

[SoC Lab] Lab5

tags: SoC Lab, SOC Design

Team 13

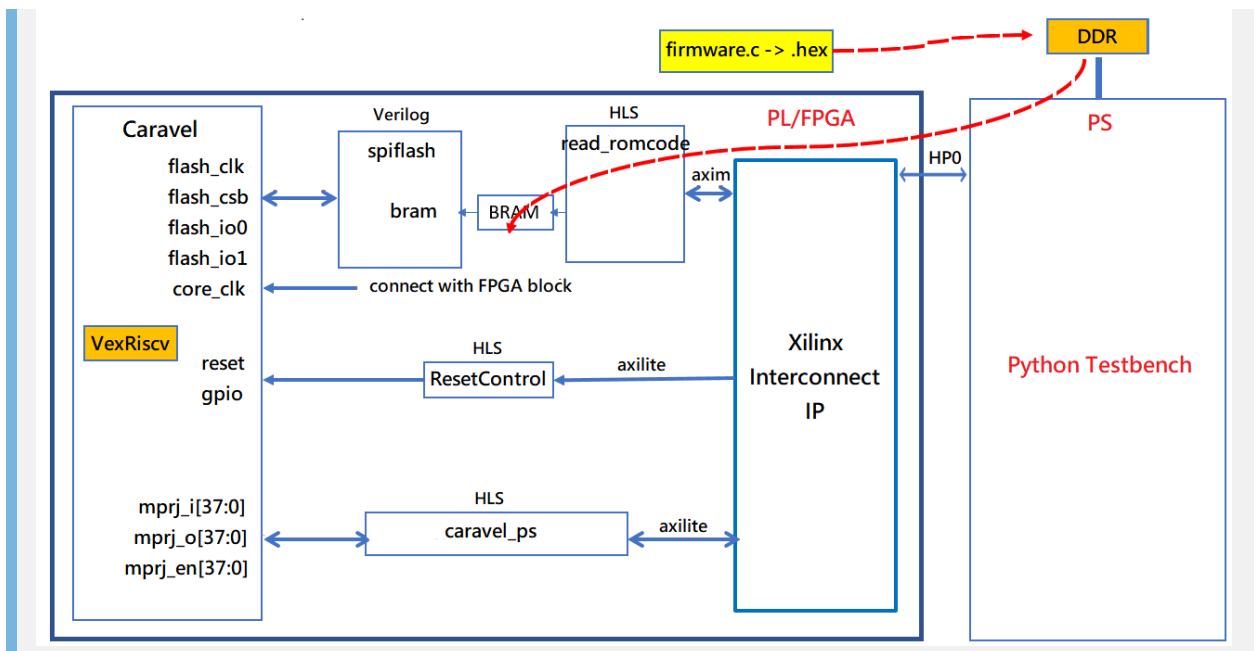
| Student ID | Name |
|------------|------|
| 311551095 | 林聖博 |
| 312551174 | 張祐誠 |

附上此篇Hackmd Link: https://hackmd.io/Ae-A2Z_hTxirA8ZjISL8mQ

- [SoC Lab] Lab5
 - Lab 5 Spec
 - Practical Step
 - Block diagram
 - FPGA utilization
 - run_vivado
 - run_vivado_gcd
 - Explain the function of IP in this design
 - HLS
 - Verilog
 - Screenshot of Execution result on all workload
 - Study caravel_fpga.ipynb, and be familiar with caravel SoC control flow
 - Github link

Lab 5 Spec

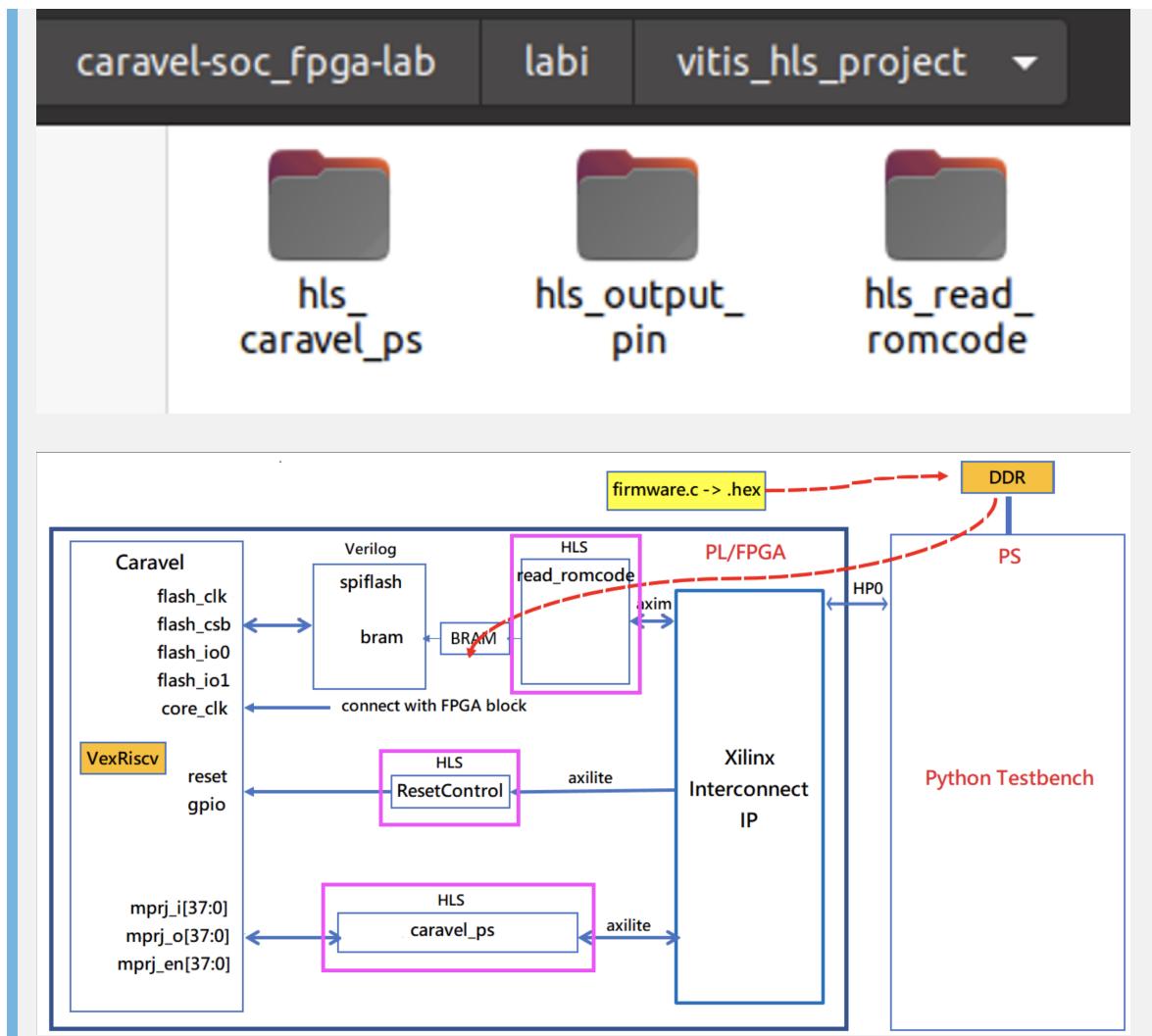
- https://github.com/bol-edu/caravel-soc_fpga-lab/tree/main/lab1



Practical Step

1. Run vitis

- 執行 `run_vitis.sh`
- 利用腳本執行 Vitis HLS 產生 IP，包含：
 - caravel_ps, output_pin, read_romcode (粉色框)



2. Run vivado

- 執行 `run_vivado.sh`, `run_vivado_gcd.sh`
- 利用腳本執行 vivado 進行 IP 整合與 Block Design

■ `run_vivado.sh`

```
INFO: [Project 1-111] Unisim Transformation Summary:
A total of 5 instances were transformed.
RAM32M => RAM32M (RAM32(x6), RAMS32(x2)): 4 instances
RAM32X1D => RAM32X1D (RAM32(x2)): 1 instance

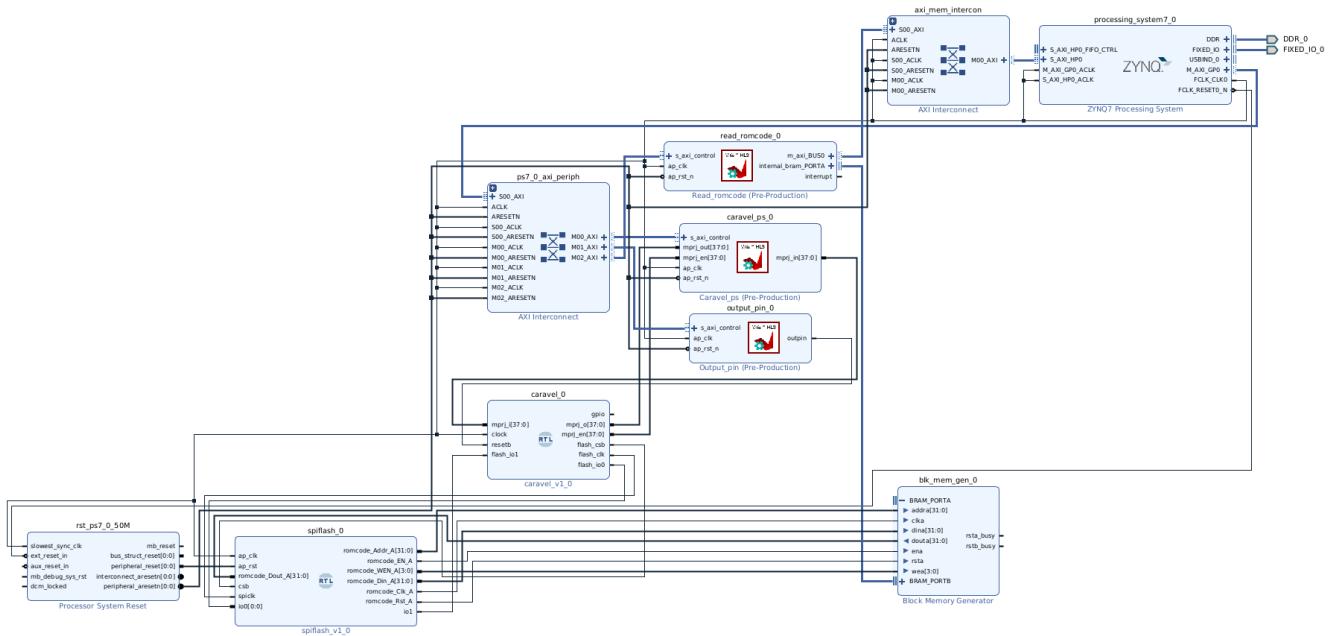
open_run: Time (s): cpu = 00:00:17 ; elapsed = 00:00:17 . Memory (MB): peak = 3309.934 ; gain = 222.074 ; free physical = 9346 ; free virtual = 13635
# report_timing_summary -file timingreport.txt
INFO: [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min max.
INFO: [Timing 38-191] Multithreading enabled for timing update using a maximum of 4 CPUs
# exit
INFO: [Common 17-206] Exiting Vivado at Sat Nov 25 12:45:20 2023...
=====
vivado complete
=====
ubuntu@ubuntu2004:~/lab/caravel-soc_fpga-lab/lab1$ ./run_vivado.sh
```

■ `run_vivado_gcd.sh`

```
INFO: [Project 1-111] Unisim Transformation Summary:
A total of 5 instances were transformed.
RAM32M => RAM32M (RAM32(x6), RAMS32(x2)): 4 instances
RAM32X1D => RAM32X1D (RAM32(x2)): 1 instance

open_run: Time (s): cpu = 00:00:19 ; elapsed = 00:00:22 . Memory (MB): peak = 3295.953 ; gain = 224.105 ; free physical = 12526 ; free virtual = 14112
# report_timing_summary -file timingreport.txt
INFO: [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min_max.
INFO: [Timing 38-191] Multithreading enabled for timing update using a maximum of 4 CPUs
# exit
INFO: [Common 17-206] Exiting Vivado at Thu Nov 23 14:30:38 2023...
=====
vivado complete
=====
ubuntu@ubuntu2004:~/lab/caravel-soc_fpga-lab/lab1$ ./run_vivado_gcd.sh
```

Block Diagram



FPGA Utilization

run_vivado

1. Slice Logic

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|------------------------|------|-------|------------|-----------|-------|
| Slice LUTs | 5327 | 0 | 0 | 53200 | 10.01 |
| LUT as Logic | 5149 | 0 | 0 | 53200 | 9.68 |
| LUT as Memory | 178 | 0 | 0 | 17400 | 1.02 |
| LUT as Distributed RAM | 18 | 0 | | | |
| LUT as Shift Register | 160 | 0 | | | |
| Slice Registers | 6051 | 0 | 0 | 106400 | 5.69 |
| Register as Flip Flop | 6051 | 0 | 0 | 106400 | 5.69 |
| Register as Latch | 0 | 0 | 0 | 106400 | 0.00 |
| F7 Muxes | 169 | 0 | 0 | 26600 | 0.64 |
| F8 Muxes | 47 | 0 | 0 | 13300 | 0.35 |

2. Slice Logic Distribution

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|--|------|-------|------------|-----------|-------|
| Slice | 2303 | 0 | 0 | 13300 | 17.32 |
| SLICEL | 1625 | 0 | | | |
| SLICEM | 678 | 0 | | | |
| LUT as Logic | 5149 | 0 | 0 | 53200 | 9.68 |
| using 05 output only | 0 | | | | |
| using 06 output only | 4205 | | | | |
| using 05 and 06 | 944 | | | | |
| LUT as Memory | 178 | 0 | 0 | 17400 | 1.02 |
| LUT as Distributed RAM | 18 | 0 | | | |
| using 05 output only | 0 | | | | |
| using 06 output only | 2 | | | | |
| using 05 and 06 | 16 | | | | |
| LUT as Shift Register | 160 | 0 | | | |
| using 05 output only | 41 | | | | |
| using 06 output only | 81 | | | | |
| using 05 and 06 | 38 | | | | |
| Slice Registers | 6051 | 0 | 0 | 106400 | 5.69 |
| Register driven from within the Slice | 2815 | | | | |
| Register driven from outside the Slice | 3236 | | | | |
| LUT in front of the register is unused | 1978 | | | | |
| LUT in front of the register is used | 1258 | | | | |
| Unique Control Sets | 312 | | 0 | 13300 | 2.35 |

* * Note: Available Control Sets calculated as Slice * 1, Review the Control Sets Report for more information regarding control sets.

3. Memory

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|----------------|------|-------|------------|-----------|-------|
| Block RAM Tile | 6 | 0 | 0 | 140 | 4.29 |
| RAMB36/FIFO* | 3 | 0 | 0 | 140 | 2.14 |
| RAMB36E1 only | 3 | | | | |
| RAMB18 | 6 | 0 | 0 | 280 | 2.14 |
| RAMB18E1 only | 6 | | | | |

- Physical Synthesis Optimizations

| Summary of Physical Synthesis Optimizations | | | | | | | |
|---|-------------|---------------|----------------------|------------|------------|----------|--|
| Optimization | Added Cells | Removed Cells | Optimized Cells/Nets | Dont Touch | Iterations | Elapsed | |
| LUT Combining | 0 | 147 | 147 | 0 | 1 | 00:00:00 | |
| Retime | 0 | 0 | 0 | 0 | 1 | 00:00:00 | |
| Very High Fanout | 0 | 0 | 0 | 0 | 1 | 00:00:00 | |
| DSP Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 | |
| Shift Register to Pipeline | 0 | 0 | 0 | 0 | 0 | 00:00:00 | |
| Shift Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 | |
| BRAM Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 | |
| URAM Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 | |
| Dynamic/Static Region Interface Net Replication | 0 | 0 | 0 | 0 | 1 | 00:00:00 | |
| Total | 0 | 147 | 147 | 0 | 4 | 00:00:00 | |

- Max delay paths

| Max Delay Paths | |
|----------------------------|--|
| Slack (MET) : | 2.339ns (required time - arrival time) |
| Source: | design_1_i/processing_system7_0/inst/PS7_i/FCLKCLK[0] (clock source 'clk_fpga_0' {rise@0.000ns fall@10.000ns period=20.000ns}) |
| Destination: | design_1_i/caravel_0/inst/housekeeping/wb_dat_o_reg[15]/D (rising edge-triggered cell FDSE clocked by clk_fpga_0 {rise@0.000ns fall@10.000ns period=20.000ns}) |
| Path Group: | clk_fpga_0 |
| Path Type: | Setup (Max at Slow Process Corner) |
| Requirement: | 10.000ns (clk_fpga_0 rise@20.000ns - clk_fpga_0 fall@10.000ns) |
| Data Path Delay: | 10.028ns (logic 1.460ns (14.560%) route 8.568ns (85.440%)) |
| Logic Levels: | 8 (BUFG=1 LUT3=2 LUT5=1 LUT6=3 MUXF7=1) |
| Clock Path Skew: | 2.732ns (DCD - SCD + CPR) Destination Clock Delay (DCD): 2.732ns = (22.733 - 20.000) Source Clock Delay (SCD): 0.000ns = (10.000 - 10.000) Clock Pessimism Removal (CPR): 0.000ns |
| Clock Uncertainty: | 0.302ns $((TSJ^2 + TIJ^2)^{1/2} + DJ) / 2 + PE$ |
| Total System Jitter (TSJ): | 0.071ns |
| Total Input Jitter (TIJ): | 0.600ns |
| Discrete Jitter (DJ): | 0.000ns |
| Phase Error (PE): | 0.000ns |

- Slack met

| | | (clock clk_fpga_0 rise edge) | 20.000 | 20.000 | r | |
|---------------|----------------------|------------------------------|--------|---------|---|---|
| PS7_X0Y0 | PS7 | net (fo=1, routed) | 0.000 | 20.000 | r | design_1_i/processing_system7_0/inst/PS7_i/FCLKCLK[0] |
| BUFCTRL_X0Y22 | BUFG (Prop_bufg_I_0) | net (fo=5029, routed) | 0.091 | 21.088 | r | design_1_i/processing_system7_0/inst/buffer_fclk_clk_0.FCLK_CLK_0_BUFG[0] |
| SLICE_X54Y46 | FDSE | clock pessimism | 1.553 | 21.179 | r | design_1_i/caravel_0/inst/housekeeping/clock |
| | | clock uncertainty | 0.000 | 22.733 | r | design_1_i/caravel_0/inst/housekeeping/wb_dat_o_reg[15]/C |
| | | FDSE (Setup_fdse_C_D) | -0.302 | 22.430 | | |
| | | required time | -0.064 | 22.366 | | design_1_i/caravel_0/inst/housekeeping/wb_dat_o_reg[15] |
| | | arrival time | | -20.028 | | |
| | | slack | | 2.339 | | |

- Report Cell Usage

| Report Cell Usage: | | |
|--------------------|-------------------------------|-------|
| | Cell | Count |
| 1 | design_1_auto_pc | 2 |
| 3 | design_1_auto_us | 1 |
| 4 | design_1_blk_mem_gen_0 | 1 |
| 5 | design_1_caravel_0 | 1 |
| 6 | design_1_caravel_ps_0 | 1 |
| 7 | design_1_output_pin_0 | 1 |
| 8 | design_1_processing_system7_0 | 1 |
| 9 | design_1_read_romcode_0 | 1 |
| 10 | design_1_RST_ps7_0_50M | 1 |
| 11 | design_1_spiflash_0 | 1 |
| 12 | design_1_xbar | 1 |

run_vivado_gcd

1. Slice Logic

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|------------------------|------|-------|------------|-----------|-------|
| Slice LUTs | 6457 | 0 | 0 | 53200 | 12.14 |
| LUT as Logic | 6279 | 0 | 0 | 53200 | 11.80 |
| LUT as Memory | 178 | 0 | 0 | 17400 | 1.02 |
| LUT as Distributed RAM | 18 | 0 | | | |
| LUT as Shift Register | 160 | 0 | | | |
| Slice Registers | 6082 | 0 | 0 | 106400 | 5.72 |
| Register as Flip Flop | 6082 | 0 | 0 | 106400 | 5.72 |
| Register as Latch | 0 | 0 | 0 | 106400 | 0.00 |
| F7 Muxes | 168 | 0 | 0 | 26600 | 0.63 |
| F8 Muxes | 47 | 0 | 0 | 13300 | 0.35 |

2. Slice Logic Distribution

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|--|------|-------|------------|-----------|-------|
| Slice | 2541 | 0 | 0 | 13300 | 19.11 |
| SLICEL | 1820 | 0 | | | |
| SLICEM | 721 | 0 | | | |
| LUT as Logic | 6279 | 0 | 0 | 53200 | 11.80 |
| using O5 output only | 0 | | | | |
| using O6 output only | 5296 | | | | |
| using O5 and O6 | 983 | | | | |
| LUT as Memory | 178 | 0 | 0 | 17400 | 1.02 |
| LUT as Distributed RAM | 18 | 0 | | | |
| using O5 output only | 0 | | | | |
| using O6 output only | 2 | | | | |
| using O5 and O6 | 16 | | | | |
| LUT as Shift Register | 160 | 0 | | | |
| using O5 output only | 41 | | | | |
| using O6 output only | 81 | | | | |
| using O5 and O6 | 38 | | | | |
| Slice Registers | 6082 | 0 | 0 | 106400 | 5.72 |
| Register driven from within the Slice | 2865 | | | | |
| Register driven from outside the Slice | 3217 | | | | |
| LUT in front of the register is unused | 1827 | | | | |
| LUT in front of the register is used | 1390 | | | | |
| Unique Control Sets | 311 | | 0 | 13300 | 2.34 |

* * Note: Available Control Sets calculated as Slice * 1, Review the Control Sets Report for more information regarding control sets.

3. Memory

| Site Type | Used | Fixed | Prohibited | Available | Util% |
|----------------|------|-------|------------|-----------|-------|
| Block RAM Tile | 6 | 0 | 0 | 140 | 4.29 |
| RAMB36/FIFO* | 3 | 0 | 0 | 140 | 2.14 |
| RAMB36E1 only | 3 | | | | |
| RAMB18 | 6 | 0 | 0 | 280 | 2.14 |
| RAMB18E1 only | 6 | | | | |

• Physical Synthesis Optimizations

| Summary of Physical Synthesis Optimizations | | | | | | |
|---|-------------|---------------|----------------------|------------|------------|----------|
| Optimization | Added Cells | Removed Cells | Optimized Cells/Nets | Dont Touch | Iterations | Elapsed |
| LUT Combining | 0 | 143 | 143 | 0 | 1 | 00:00:00 |
| Retime | 0 | 0 | 0 | 0 | 1 | 00:00:00 |
| Very High Fanout | 0 | 0 | 0 | 0 | 1 | 00:00:00 |
| DSP Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 |
| Shift Register to Pipeline | 0 | 0 | 0 | 0 | 0 | 00:00:00 |
| Shift Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 |
| BRAM Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 |
| URAM Register | 0 | 0 | 0 | 0 | 0 | 00:00:00 |
| Dynamic/Static Region Interface Net Replication | 0 | 0 | 0 | 0 | 1 | 00:00:00 |
| Total | 0 | 143 | 143 | 0 | 4 | 00:00:00 |

- Max delay paths

| Max Delay Paths | |
|--------------------------------|---|
| Slack (MET) : | 3.954ns (required time - arrival time) |
| Source: | design_1_i/caravel_0/inst/mprij/mprij/seq_gcd/b_q_reg[5]/C (rising edge-triggered cell FDCE clocked by clk_fpga_0 {rise@0.000ns fall@50.000ns period=100.000ns}) |
| Destination: | design_1_i/caravel_0/inst/mprij/mprij/seq_gcd/b_q_reg[30]/D (rising edge-triggered cell FDCE clocked by clk_fpga_0 {rise@0.000ns fall@50.000ns period=100.000ns}) |
| Path Group: | clk_fpga_0 |
| Path Type: | Setup (Max at Slow Process Corner) |
| Requirement: | 100.000ns (clk_fpga_0 rise@100.000ns - clk_fpga_0 rise@0.000ns) |
| Data Path Delay: | 94.530ns (logic 59.216ns (62.643%) route 35.314ns (37.357%)) |
| Logic Levels: | 315 (CARRY4=281 LUT1=1 LUT3=29 LUT5=1 LUT6=3) |
| Clock Path Skew: | -0.095ns (DCD - SCD + CPR) |
| Destination Clock Delay (DCD): | 5.222ns = (105.222 - 100.000) |
| Source Clock Delay (SCD): | 5.918ns |
| Clock Pessimism Removal (CPR): | 0.601ns |
| Clock Uncertainty: | 1.500ns $((TSJ^2 + TIJ^2)^{1/2} + DJ) / 2 + PE$ |
| Total System Jitter (TSJ): | 0.071ns |
| Total Input Jitter (TIJ): | 3.000ns |
| Discrete Jitter (DJ): | 0.000ns |
| Phase Error (PE): | 0.000ns |

- Slack met

| | | (clock clk_fpga_0 rise edge) | 100.000 | 100.000 | r |
|----------------|-----------------------|------------------------------|---------|----------|--|
| PS7_X0Y0 | PS7 | | 0.000 | 100.000 | r |
| | net (fo=1, routed) | | 1.088 | 101.088 | design_1_i/processing_system7_0/inst/PS7_i/FCLKCLK[0] |
| BUFGCTRL_X0Y22 | BUFG (Prop_bufg_I_0) | | 0.091 | 101.179 | r design_1_i/processing_system7_0/inst/buffer_fclk_clk_0.FCLK_CLK_0_BUFG/0 |
| | net (fo=5028, routed) | | 1.790 | 102.970 | design_1_i/caravel_0/inst/soc/core/clock |
| SLICE_X50Y50 | LUT3 (Prop_lut3_I2_0) | | 0.100 | 103.070 | r design_1_i/caravel_0/inst/soc/core/a_q[31]_i_2/0 |
| | net (fo=1, routed) | | 0.512 | 103.581 | design_1_i/caravel_0/inst/soc/core/CLK |
| BUFGCTRL_X0Y20 | BUFG (Prop_bufg_I_0) | | 0.091 | 103.672 | r design_1_i/caravel_0/inst/soc/core/CLK_BUFG_inst/0 |
| | net (fo=97, routed) | | 1.550 | 105.222 | design_1_i/caravel_0/inst/mprij/mprij/seq_gcd/CLK |
| SLICE_X54Y38 | FDCE | | | | r design_1_i/caravel_0/inst/mprij/mprij/seq_gcd/b_q_reg[30]/C |
| | clock pessimism | | 0.601 | 105.824 | |
| | clock uncertainty | | 1.500 | 104.323 | |
| SLICE_X54Y38 | FDCE (Setup_fdce_C_D) | | 0.079 | 104.402 | design_1_i/caravel_0/inst/mprij/mprij/seq_gcd/b_q_reg[30] |
| | required time | | | 104.402 | |
| | arrival time | | | -100.448 | |
| | slack | | | | 3.954 |

- Report Cell Usage

| Report Cell Usage: | | |
|--------------------|-------------------------------|-------|
| | Cell | Count |
| 1 | design_1_auto_pc | 2 |
| 3 | design_1_auto_us | 1 |
| 4 | design_1_blk_mem_gen_0 | 1 |
| 5 | design_1_caravel_0 | 1 |
| 6 | design_1_caravel_ps_0 | 1 |
| 7 | design_1_output_pin_0 | 1 |
| 8 | design_1_processing_system7_0 | 1 |
| 9 | design_1_read_romcode_0 | 1 |
| 10 | design_1_RST_ps7_0_10M | 1 |
| 11 | design_1_spiflash_0 | 1 |
| 12 | design_1_xbar | 1 |

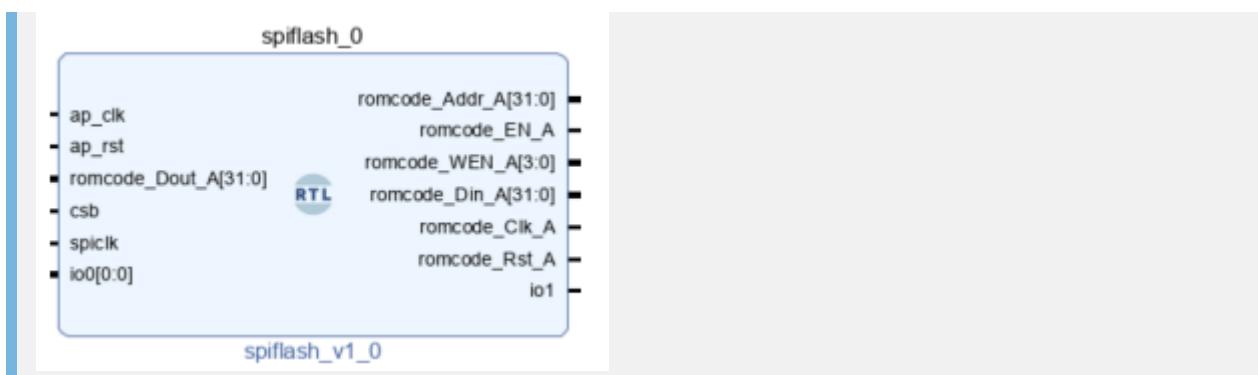
Explain the function of IP in this design

HLS

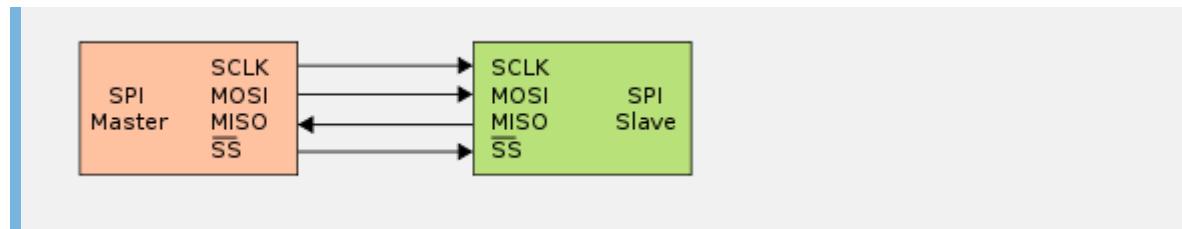
- **read_romcode:**
 - Copy PS dram buffer to BRAM base on the size of binary file.
 - Through the AXI-Master.
 - Triggered by `ipReadROMCODE.write(0x00, 1)`.
- **ResetControl:**
 - Use for reset the cpu program.
 - Triggered by `ipOUTPIN.write(0x10, 1)`.
 - After resetting, the RISC-V start running , which fetch the code (instruction) from the spiflash (0x1000_0000). In the meanwhile, when cpu read from spiflash, spiflash receive the signal and ack to cpu data that from Bram.
- **caravel_ps:**
 - Cpu put the data on mprj pins and the **caravel_ps** will pass the value to PS with MMIO.
 - PS will monitor the mprj to get the value.

Verilog

- **spiflash:**



- `csb` as enable when `csb = 0`
- Use shift buffer for output the value to cpu
- Get the data from Bram and pass to cpu
- **SPI Description:**



| Master Output Slave Input (MOSI) | Master Output Slave Input (MOSI) | Serial Clock (SCK) | SS |
|-------------------------------------|-------------------------------------|-------------------------------------|------------------------------|
| From Master to Slave | From Slave to Master | Shared clk between Master and Slave | Select the Slave to transfer |

Screenshot of Execution result on all workload

Run these workload on caravel FPGA

- **counter_wb.hex**

- `mpkj_io` 的 `0x1c` 腳位輸出值為 `0xab610008`

```
In [5]: # Create np with 8K/4 (4 bytes per index) size and be initiled to 0
rom_size_final = 0

# Allocate dram buffer will assign physical address to ip ipReadROMCODE
npROM = allocate(shape=(ROM_SIZE >> 2,), dtype=np.uint32)

# Initial it by 0
for index in range (ROM_SIZE >> 2):
    npROM[index] = 0

npROM_index = 0
npROM_offset = 0
fiROM = open("counter_wb.hex", "r+")
# fiROM = open("counter_la.hex", "r+")
# fiROM = open("gcd_la.hex", "r+")
```

```
In [10]: # Check MPRJ_IO input/out/en
# 0x10 : Data signal of ps_mprj_in
#         bit 31~0 - ps_mprj_in[31:0] (Read/Write)
# 0x14 : Data signal of ps_mprj_in
#         bit 5~0 - ps_mprj_in[37:32] (Read/Write)
#         others - reserved
# 0x1c : Data signal of ps_mprj_out
#         bit 31~0 - ps_mprj_out[31:0] (Read)
# 0x20 : Data signal of ps_mprj_out
#         bit 5~0 - ps_mprj_out[37:32] (Read)
#         others - reserved
# 0x34 : Data signal of ps_mprj_en
#         bit 31~0 - ps_mprj_en[31:0] (Read)
# 0x38 : Data signal of ps_mprj_en
#         bit 5~0 - ps_mprj_en[37:32] (Read)
#         others - reserved

print ("0x10 = ", hex(ipPS.read(0x10)))
print ("0x14 = ", hex(ipPS.read(0x14)))
print ("0x1c = ", hex(ipPS.read(0x1c)))
print ("0x20 = ", hex(ipPS.read(0x20)))
print ("0x34 = ", hex(ipPS.read(0x34)))
print ("0x38 = ", hex(ipPS.read(0x38)))

0x10 = 0x0
0x14 = 0x0
0x1c = 0xab610008
0x20 = 0x2
0x34 = 0xfffff7
0x38 = 0x37
```

- **counter_la.hex**

- `mpoj_io` 的 `0x1c` 腳位輸出值為 `0xab5153d1`

```
In [5]: # Create np with 8K/4 (4 bytes per index) size and be initiled to 0
rom_size_final = 0

# Allocate dram buffer will assign physical address to ip ipReadROMCODE
npROM = allocate(shape=(ROM_SIZE >> 2,), dtype=np.uint32)

# Initial it by 0
for index in range (ROM_SIZE >> 2):
    npROM[index] = 0

npROM_index = 0
npROM_offset = 0
# fiROM = open("counter_wb.hex", "r+")
fiROM = open("counter_la.hex", "r+")
# fiROM = open("gcd_la.hex", "r+")

In [9]: # Check MPRJ_IO input/out/en
# 0x10 : Data signal of ps_mprj_in
#         bit 31~0 - ps_mprj_in[31:0] (Read/Write)
# 0x14 : Data signal of ps_mprj_in
#         bit 5~0 - ps_mprj_in[37:32] (Read/Write)
#         others - reserved
# 0x1c : Data signal of ps_mprj_out
#         bit 31~0 - ps_mprj_out[31:0] (Read)
# 0x20 : Data signal of ps_mprj_out
#         bit 5~0 - ps_mprj_out[37:32] (Read)
#         others - reserved
# 0x34 : Data signal of ps_mprj_en
#         bit 31~0 - ps_mprj_en[31:0] (Read)
# 0x38 : Data signal of ps_mprj_en
#         bit 5~0 - ps_mprj_en[37:32] (Read)
#         others - reserved

print ("0x10 = ", hex(ipPS.read(0x10)))
print ("0x14 = ", hex(ipPS.read(0x14)))
print ("0x1c = ", hex(ipPS.read(0x1c)))
print ("0x20 = ", hex(ipPS.read(0x20)))
print ("0x34 = ", hex(ipPS.read(0x34)))
print ("0x38 = ", hex(ipPS.read(0x38)))

0x10 = 0x0
0x14 = 0x0
0x1c = 0xab5153d1
0x20 = 0x0
0x34 = 0x0
0x38 = 0x3f
```

- gcd_la.hex

- mprj_io 的 0x1c 腳位輸出值為 0xab510041

```
In [5]: # Create np with 8K/4 (4 bytes per index) size and be initiled to 0
rom_size_final = 0

# Allocate dram buffer will assign physical address to ip ipReadROMCODE
npROM = allocate(shape=(ROM_SIZE >> 2,), dtype=np.uint32)

# Initial it by 0
for index in range (ROM_SIZE >> 2):
    npROM[index] = 0

npROM_index = 0
npROM_offset = 0
# fiROM = open("counter_wb.hex", "r+")
# fiROM = open("counter_la.hex", "r+")
fiROM = open("gcd_la.hex", "r+")

In [9]: 1 # Check MPRJ_IO input/out/en
2 # 0x10 : Data signal of ps_mprj_in
3 #           bit 31~0 - ps_mprj_in[31:0] (Read/Write)
4 # 0x14 : Data signal of ps_mprj_in
5 #           bit 5~0 - ps_mprj_in[37:32] (Read/Write)
6 #           others - reserved
7 # 0x1c : Data signal of ps_mprj_out
8 #           bit 31~0 - ps_mprj_out[31:0] (Read)
9 # 0x20 : Data signal of ps_mprj_out
10 #           bit 5~0 - ps_mprj_out[37:32] (Read)
11 #           others - reserved
12 # 0x34 : Data signal of ps_mprj_en
13 #           bit 31~0 - ps_mprj_en[31:0] (Read)
14 # 0x38 : Data signal of ps_mprj_en
15 #           bit 5~0 - ps_mprj_en[37:32] (Read)
16 #           others - reserved
17
18 print ("0x10 = ", hex(ipPS.read(0x10)))
19 print ("0x14 = ", hex(ipPS.read(0x14)))
20 print ("0x1c = ", hex(ipPS.read(0x1c)))
21 print ("0x20 = ", hex(ipPS.read(0x20)))
22 print ("0x34 = ", hex(ipPS.read(0x34)))
23 print ("0x38 = ", hex(ipPS.read(0x38)))

0x10 = 0x0
0x14 = 0x0
0x1c = 0xab510041
0x20 = 0x0
0x34 = 0x0
0x38 = 0x3f
```

Study caravel_fpga.ipynb, and be familiar with caravel SoC control flow

本次 Lab5 的核心主要是理解 Caravel SoC 控制流程，運用不同的 IP（Intellectual Property）來完成資料傳輸和驗證。並集中在 Block Design 階段，產生出實際 IP 使用 FPGA 和 Python code 來執行與驗證設計。

首先，使用 `read_romcode` IP 將 `.hex` 由 DDR(PS side) 傳輸到 BRAM 中。接著，透過 `ResetControl` IP (`output_pin`)，在 PS (Processing System) 和 Caravel SoC 上的 `mprj_io` 之間進行溝通。`ResetControl` 主要能夠控制 `reset` 信號，進而控制 Caravel SoC 的 `reset pin` 是處於啟動狀態 (`assert`) 還是停止狀態 (`de-assert`)。

此外，透過 `mprj_io` 和 `caravel_ps` IP，能夠在執行結束後與 Python code 進行溝通，並使用 AXI Lite interface。此機制允許我們可以從硬體端接收執行結果並在軟體端進行分析。也因此可以利用 Python code 設定 `ResetControl` IP 中的 `reset` 訊號，觸發 Caravel SoC 的啟動。當 Firmware 執行完畢後，可以釋放 `reset` 訊號，讓系統進入待機狀態。

與 Lab4 不同，Lab5 重點在於使用 Python code 而非 Simulator 來執行測試和驗證。以實際應用的測試環境 (Remote FPGA) 完成資料的傳輸和溝通。

總結本次實驗，Lab5 主要讓我們熟悉對 FPGA、SoC 控制流程、以及 IP 合成與整合的理解，透過實際操作和驗證，並有著先前 Lab 1~4 的經驗，對於這次 Lab 有更加深刻地體會，了解 SoC 設計個階段的意義與原理，並進一步了解軟硬體整合的重要性。

Github link

- <https://github.com/Sheng08/Soc-Lab-lab5>
-

补充

- 由於在執行 `run_vivado_gcd.sh` 會因為虛擬機資源不夠導致無法成功完成，因此需更改 `vvd_caravel_fpga_gcd.tcl` 內中進行 `write_bitstream` 的 job 數
 - 由原本 `-job 16 -> -job 8`

```

1697  # Will add for bitstream generated
1698  update_compile_order -fileset sources_1
1699  launch_runs impl_1 -to_step write_bitstream -job 8
1700  wait_on_run impl_1

```