony image sensor
- Image sensor interface to support up to two MIPI CSI-2 cameras
- Display Serial Interface (DSI)
- NVIDIA[®] Jetson interface (MIPI CSI-2 TX connector)
- A High Pin Count (HPC) FMC connector to connect to high-speed interfaces (such as 12G-SDI and USXGMII)

For more information about the video kit, see [PolarFire SoC Video Kit](#).

1.1 Known Issue

There is a known Reset bug where sometimes the frame buffer management comes out of Reset, incorrectly. If the output video is out of sync, try power cycling the board.

2. Design Requirements

The following table lists the hardware and software required to run the demo.

Design Requirements	Description
Hardware Requirements	
PolarFire® SoC Video Kit	MPFS250TS
USB A to micro cable ⁽¹⁾	Required for the following: <ul style="list-style-type: none"> FPGA programming Serial connection to Linux® running on MSS Optional serial connection to the MSS HSS
HDMI cable ⁽¹⁾	HDMI A Male-to-Male cable
Power adapter ⁽¹⁾	12V, 5A
HDMI monitor	A 1920 x1080 60 Hz resolution monitor for the HDMI 2.0 TX port
Host PC	A host PC with a USB port and HDMI output
Ethernet cable	Used for ssh commands and downloading sample networks and SDK
Software Requirements	
Libero® System-on-Chip (SoC) v2024.2	You must install the full Libero SoC software if you want to re-build the Libero SoC project with the provided Tcl scripts. If you want to only program the FPGA with the pre-built Job file, use Libero SoC FPEXpress or stand-alone FPEXpress. A Libero license is necessary; the video kit comes with a Gold license or an evaluation license that can be obtained from the Licensing tab of the following page. Libero SoC v12.0 and later
CoreVectorblox License	To configure and synthesize the CoreVectorblox IP, a license is required. It is available at SoC portal . Follow these steps to obtain the license: <ol style="list-style-type: none"> Click on the Request Free License option available on the top right corner. Click on the Register button for the required free license option. Enter the Disk ID or MAC ID based on the selected license option and click on the Register button. You will receive the license through email. You can also download the license from the MicrochipDirect site.

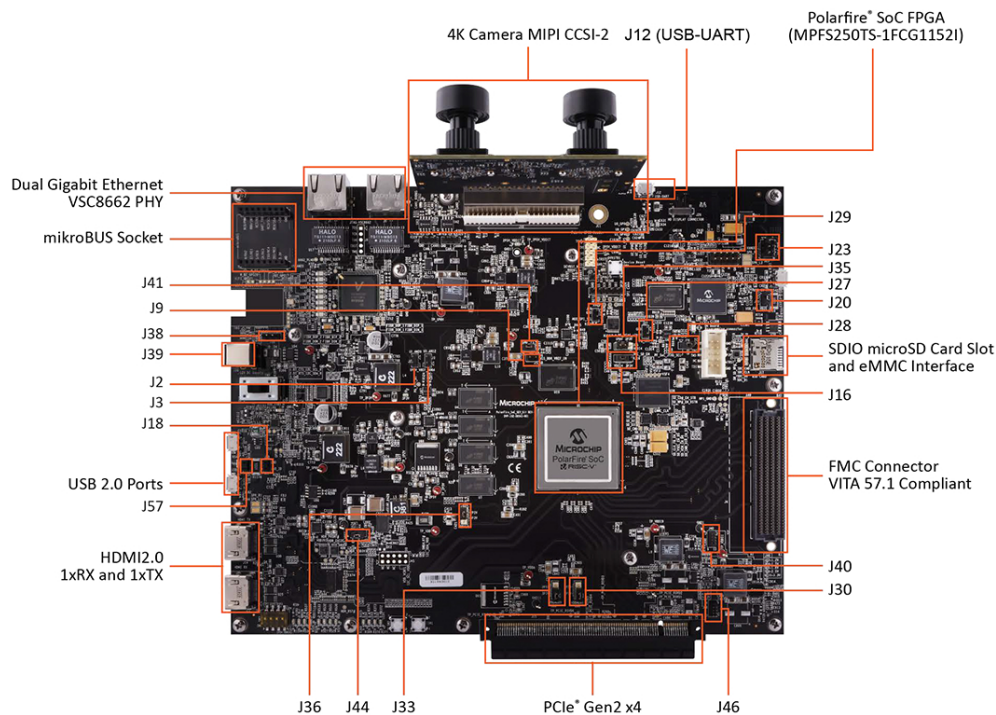
Note:

1. Included with the PolarFire SoC Video Kit.

3. Development Kit for Demo

The following figure highlights the features of PolarFire SoC Video Kit.

Figure 3-1. PolarFire SoC Video Kit Features



The video kit jumper settings required to boot Linux are as follows.

Table 3-1. Jumper Setting Description

Jumper/Switch	Default Position	Functionality
J2 and J3	Open	DDR controller reference voltage. Open: no external reference. Closed: External reference provided.
J9	Open	MSS DDR Vref. Open: no external reference. Closed: External reference provided.
J16 and J35	Open	VCC for eMMC/SD.
J18	Open	USB device mode selection. Open: USB client Closed: USB host
J20	Closed	Required for Embedded FlashPro6 (eFP6). VBUS source. Leave closed.
J23	1 and 2	Backlight LED driver VANODE. FlashPro Jumpers Jumper Description

.....continued

Jumper/Switch	Default Position	Functionality
J28	—	Used to select between Embedded FlashPro6 and external FlashPro header. Closed: eFP6 connected to J5 micro-USB port. Open: External FlashPro connected to J31 header.
J27	—	JTAG nTRST interface pull down enable. Leave it open. J20 VBUS source. Leave it closed.
J30	1 and 2	VDDAUX1 voltage.
J33	1 and 2	VDDAUX9 voltage.
J38	Open	WiFi chip I2C CLK and data.
J36	Open	VDDAUX4 voltage.
J40	9 and 10	Bank9 voltage 3v3.
J41	1 and 2	125 MHz output.
J44	1 and 2	Core voltage (VDD) set to 1.05v.
J46	9 and 10	Bank1 voltage 3v3 for CAN testing.
J57	Open	USB device mode selection. Open: USB client Closed: USB host

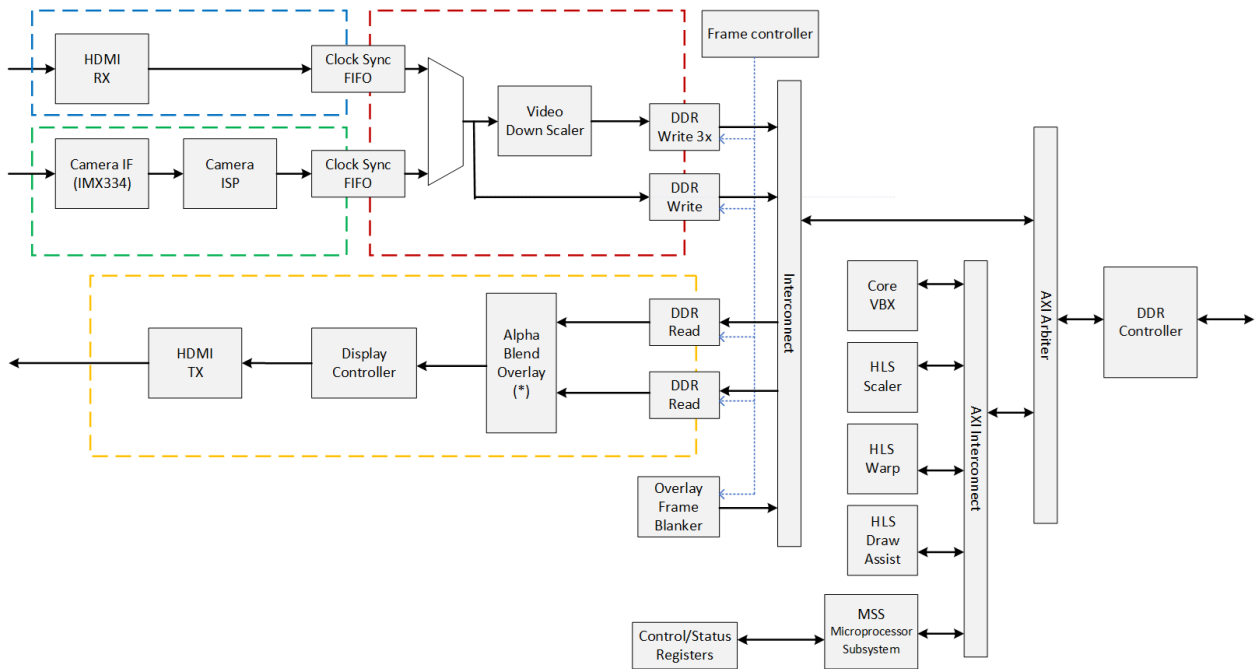
4. Demo Design Description

The following section provides an overview of the dataflow in the demo design. This system design allows various single and multi-stage networks to run on the same hardware.

4.1 System Design

The following diagram illustrates an overview of the dataflow in the design.

Figure 4-1. System Dataflow



Sequence of data flow shown in the figure above is as follows:

1. Input
 - a. Received from HDMI Rx: If a source is connected to J13, the video frame data is sent directly to a Frame Writer which writes the video data to DDR.
 - b. Received from Camera (Camera IF IMX334): By default, if the IMX334 is connected to the board, video frame data is sent to the Camera ISP, and then the Video Down Scaler writes the video data to the DDR.
2. Based on demo requirements, MSS instructs the Image Scaler and/or the Image WarpAffine block to process the input frame appropriately for CNN(s). The result of scaling and warping is written to DDR.
3. MSS instructs CoreVectorBlox IP to run the appropriate CNN using the scaled and warped image as input to CNN. The result of CNN is written back to DDR by CoreVectorBlox IP.
4. MSS reads the result of CNN from DDR and runs post-processing software routines. The result is then drawn on the original Input Frame Buffer.
5. The Frame Reader reads the frame and streams the frame data to the HDMI Tx block.

5. Setting Up the Demo

The following steps describe how to set up the demo:

1. Setting up the Hardware
2. Programming and operating the PolarFire Device

5.1 Setting Up the Hardware

Setting up the hardware involves interfacing the dual camera sensor module and, optionally, the HDMI input (HDMI 2.0 RX) and the HDMI monitor with the PolarFire SoC Video Kit, and verifying the jumper settings.

To set up the hardware, perform the following steps:

1. Connect the IMX 334 camera to the PolarFire SoC Video Kit as shown in [Figure 3-1](#). Connect an HDMI (1080p) input source to J13 (HDMI 2.0 RX port) of the video kit through HDMI cable.
2. Connect the Full HD HDMI (1080P) monitor to J14 (HDMI 2.0 TX port) of the video kit using the HDMI cable.
3. Connect the host PC to J12 (USB-UART) of the video kit using the USB micro cable.
Note: If the UARTs on the Silicon Labs Quad CP2108 USB to UART Bridge are not found you may need to Install the CP210x Windows Drivers from SiLabs.
4. You will require two serial communication terminals using the Quad USB-UART on J12. These will usually be the lowest numbered COM ports.
5. You can use software like PuTTY <https://www.putty.org/> (Baud rate = 115200, Data bits = 8, Stop bits = 1, and No Parity or Flow control)
6. Connect the power supply cable to J39 of the video kit.
7. Ensure that the jumper settings are set on the video kit. The video kit is shipped in this configuration. For jumper position and functionality, see [Table 3-1](#).
8. Power-up the HDMI monitor.
9. Power-up the board using the SW5 slide switch.
10. Update the board through FlashPro Express using the [2023.02 pre-built release](#). (This will require another micro-USB cable to be connected from the PC to J5).

5.2 Programming and Operating the PolarFire SoC Video Kit

For current instructions on programming and operating the Demo, see the [Github Repository](#).

6. Adding Additional Models

By default, the demo runs the face demo on start-up and can switch to run other default two-stage recognition or single-stage detection demos. In this document, you will see an example of adding in a new network, posenet.

Note: If adding a new model, simple detection models will use an interrupt based approach, while recognition focused demos will use a poll based approach.

Currently, post-processing routines exist for object detection, classification, facial recognition, license plate recognition, and pose estimation. Users will have to write additional post-processing code to run other networks.

The following sections describe how to run the alternate models

6.1 Obtaining Model File

To obtain the model files, you can run the tutorial available in the VectorBlox SDK in github.com/Microchip-Vectorblox/VectorBlox-SDK. You can find instructions for running the tutorials in the Programmer's Guide, which is available as part of the SDK documentation. Generate the model using this tutorial: `VectorBlox-SDK/tree/release-2.0/tutorials/tensorflow/posenet`.

Generated .vnnx networks can be transferred to the SoC-videokit via SCP.

In the `run-video-model.cpp` file in the `soc-video-c` directory, you see the following code.

```
struct model_descr_t models[] = {
    {"Yolo v8n", "/home/root/samples_V1000_2.0/yolov8n.vnnx", 0,
"ULTRALYTICS CUT"},
    {"SCRFD", "/home/root/samples_V1000_2.0/scrfd_500m_bnkps.vnnx", 0, "SCRFD"},
    {"mobileface-arcface", "/home/root/samples_V1000_2.0/mobilefacenet-arcface.vnnx", 0,
"ARCFACE"},
    {"MobileNet V2", "/home/root/samples_V1000_2.0/mobilenet-v2.vnnx", 0, "IMAGENET"},
};
```

Change the code to the following:

```
struct model_descr_t models[] = {
    {"Yolo v8n", "/home/root/samples_V1000_2.0/yolov8n.vnnx", 0,
"ULTRALYTICS CUT"},
    {"SCRFD", "/home/root/samples_V1000_2.0/scrfd_500m_bnkps.vnnx", 0, "SCRFD"},
    {"mobileface-arcface", "/home/root/samples_V1000_2.0/mobilefacenet-arcface.vnnx", 0,
"ARCFACE"},
    {"MobileNet V2", "/home/root/samples_V1000_2.0/mobilenet-v2.vnnx", 0, "IMAGENET"},
    {"Pose Estimation", "/home/root/samples_V1000_2.0/posenet.vnnx", 0,
"POSENET"},
};
```

7. Error Debugging

If an error is encountered, both the relevant error message and error code are written out to the UART. For more information and to proceed with the next steps, see the *CoreVectorBlox IP Handbook*.

8. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
C	12/2024	The following is the list of changes in this revision of the document. <ul style="list-style-type: none">• Added camera functionality• Implemented support for TFLite on hardware
B	06/2024	The following is the list of changes in this revision of the document. <ul style="list-style-type: none">• Updated face capture features• Added additional posenet demo
A	12/2023	Initial Revision

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>.

ISBN: 979-8-3371-0373-0

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.