



FPGA Made Easy

Start with Kiwi board and Gowin EDA tool

Advisors: Duong Van Thanh
Truong Ngoc Phu
Hoang Duc Hieu
Authors: Dinh Xuan Quang

HO CHI MINH CITY, JULY 2025

Contents

1	Arm cortex-m3 with Gowin-EMPU IP	3
1.1	MCU design	3
1.1.1	Installing GMD software	3
1.1.2	Setting GMD Software	3
1.2	FPGA module design	8
1.3	Loading code to FPGA	11
2	References	15

1 Arm cortex-m3 with Gowin-EMPU IP

Gowin_EMPU is a System-on-Chip FPGA device that embeds MCU hard core, Gowin FPGA Fabric as well as other standard peripherals and featured hard cores, including 32K-Byte FLASH and configurable 2K-Byte or 4K-Byte or 8K-Byte or 16K-Byte SRAM. The MCU contains a low power, low cost and high-performance ARM Cortex-M3 32-bit RISC. The flexible FPGA Fabric serves as user programmable peripherals, or soft-core IPs.

All internal block and sub-block functions of the Gowin_EMPU, including the MCU(with supporting AHB/APB Bus System/Extension Ports and Standard Peripherals – GPIO , UART, Timer, RTC, Watch-Dog), FPGA Fabric (with BSRAM, User Flash, Sub-Memory Controller, rich clock and flexible Logic Resources) and JTAG Config-Core are all designed around the ARM Cortex-M3 processor core. The ARM Cortex-M3 processor provides the core for high-performance, low-cost platform that meets the needs of minimal memory implementation, reduced pin count, and low-power consumption, while delivering outstanding computational performance and exceptional system response to interrupts.

Before starting, you should install **Gowin MCU design(GMD)** in gowin website or using **arm keilV5**. In this project, we use GMD and guide you setting simple project for Arm with FPGA onekiwi4k board

For reference software and setting, you can find document IPUG928 Gowin

1.1 MCU design

For more information and document, you should find in https://www.gowinsemi.com/en/support/database/569/?support_search=GOWIN+MCU+Designer and remember to register before downloading

1.1.1 Installing GMD software

You can download GMD installation package (GMD_V1.2.exe) via Gowin official website https://www.gowinsemi.com/en/support/database/569/?support_search=GOWIN+MCU+Designer The setting up for GMD software is in the file SUG549 Gowin MCU designer user guide

1.1.2 Setting GMD Software

Gowin contribute all reference design in MCU, you can install [https://cdn.gowinsemi.com/cn/Gowin_EMPU\(GW1NS-4C\)_V2.0.zip](https://cdn.gowinsemi.com/cn/Gowin_EMPU(GW1NS-4C)_V2.0.zip) Next open GMD software and choosing file -> Open project from File system

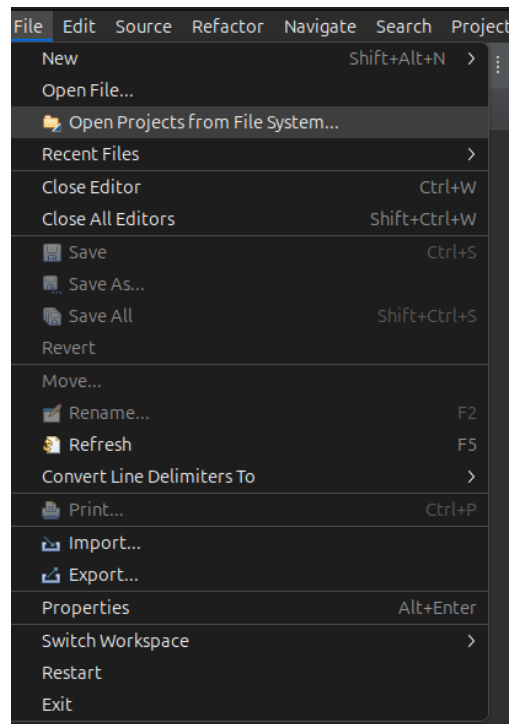


Figure 1: *File setting*

Then choosing directory for finding folder

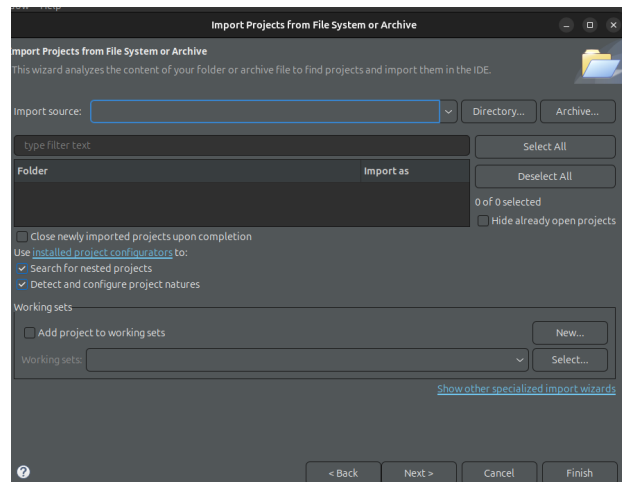


Figure 2: *File setting1*

Next,choosing cm3 folder ,click ok , click finish

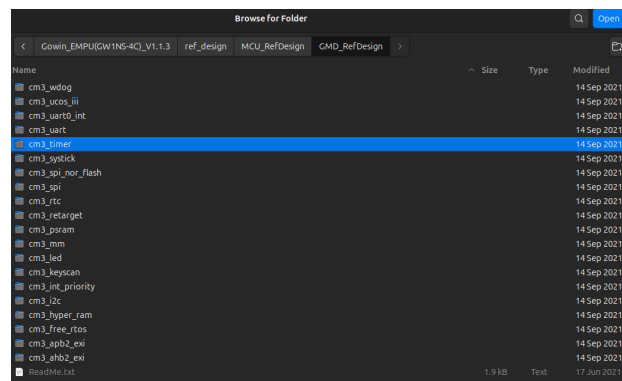


Figure 3: *File setting2*

You can see that the project already include library , you can read and call function in main for using it

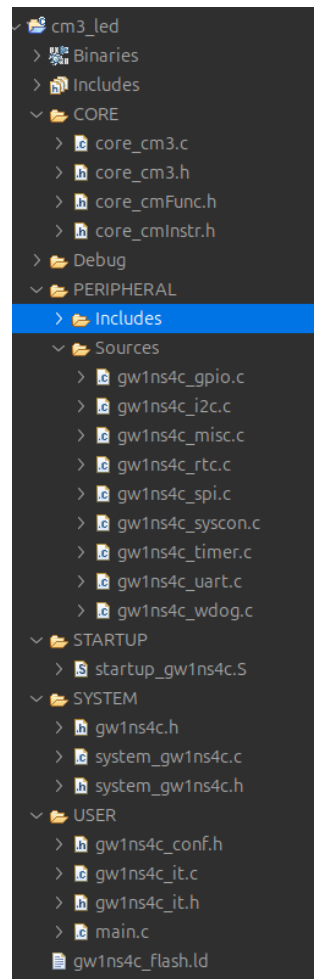


Figure 4: *File setting3*

Now we will run and coding for example project blinkly led

```
int main(void)
{
    SystemInit(); //Initializes system
    GPIOInit();  //Initializes GPIO

    while(1)
    {
        GPIO_ResetBit(GPIO0,GPIO_Pin_0); //LED1 on
        delay_ms(1000);

        GPIO_SetBit(GPIO0,GPIO_Pin_0);  //LED1 off
        delay_ms(1000);
    }
}
```

}

Then click Build in taskbar like figure below:

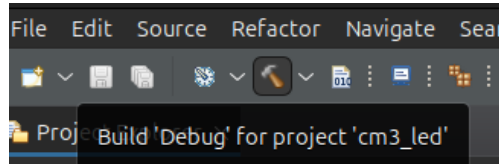


Figure 5: *Building logo*

Checking the error after build

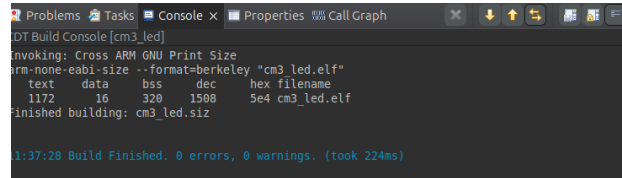


Figure 6: *Error checking*

Then you will see the file bin created:

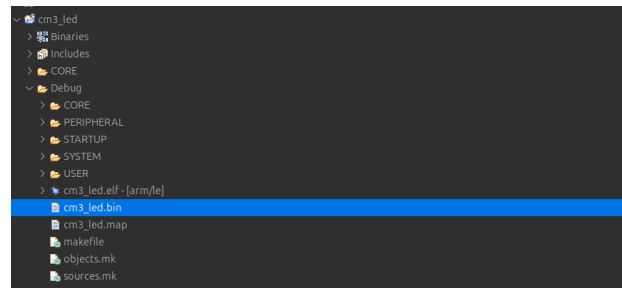


Figure 7: *Checking file bin 1*

Checking the time for build:

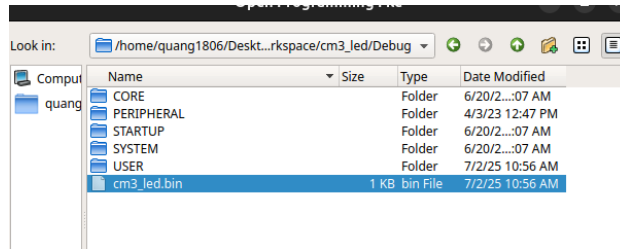


Figure 8: *Checking file bin 2*

1.2 FPGA module design

Choosing the Gowin_EMPU(GW1NS-4C) in the folder Hard-Core-MCU.

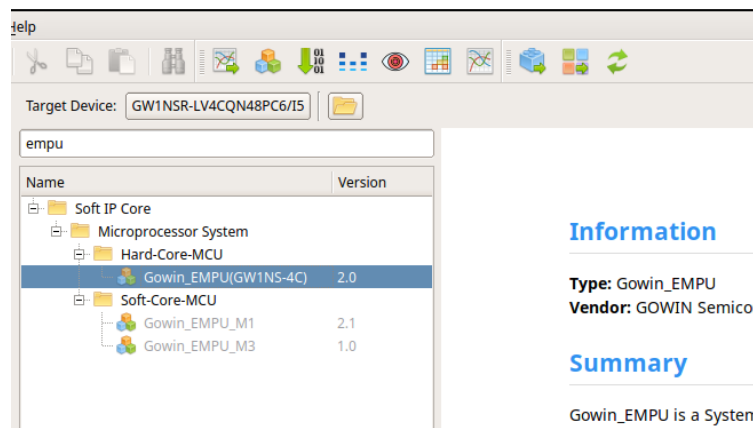


Figure 9: *Finding Gowin EMPU IP*

You will see the table below, Gowin EMPU support a lot of feature Then choosing GPIO then enable GPIO like figure below :

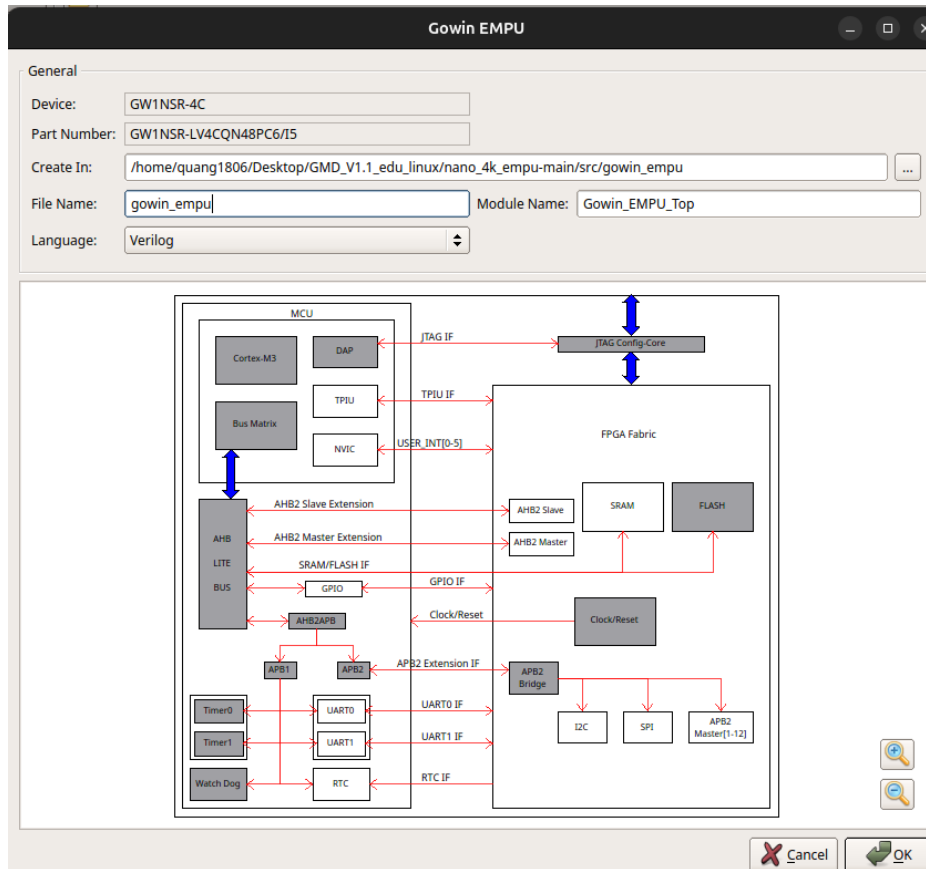


Figure 10: All features of MCU module

The following IP modules are integrated into the Gowin EMPU platform, based on the Cortex-M3 subsystem and FPGA fabric:

- **MCU Core Subsystem:** Cortex-M3 Processor Core, NVIC (Nested Vectored Interrupt Controller), DAP (Debug Access Port), TPU (Trace Port Unit), Bus Matrix (AHB interconnect)
- **AHB/APB Interfaces:** AHB-Lite Interface, AHB2 Master and Slave Extension Interfaces, APB1 and APB2 buses, APB2 Extension Interface
- **Peripheral IPs:** Timer0, Timer1, Watchdog Timer, RTC (Real-Time Clock), UART0, UART1, GPIO Interface, I2C (via APB2 Bridge), SPI (via APB2 Bridge)
- **Interrupt and Control:** USER_INT[0:5] External Interrupts, Clock / Reset Manager
- **Memory Interfaces:** SRAM Interface, FLASH Interface, SRAM/FLASH combined interface on AHB2

- **FPGA Fabric Interfaces:** AHB2 Master / Slave Interfaces, APB2 Bridge, APB2 Master (1–12) for peripheral extensions
- **Debug and Configuration:** JTAG Interface, JTAG Config Core, TPU Interface

In this project, we just use GPIO and UART, choosing the GPIO the table will appear, then you just choose enable GPIO then click OK.

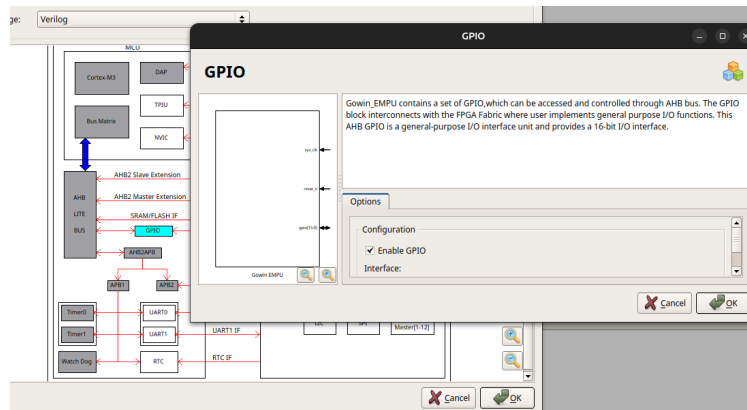


Figure 11: GPIO setting

Next, we will choose UART0 then choosing enable UART0 then click OK

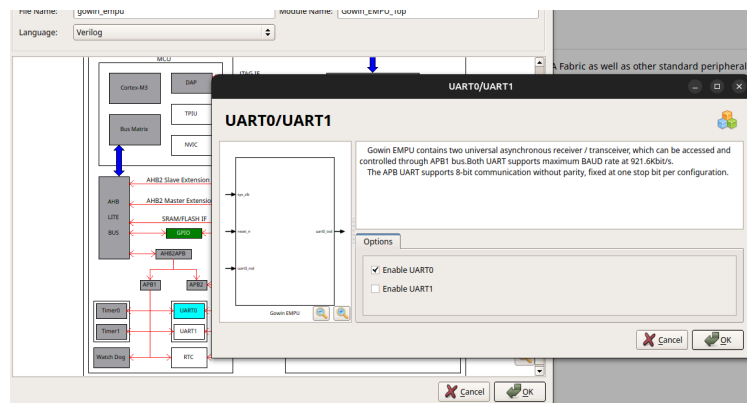


Figure 12: UART setting

Then click OK for finish IP Setting, you will see module:

```
Gowin_EMPU_Top your_instance_name(
    .sys_clk(sys_clk), //input sys_clk
    .gpio(gpio), //input [15:0] gpio
    .uart0_rxd(uart0_rxd), //input uart0_rxd
    .uart0_txd(uart0_txd), //output uart0_txd
    .reset_n(reset_n) //input reset_n
);;
```

1.3 Loading code to FPGA

The code in the top module :

```
module empu_top(  
    input resetButton,  
    input crystalClk,  
    input UART_RX,  
  
    output UART_TX,  
    inout[7:0] GPIO  
);  
Gowin_EMPU_Top cortexM3(  
    .sys_clk(crystalClk), //27MHz system clock  
    .gpio(GPIO), //only 8 GPIO pins (GPIO_Pin_7 to GPIO_Pin_0) are exposed  
    .uart0_rxd(UART_RX), //input uart0_rxd  
    .uart0_txd(UART_TX), //output uart0_txd  
    .reset_n(resetButton) //input reset_n  
);  
endmodule
```

Setting physic constraint base on FPGA board

```
IO_LOC "GPIO[7]" 35;  
IO_PORT "GPIO[7]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "GPIO[6]" 34;  
IO_PORT "GPIO[6]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "GPIO[5]" 32;  
IO_PORT "GPIO[5]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "GPIO[4]" 31;  
IO_PORT "GPIO[4]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "GPIO[3]" 27;  
IO_PORT "GPIO[3]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "GPIO[2]" 28;  
IO_PORT "GPIO[2]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "GPIO[1]" 29;  
IO_PORT "GPIO[1]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "GPIO[0]" 30;  
IO_PORT "GPIO[0]" IO_TYPE=LVCMS25 PULL_MODE=UP DRIVE=8;  
IO_LOC "UART_TX" 19;  
IO_PORT "UART_TX" PULL_MODE=NONE DRIVE=8;  
IO_LOC "UART_RX" 20;  
IO_PORT "UART_RX" PULL_MODE=UP;  
IO_LOC "crystalClk" 45;
```

```
IO_PORT "crystalClk" PULL_MODE=UP;
IO_LOC "resetButton" 15;
IO_PORT "resetButton" PULL_MODE=UP;
```

Next step choosing programmer, a table of gowin programmer will appear, click edit on taskbar then choosing configure device like figure below:

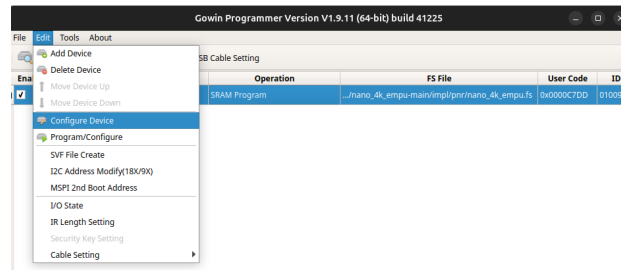


Figure 13: *Setting configure device 1*

Next change access mode to MCU Mode:

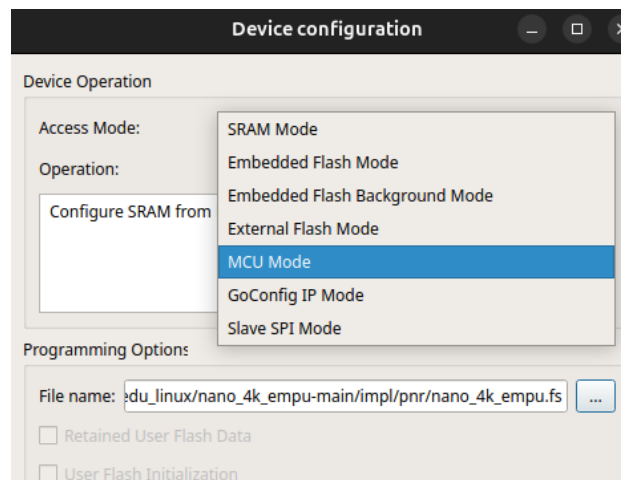


Figure 14: *Setting configure device 2*

Then change the Operation to Firmware Erase,Program

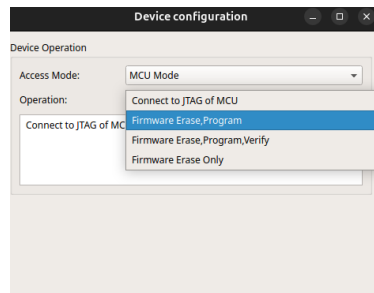


Figure 15: *Setting configure device 3*

Next setting FW/MCU/Binary Input Option, choosing file bin create from build in GMD like figures below:

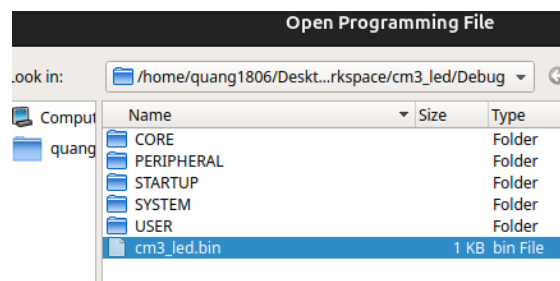


Figure 16: *Setting configure device 4*

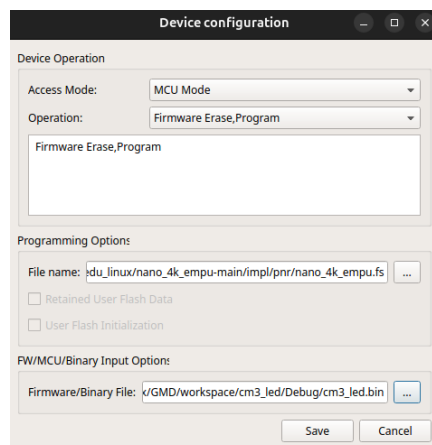


Figure 17: *Setting configure device 5*

Then click save and click run programm to upload coard to FPGA

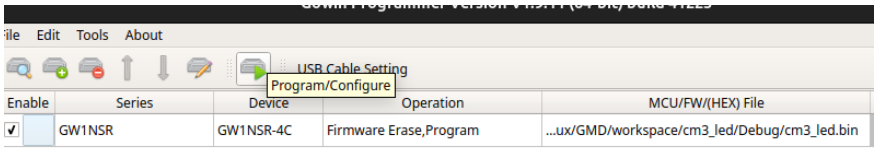


Figure 18: Running program

2 References

1. Gowin Semiconductor Corp., *Gowin HyperRAM Memory Interface IP User Guide*, Version 2.1E.
2. Gowin Semiconductor Corp., *Gowin Video Frame Buffer IP User Guide*, Version 2.1E.
3. Gowin Semiconductor Corp., *Gowin DVI TX IP User Guide*, available with IDE package.
4. Gowin Semiconductor Corp., *Clock Resource and PLL Guide*, documentation for GW_PLLVR, TMDS_PLLVR, and CLKDIV modules.
5. CoreCourse.cn — FPGA learning portal with practical Verilog tutorials and IP integration: <http://www.corecourse.cn>
6. Gowin EDA Tools Help System — Built-in IDE documentation for IP configuration, timing constraints, and simulation setup.
7. Gowin Semiconductor Corp., *Gowin HDMI/DVI RX and TX IP User Guide*, describing HDMI/DVI Rx and Tx functionality and integration in Gowin FPGAs.
8. Gowin Semiconductor Corp., *Gowin MIPI D-PHY RX/TX Advance IP User Guide*, IPUG948-2.1.1E, covering MIPI CSI/DSI physical layer.
9. Gowin Semiconductor Corp., *Gowin MIPI D-PHY RX/TX IP User Guide*, IPUG112-2.02E, earlier version of MIPI D-PHY documentation.
10. Gowin Semiconductor Corp., *Arora V Hardened MIPI D-PHY User Guide*, UG296-1.1.1E, MIPI for Arora-V devices
11. Gowin Semiconductor Corp., *White Paper: Gowin MIPI Solution Capability Overview*, WP1153-1.1E, a summary of MIPI IP offerings.