AN-1125 In-System Debug for GreenPAK™ Devices with I2C Serial Communication

Silego Technology Inc. has introduced many different GreenPAK (GPAK) Programmable Mixed-signal Matrix devices over the last six years. The development cycle for these devices has traditionally included debugging sessions that would take place on the GPAK Universal Development Board, and include the option to test and debug multiple versions of the customer configuration quickly. Because the debugging could take place without the need to program individual devices, it would significantly increase the speed of development. Once the configuration is sufficiently verified as a stand-alone function, the user can program some number of devices and place on their board for further debug in a system environment.

In-System Debug (ISD)

Some of the newest GPAK devices have I2C communication capability, which can be useful as a function in the end-user system.

The I2C communication macro-cell in these devices can also serve as a portal to allow the designer to migrate their GPAK debug from the Universal Development Board to their project PCB for In-System Debug (ISD). This new ISD capability allows designers to speed up their already fast GPAK development cycle.

GPAK is composed of functional macro-cells (such as Look Up Tables, Timer/Counters, Oscillators, etc.) that are configured by the designer using Silego's GPAK Designer software. The choices made by the designer in the GPAK Designer GUI are translated into register values that both configure the individual macro-cells, and establish the signal connections and routing through the Connection Matrix on the GPAK device. The resulting register values are programmed into Non-Volatile Memory (NVM) on the GPAK device. Each GPAK device is designed to load this information into latches on the device during Power-On Reset (see Figure 1 below). The latches are what actually control the configuration of the device, but they are volatile, and lose their stored information when power is removed.

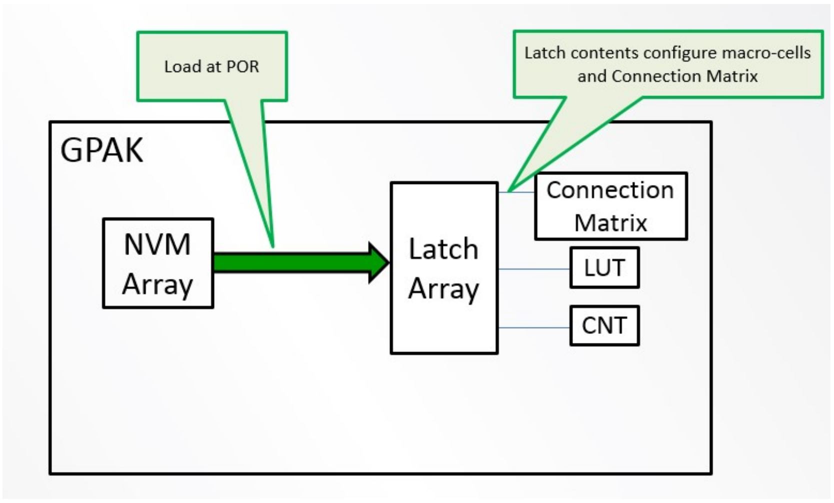


Figure 1. Latch configuration during Power-On Reset

One advantage of this architecture is the ability to write other register values via I2C into the latches during operation. This provides the mechanism to support debugging multiple user configurations without programming the NVM. In previous GPAK devices, this step could only take place with the device placed in a socket on the Universal Development Board. The communication between the Universal Development Board and the GPAK to set the latch values in previous devices used multiple pins and special signal sequences (see Figure 2 below), including an overvoltage (~7V) signal on one of the pins. This is the same communication required to program the NVM on a GPAK device.

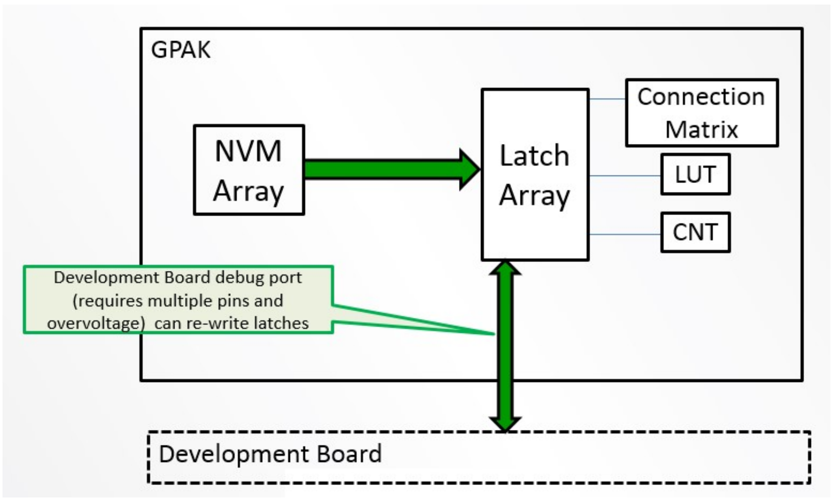

GPAK
NVM Array
Latch Array
Connection Matrix
LUT
CNT
Development Board debug port (requires multiple pins and overvoltage) can re-write latches
Development Board

Figure 2. Development Board debug port

The GPAK devices which include the I2C communication macro-cell have an alternative way to load this information into the latches. As stated above, the latches will always be loaded on POR from the NVM, but this information can be modified after POR takes place on. This can serve as a convenient "debug port" to allow modification of GPAK device configuration. Modifying the latches (and therefore changing the configuration) via I2C commands can take place either on the development board, or even when the GPAK device is mounted on the target hardware, allowing for In-System Debug capability (see Figure 3).

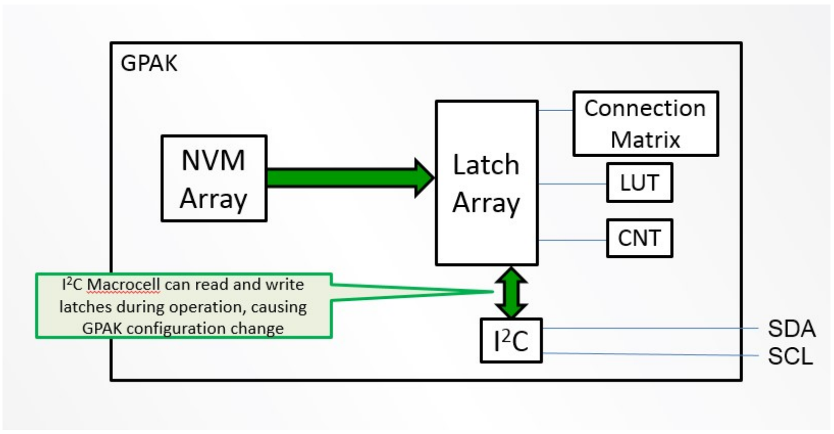

GPAK
NVM Array
Latch Array
Connection Matrix
LUT
CNT
I²C Macrocell can read and write latches during operation, causing GPAK configuration change
I²C
SDA
SCL

Figure 3. I2C In-System Debug capability

The new version of GPAK Designer software is set up to easily allow the user to implement In-System Debugging on their product. The Emulator window inside GPAK Designer has a new button to support this function. Shown highlighted on Figure 4 below in the upper right corner, this new button displays "Device Onboard" when targeting the GPAK device in the Universal Development Board socket, and can also be used to select communication to GPAK devices located on the target board.

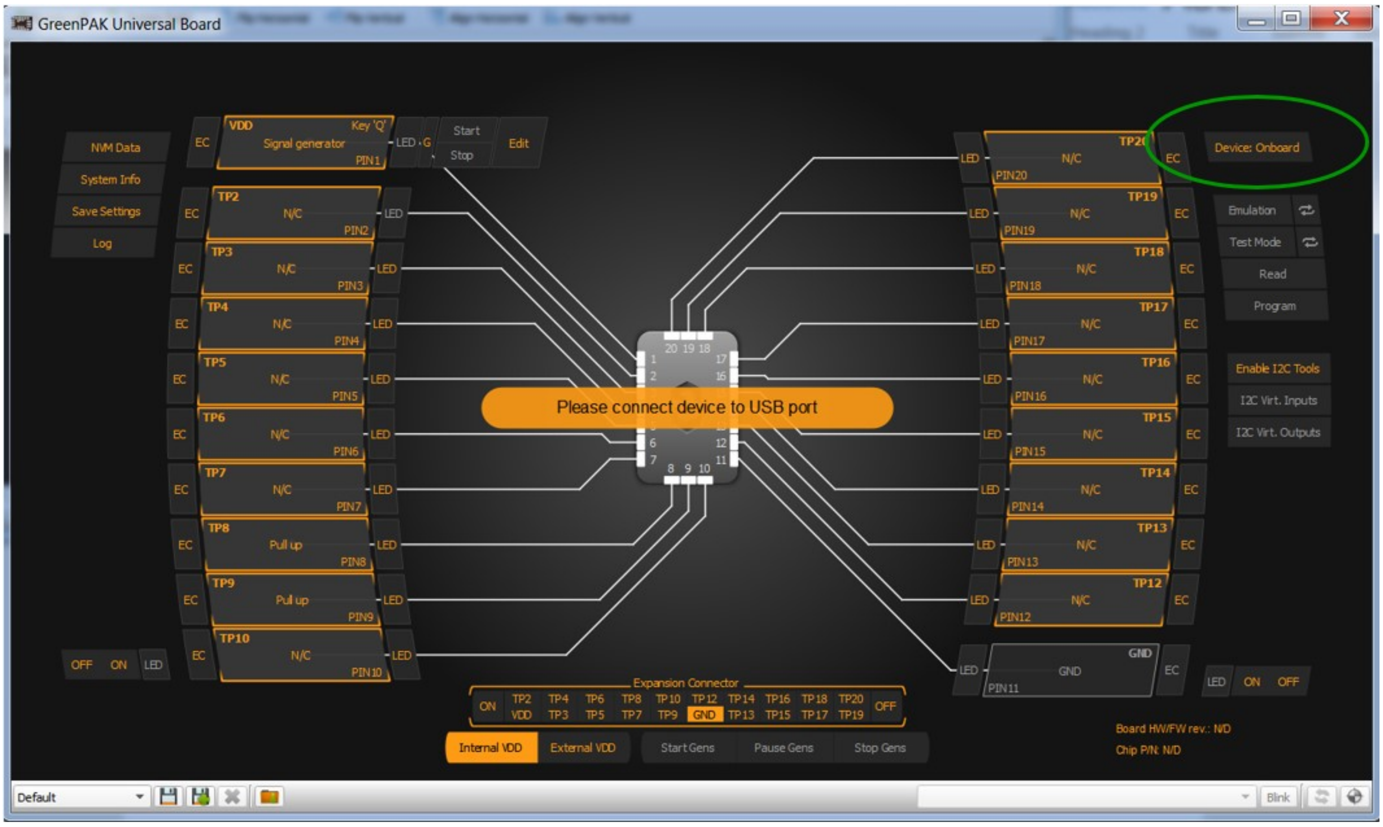
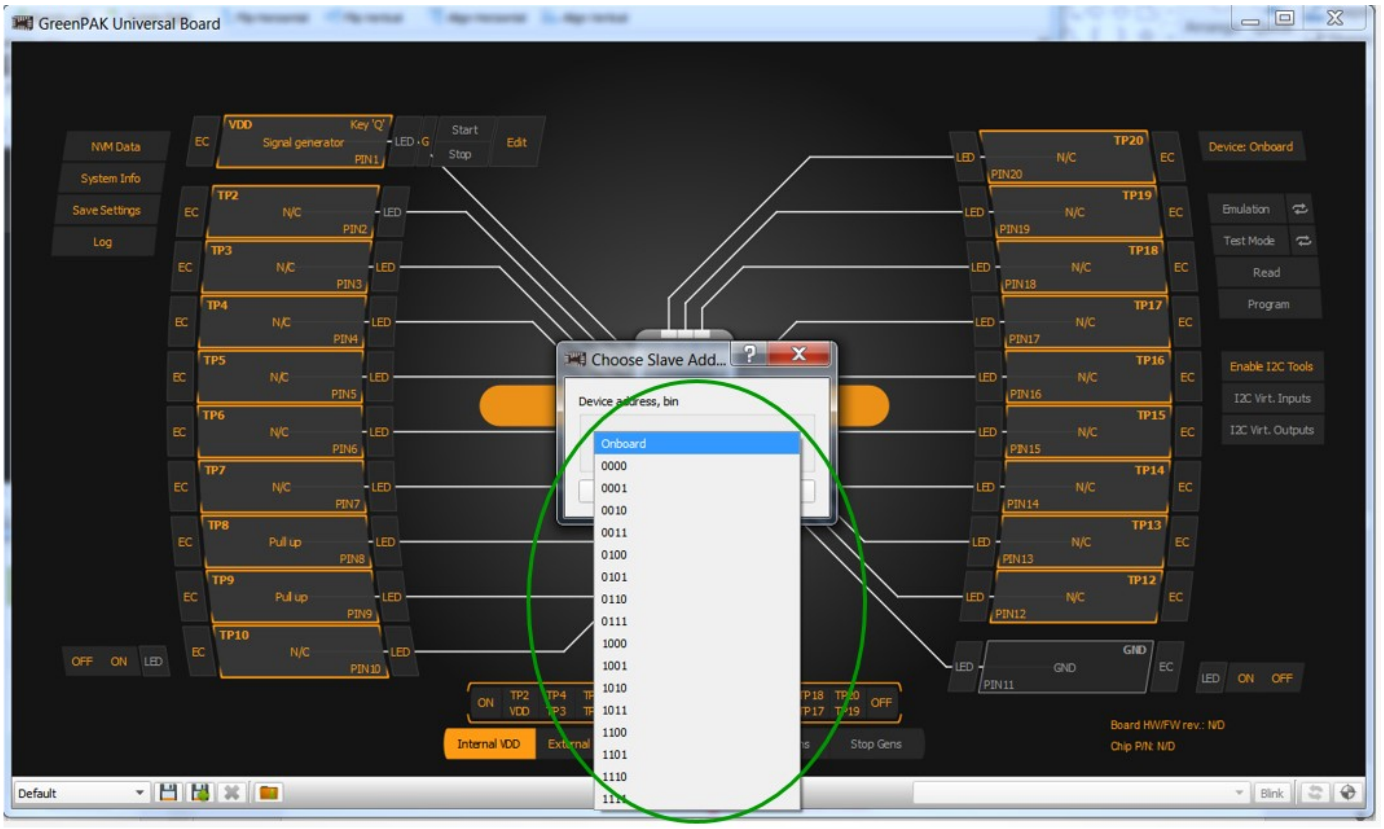

Please connect device to USB port

Figure 4. "Device Onboard" button in GreenPAK Emulation Tool window

When pushing this button, it opens up a new window (shown in center on Figure 5 below) that allows the user to either select the default "Device Onboard", or one of the 16 possible wake-up addresses that the GPAK device can respond to.


Choose Slave Add...

Device address, bin

Onboard
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111

Figure 5. 16 possible wake-up addresses

If "Device Onboard" is selected, this tells the development hardware to expect a device in a socket on the Development board, as is traditional in doing debug on GPAK devices, and to communicate with the GPAK device through the socket interface.

If the user selects one of the 16 possible wake-up addresses, this tells the development hardware that the user wants to do In-System Debug on their target board via I2C commands.

By default, the GPAK device will wake up to address 0000b. If there is only one GPAK device on the target board (and no other I2C slave devices that could cause address contention), this default address can be used with no extra programming step.

If, however there are other I2C slave devices on the target board, or if the user would like to have more than one GPAK device on the same I2C bus, the user will need to pre-program the slave address into the GPAK before mounting on their board.

When the user pushes the “Device Onboard” button and selects a wakeup address, the Development Board becomes a USB to I2C bridge, and will execute all GUI commands to the remote GPAK device. When this selection is made, and the user pushes the “Emulation” button, the system will automatically download the entire set of data for a particular configuration, which will enable that configuration, once the I2C write commands are complete.

However, before this can take place, the user must connect the appropriate terminals on the Development Board for the SDA and SCL signals, to their target board, as well as establish a common ground between the Development Board and the target board. Note that some of the Development Board functions, such as signal generators and LEDs, will not function when working in this ISD mode.

It is important to re-iterate that the GPAK device, when first powered up on the target board, will go through a standard POR cycle which will include doing an initial load of configuration information from NVM to the latches. This highlights one area where ISD may not be too effective, which is to do debug of the behavior of the GPAK device during the POR period.

This is the case because the GPAK device on the target board will first exhibit the behavior based on the internal NVM configuration, before it is re-configured through I2C commands to the configuration under test. To test POR operation and performance, it may be better to program actual GPAK devices and test directly on the target board.

It is also important to note that GPAK devices offer the option to lock some or all of the access to the latches, and this locking mechanism will prevent ISD (or any access to the stored configuration for design security).

Conclusion

In-System Debug can allow the designer to do the debug of the GPAK configuration at the same time as the debug of the overall system. This approach is more flexible, and can result in a reduction in the total amount of time required to bring your product that includes GPAK from concept to production.

About the Author

Name: Nathan John

Background: Nathan John is the Director of Marketing at Silego Technology responsible for managing the GreenPAK product-line. In his spare time, he enjoys outdoor activities, including backpacking and photography.

Contact: appnotes@silego.com

Files

- AN-1125 In-System Debug for GreenPAK™ Devices with I2C Serial Communication.pdf- (449 KB)
- AN-1125.zip- (381 KB)

See full list of Application Notes