

Go Board iCE40 HX1K - SRAM Programming Investigation Report

Hardware Information

- **Board:** Go Board with iCE40 HX1K FPGA
- **Package:** VQ100
- **USB Interface:** FTDI FT2232HL (VID:0403, PID:6010)
- **Dual Interface Architecture:** Interface A (Flash), Interface B (SRAM)

Issue Summary

SRAM programming via Interface B completes successfully without errors but does not take effect. Flash programming via Interface A works correctly. The SRAM programming commands execute and report success, but the flash configuration remains active.

Test Environment

- **OS:** Windows 11
- **Toolchain:** OSS CAD Suite (icepack, iceprog)
- **Test Project:** Simple LED control (combinational logic, no clock)

Detailed Test Results

1. USB Device Detection

```
$ lsusb
Bus 001 Device 004: ID 0403:6010 Future Technology Devices International, Ltd
FT2232C/D/H Dual UART/FIFO IC
```

2. Flash Programming (Interface A) - WORKING ☒

```
$ iceprog -I A -v build/playground.bin
init..
cdone: high
reset..
cdone: high
programming..
sending 4096 bytes.
sending 4096 bytes.
[... additional blocks ...]
sending 3548 bytes.
VERIFY OK
cdone: high
Bye.
```

Result: LEDs show expected pattern (1:ON, 2:OFF, 3:ON, 4:OFF) ✓

3. SRAM Programming (Interface B) - NOT WORKING ✗

```
$ iceprog -I B -S -v build/playground.bin
init..
cdone: high
reset..
cdone: high
programming..
sending 4096 bytes.
sending 4096 bytes.
[... additional blocks ...]
sending 3548 bytes.
cdone: high
Bye.
```

Result: Command completes successfully, but LEDs maintain previous flash configuration ✗

4. Test Code Used

```
module playground
  (input i_Switch_1,
   input i_Switch_2,
   input i_Switch_3,
   input i_Switch_4,
   output o_LED_1,
   output o_LED_2,
   output o_LED_3,
   output o_LED_4);

  assign o_LED_1 = 1'b1;           // ON - Clear SRAM test
  assign o_LED_2 = 1'b0;           // OFF - Clear SRAM test
  assign o_LED_3 = 1'b1;           // ON - Clear SRAM test
  assign o_LED_4 = 1'b0;           // OFF - Clear SRAM test

endmodule
```

5. Pin Constraints (VQ100 Package)

```
set_io o_LED_1 56
set_io o_LED_2 57
set_io o_LED_3 59
set_io o_LED_4 60
set_io i_Switch_1 51
set_io i_Switch_2 52
```

```
set_io i_Switch_3 53
set_io i_Switch_4 54
```

Detailed Behavior Analysis

Expected SRAM Programming Behavior

- 1. SRAM programming should provide temporary configuration
- 2. Configuration should take effect immediately after programming
- 3. Configuration should be lost on power cycle
- 4. Should allow rapid development iteration

Observed Behavior

- 1. ☒ Interface B programming commands execute without errors
- 2. ☒ iceprog reports successful completion
- 3. ☒ **SRAM configuration does not take effect**
- 4. ☒ Flash configuration remains active after SRAM programming
- 5. ☒ Flash programming (Interface A) works correctly

Interface Comparison

Feature	Interface A (Flash)	Interface B (SRAM)
Command	iceprog -I A -v file.bin	iceprog -I B -S -v file.bin
Programming	<input checked="" type="checkbox"/> Works	<input checked="" type="checkbox"/> No Effect
Error Messages	None	None
iceprog Exit Code	0 (Success)	0 (Success)
Configuration Active	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> No
cdone Signal	High	High

Command Line Session Log

Programming Different Patterns to Test SRAM Override

Step 1: Program Pattern A to Flash (Interface A)

```
# Pattern: LED1=OFF, LED2=ON, LED3=OFF, LED4=ON
$ iceprog -I A -v build/playground.bin
[... successful flash programming ...]
# Result: Board shows Pattern A ☒
```

Step 2: Program Pattern B to SRAM (Interface B)

```
# Pattern: LED1=ON, LED2=OFF, LED3=ON, LED4=OFF
$ iceprog -I B -S -v build/playground.bin
[... successful SRAM programming ...]
# Expected: Board should show Pattern B
# Actual: Board still shows Pattern A ✖
```

Step 3: Verification with Flash Programming

```
# Program Pattern B to Flash (Interface A)
$ iceprog -I A -v build/playground.bin
[... successful flash programming ...]
# Result: Board shows Pattern B ✔
```

Questions for Manufacturer

1. **SRAM Programming Support:** Does this specific Go Board model support SRAM programming via Interface B?
2. **Alternative SRAM Commands:** Are there alternative commands or procedures required for SRAM programming on this board?
3. **Hardware Configuration:** Are there jumpers, switches, or configuration requirements to enable SRAM programming mode?
4. **Interface Documentation:** Is there specific documentation for the dual-interface programming modes on this board?
5. **Firmware Version:** Could the FTDI chip firmware version affect SRAM programming capability?

Technical Specifications Needed

- Complete programming interface documentation
- SRAM programming procedure (if supported)
- Any hardware configuration requirements
- Expected behavior differences between Interface A and Interface B
- Troubleshooting guide for SRAM programming issues

Development Impact

Currently using flash programming for development, which works reliably but:

- Longer development cycles (flash write time vs SRAM)
- More flash write cycles (wear concern for extensive development)
- Would benefit from SRAM programming for rapid iteration

Conclusion

The Go Board iCE40 HX1K works excellently with flash programming (Interface A) but SRAM programming (Interface B) does not function as expected. The board appears to only support persistent flash configuration, or requires additional setup/commands for SRAM programming that are not documented in standard iCE40 programming procedures.

Generated: December 19, 2025

Tools: OSS CAD Suite, iceprog

Board: Go Board iCE40 HX1K VQ100 with FT2232HL interface