

Overall system :

此 lab 的 kernel function 很簡單，即是輸入2個 input，然後返回相乘的結果。不過要利用 AXILite adaptor 中的 Register 當作 input 傳遞的工具，傳入 IN1、IN2 進去，得到的結果也用 AXILite adaptor 的 Register 存起來。然後 Register 中，input 的傳入是利用 AXILite 的 AW 和 W，將資料寫入 Register。而讀取返回的結果則是利用 AXILite 的 AR 和 R，將 output 從 Register 讀出來。

What is observed & learned :

學到 AXILite 協議的內容，知道 master 和 slave 是怎麼溝通的。也知道怎麼將東西包成 IP，相比於暑假自學時，也更了解這次實驗背後的原理。

Screen dump :

## 1. Performance

```
=====
== Performance Estimates
=====
+ Timing:
  * Summary:
  +-----+-----+-----+-----+
  | Clock | Target | Estimated | Uncertainty |
  +-----+-----+-----+-----+
  | ap_clk | 10.00 ns | 6.912 ns | 2.70 ns |
  +-----+-----+-----+-----+

+ Latency:
  * Summary:
  +-----+-----+-----+-----+-----+-----+
  | Latency (cycles) | Latency (absolute) | Interval | Pipeline |
  | min | max | min | max | min | max | Type |
  +-----+-----+-----+-----+-----+-----+
  | 3 | 3 | 30.000 ns | 30.000 ns | 4 | 4 | no |
  +-----+-----+-----+-----+-----+-----+

+ Detail:
  * Instance:
  N/A

  * Loop:
  N/A
```

## 2. Utilization

```
=====
== Utilization Estimates
=====
* Summary:
+-----+-----+-----+-----+-----+-----+
| Name | BRAM_18K | DSP | FF | LUT | URAM |
+-----+-----+-----+-----+-----+-----+
| DSP | - | - | - | - | - |
| Expression | - | - | - | - | - |
| FIFO | - | - | - | - | - |
| Instance | 0 | 3 | 309 | 282 | - |
| Memory | - | - | - | - | - |
| Multiplexer | - | - | - | 25 | - |
| Register | - | - | 100 | - | - |
+-----+-----+-----+-----+-----+-----+
| Total | 0 | 3 | 409 | 307 | 0 |
+-----+-----+-----+-----+-----+-----+
| Available | 280 | 220 | 106400 | 53200 | 0 |
+-----+-----+-----+-----+-----+-----+
| Utilization (%) | 0 | 1 | ~0 | ~0 | 0 |
+-----+-----+-----+-----+-----+-----+
```

## 3. Interface

```
=====
== Interface
=====
* Summary:
```

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
ls_axi_control_AWVALID	in	1	s_axi	control	pointer
ls_axi_control_AWREADY	out	1	s_axi	control	pointer
ls_axi_control_AWADDR	in	6	s_axi	control	pointer
ls_axi_control_WVALID	in	1	s_axi	control	pointer
ls_axi_control_WREADY	out	1	s_axi	control	pointer
ls_axi_control_WDATA	in	32	s_axi	control	pointer
ls_axi_control_WSTRB	in	4	s_axi	control	pointer
ls_axi_control_ARVALID	in	1	s_axi	control	pointer
ls_axi_control_ARREADY	out	1	s_axi	control	pointer
ls_axi_control_ARADDR	in	6	s_axi	control	pointer
ls_axi_control_RVALID	out	1	s_axi	control	pointer
ls_axi_control_RREADY	in	1	s_axi	control	pointer
ls_axi_control_RDATA	out	32	s_axi	control	pointer
ls_axi_control_RRESP	out	2	s_axi	control	pointer
ls_axi_control_BVALID	out	1	s_axi	control	pointer
ls_axi_control_BREADY	in	1	s_axi	control	pointer
ls_axi_control_BRESP	out	2	s_axi	control	pointer
lap_clk	in	1	ap_ctrl_none	multip_2num	return value
lap_rst_n	in	1	ap_ctrl_none	multip_2num	return value

```
=====
```

## 4. Co-simulation transcript /waveform

```

multip_2num_csim.log
檔案 編輯 檢視
-----
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
-----|
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
-----
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
-----
>> Test passed!
-----
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] ***** CSIM finish *****
第 76 行, 第 25 欄

```

## 5. Jupyter Notebook execution results

```

sys.path.append('/home/aaaaaa/')
os.environ['XILINX_XRT'] = '/usr'
from pynq import Overlay

if __name__ == "__main__":
    print("Entry:", sys.argv[0])
    print("System argument(s):", len(sys.argv))

    print("Start of \" " + sys.argv[0] + "\"")

    ol = Overlay("/home/xilinx/jupyter_notebooks/Multip2Num.bit")
    regIP = ol.multip_2num_0

    for i in range(9):
        print("=====")
        for j in range(9):
            regIP.write(0x10, i + 1)
            regIP.write(0x18, j + 1)
            Res = regIP.read(0x20)
            print(str(i + 1) + " * " + str(j + 1) + " = " + str(Res))
        print("=====")
    print("Exit process")

```

Entry: /usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel\_launcher.py  
 System argument(s): 3  
 Start of "/usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel\_launcher.py"  
 =====  
 1 \* 1 = 1  
 1 \* 2 = 2  
 1 \* 3 = 3  
 1 \* 4 = 4  
 =====  
 8 \* 1 = 8  
 8 \* 2 = 16  
 8 \* 3 = 24  
 8 \* 4 = 32  
 8 \* 5 = 40  
 8 \* 6 = 48  
 8 \* 7 = 56  
 8 \* 8 = 64  
 8 \* 9 = 72  
 =====  
 9 \* 1 = 9  
 9 \* 2 = 18  
 9 \* 3 = 27  
 9 \* 4 = 36  
 9 \* 5 = 45  
 9 \* 6 = 54  
 9 \* 7 = 63  
 9 \* 8 = 72  
 9 \* 9 = 81  
 =====  
 Exit process

In [ ]: