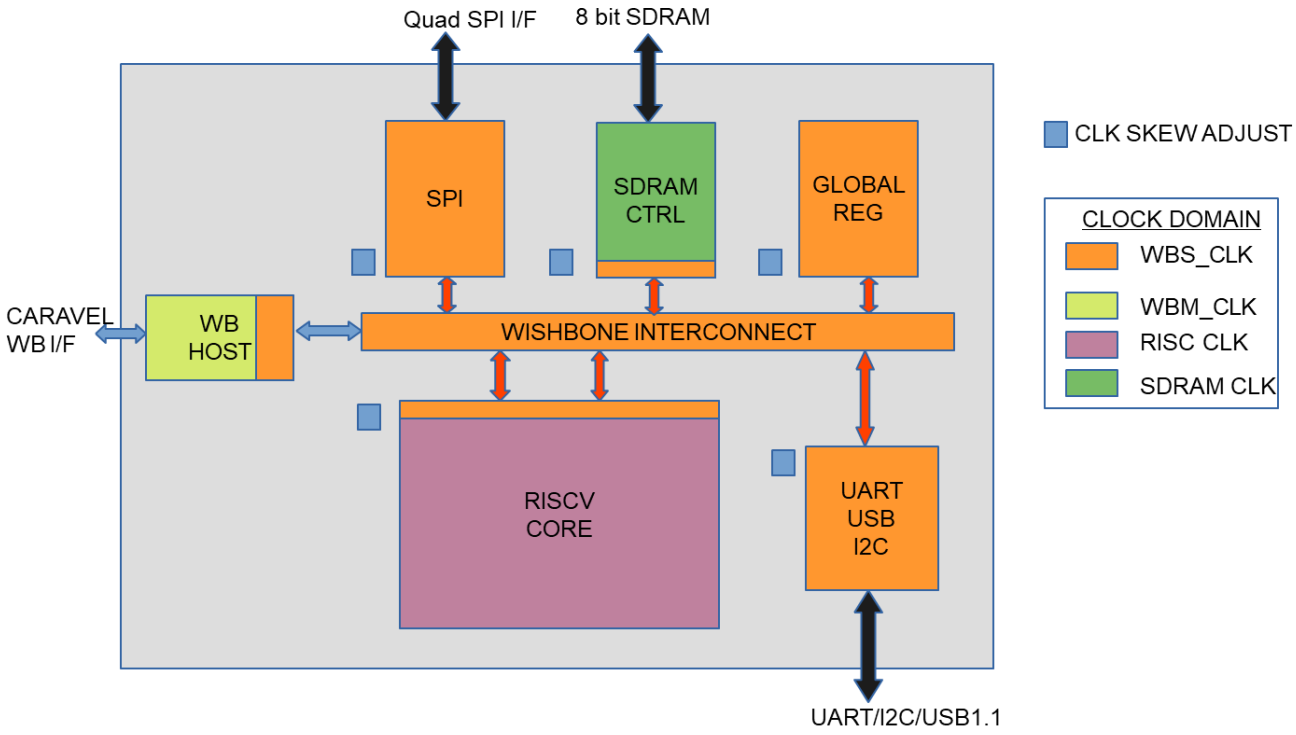
MPW 2 Silicon Bring Up Status

SWT Frame Location	SLOT	PROJECT	ID	Git Repo
B1	7	yiFive	20007	Link

Project Block Diagram:



Design includes, 32 Bit RISCV Core, Quad SPI, 8 Bit SDRAM, UART, I2C and USB 1.1 and wishbone inter-connect. Design uses hierarchy implementation strategy with 7 Macros and 4 clock domains. The design complexity is more than 100K cells.

Why I thought this chip will be dead on arrival for silicon bring-up:-

MPW-2 Critical Issues	Design Details	Is there is any hope?
Hierarchical Timing closure may not be accurate	1. This chip uses hierarchical timing closure 2. SOC has 7 Macros 3. 4 different clock domain	1. I have added each Macro associated with clock skew adjust macro and each clock skew macro has 16 clock adjustable option. 2. I have re-analysed top-level timing with latest MPW-7 scripts shared by efabless and it's shows the Hold timing are meting with adjusting clock-skew. But there was no common clock-skew adjust value works for all the three corners.
User Wishbone address space reduced from [0x3000 0000 - 0x3FFF FFFF] to [0x3000 0000 - 0x300F FFFF]	1. This project used wishbone Address above 0x3080_0000 to access the user logic of SPI/GLBL/SDRAM block. 2. Due to this bug I will not be access SPI/GLBL/SDRAM block using caravel wishbone i/f	1. User wb_host block register was still accessible through caravel wb i/f. 2. Wb_host includes critical register like reset, clock selection and clock adjust and they were able accessible through caravel wb i/f 3. SPI IP had ability to auto initialize external FLASH on reset de-assertion. 4. On reset removal RISC CORE will automatically try to fetch the program memory data from external Flash 0x200 location through SPI. 5. Standalone RTL simulation shows, RISC Core can boot up and initiate UART related transaction by just releasing reset without requirement of addition configuration from caravel wb-if.

Silicon Bring-up: Basic GPIO Requirements

Phase	GPIO	Feature
Basic Boot	GPIO[35-30] - User Flash I/F	RISC Core boot with SPI flash and initiate uart transaction without any data memory access
	GPIO[37-36] - User UART	
Full Boot	GPIO[29-0] - SDRAM I/F	RISC Core boot with data memory support and 32MB data memory program can be tested
	GPIO[35-30] - User Flash I/F	
	GPIO[37-36] - User UART	

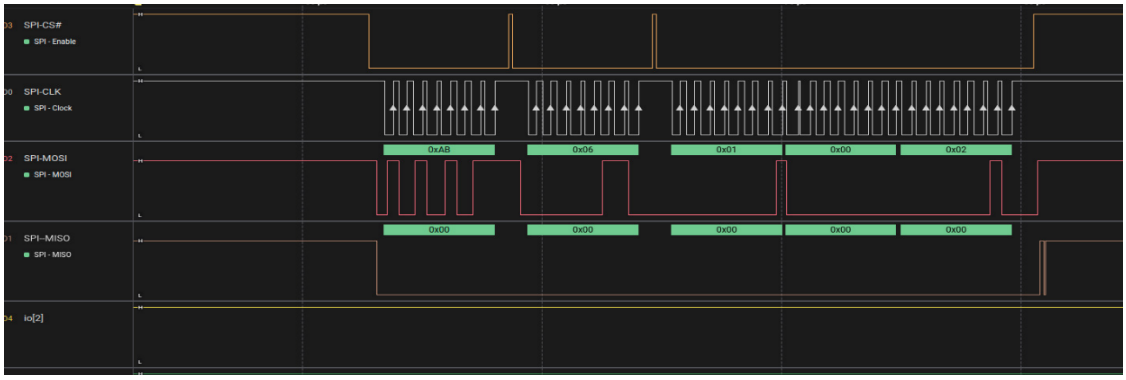
Silicon Bring-UP Status:

Baby Step-1: Identifying the working chip with GPIO[37:30] access

Chip-No	GPIO Low Working Range	GPIO High Working Range	Remark
1	0 to 18	35 to 37	Not Useful
2	0 to 13	30 to 37	Enough to try basic boot phase

Remark: All the below status are with Chip-No: 2

Baby Step-2: Remove the User Reset through caravel wishbone interface and check SPI init request at externa user SPI Interface



Logic analyzer capture at User Flash I/f (GPIO 35:30] shows that User SPI IP has initiated the flash memory initialization sequence and this matches with expected simulation waveform.

Summary: This indicates

1. Power hook-up to User SPI core is good.
2. Clock to SPI core is good.
3. User SPI reset removal working through caravel wishbone i/f.

Baby Step-3: In the design, I have connected some of the block internal status to caravel LA [127:0] port. I have updated script to read user LA value and dump it through caravel uart-tx port.

```
// Configure LA is input
reg_la0_oenb = reg_la0_iena = 0000000000;    // [31:0]
reg_la1_oenb = reg_la1_iena = 0x00000000;    // [63:32]
reg_la2_oenb = reg_la2_iena = 0x000000000;    // [95:64]
reg_la3_oenb = reg_la3_iena = 0x00000000;    // [127:96]
reg_uart_enable = 1;
print_reg(reg_la0_data_in);
print_reg(reg_la1_data_in);
```

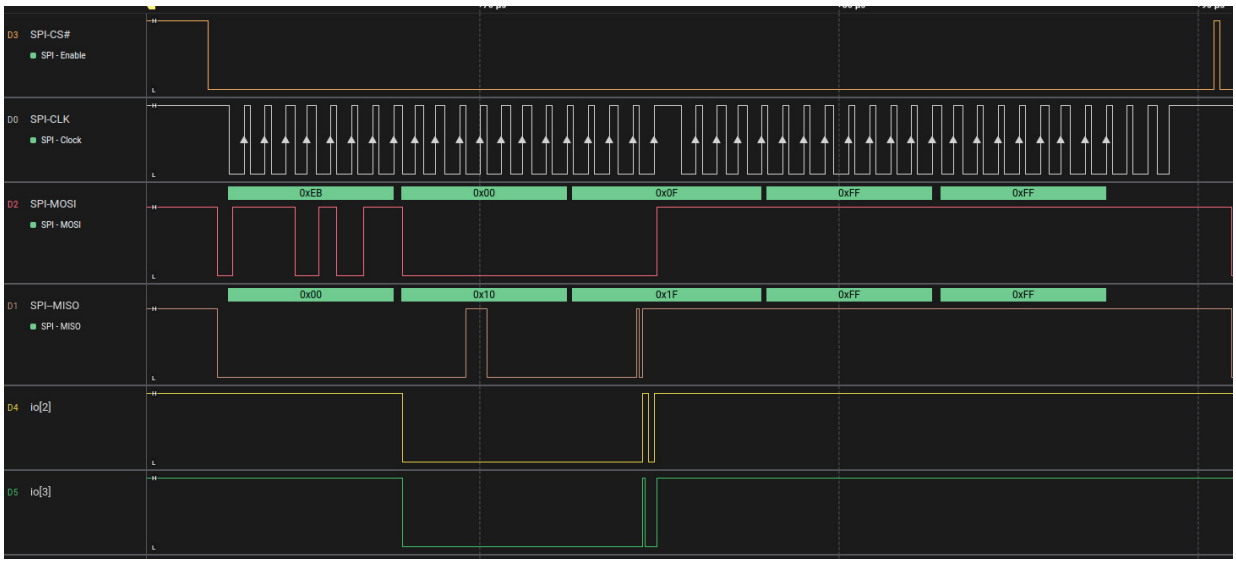
Result shows that LA value `reg_la0_data_in = 0x04000140 && reg_la1_data_in = 0x84000000 matches with simulation.`

Summary: This indicates

1> RISC Program counter is at reset phase with 0x200
2> SPI initiation is completed.
3> SRAM initiation is completed.
4> RISC Core + SDARM Clock and power hook-up are good

Baby Step-4: At user SPI interface after initial SPI init sequence, there should be RISC boot access with SPI Opcode : 0xEB followed by 0x200 address, which was not appearing at external interface.

After little adjustments on the clock_skew adjust at RISC core interface, I have notice that Quad SPI access Opcode: 0xEB followed with 0x200 address and it matches with expected simulation step.



Summary: This indicates

1> RISC To Wb-Interconnect Write access good
2> Wb-Interconnect to SPI Write access is good

Baby Step-5:

From the boot flow angle there should be continuous SPI access, but I have noticed only two boot read access; which indicates SPI => RISV Read path is broken.

After continue working on adjusting the clock skew around SPI/WB Inter-connect, Now I see there is continuous SPI Read access 0x1C0. 0x1C0 is address indicate that RISC core is not receiving the expected read-data and entering in to exception routine.

Summary: This indicates

1> RISC To WB-Interconnect Write & Response access good
2> Wb-Interconnect to SPI Write & Response access good
3> SPI to RISV Read data path is broken.

What are next steps: -

1. Planning to build new board with FPGA instead on nucleo board, which will give much better control on the timing adjustment at the user SPI interface.
2. As GPIO rise & fall time are bad, planning to reduce the user SPI clock further from the 2.5Mhz
3. Try to play around with User core voltage.
4. Try to play around with clock skew