Lab 3: Parameterized Ping-pong Counter

15:30, October 31, 2016

Design Concepts

Design Part

1. 這次 design 部分只要加入 mode port 去進行 Up/Down/Pingpong counter 三者的模式切換，大抵上我使用 data flow model 去進行操作，程式碼如下：

```
assign expression = (hold) ? result:
                    (mode == 2'b00 && result    < `UPPER_BOUND) ? result + 1:
                    (mode == 2'b00 && result == `UPPER_BOUND) ? `LOWER_BOUND:
                    (mode == 2'b01 && result    > `LOWER_BOUND) ? result-1:
                    (mode == 2'b01 && result == `LOWER_BOUND) ? `UPPER_BOUND:
                    (dir == 0 && result < `UPPER_BOUND)          ? result + 1:
                    (dir == 0 && result == `UPPER_BOUND)         ? `LOWER_BOUND:
                    (dir == 1 && result > `LOWER_BOUND)           ? result - 1:
                    (dir == 1 && result == `LOWER_BOUND)          ? `UPPER_BOUND:
                                                                    `LOWER_BOUND;
```

比較需要注意的部分是因為要對 upperbound 和 lowerbound 做參數化，因此就不能單純定義好寬度，讓 lowerbound=0 而 upperbound=lowerbound-1'b1 這種讓他溢位產生 max value，而且在 result 達到 upperbound 的時候要讓他下一次的值變成 lowerbound，在 down counter mode 的時候也是同樣道理只是方向相反。

Testbench Part

1. Testbench 部分也主要根據教授提供的 code 為架構，使用兩個 task 去做 pattern 和 golden comparison value 的 apply，但是因為 based on lab1 的 pingpong counter 架構去做修改，在第一個 pattern apply 後 out 便是 1，和 golden 的 0 會不一樣，也就是教授在作業上講到的一個 clk 的 shift，因此我在完全沒有 pattern apply 的時候先去比對第 0 筆資料，然後 for loop 再從 i=1 去開始做 pattern apply 以及 golden value comparison，便解決了 one clock shift 的問題。
2. 此外還有 cmp 和 err 系列參數，我使用額外的 always block 去做判斷，在每次 clock positive edge triggered 的時候去比對輸出值和 golden value 的區別，如果相等則 asserted 1，然後再去判斷如果是 0 (代表輸出和 golden value 不一樣)，則就讓 err + 1，以 out 為例：

```
always @(posedge clk) begin
   cmp_out = out == gold_out ? 1 : 0;
   if (cmp_out == 0) err_out = err_out + 1;
end
```

Header Part

1. Header 部分就很直觀，主要都是要是沒有 define 則 define 一個預設的值給他，default value 如下表所示：

WIDTH	UPPER_BOUND	LOWER_BOUND	PATTERN	GOLDEN
4	16	0	"pp02_w8_pat.dat"	"pp02_w8_gold.dat"
DEBUG	PERIOD	DELAY	FSDB	PATTERN_COUNT
1	8	1	"Majority.fsdb"	1024

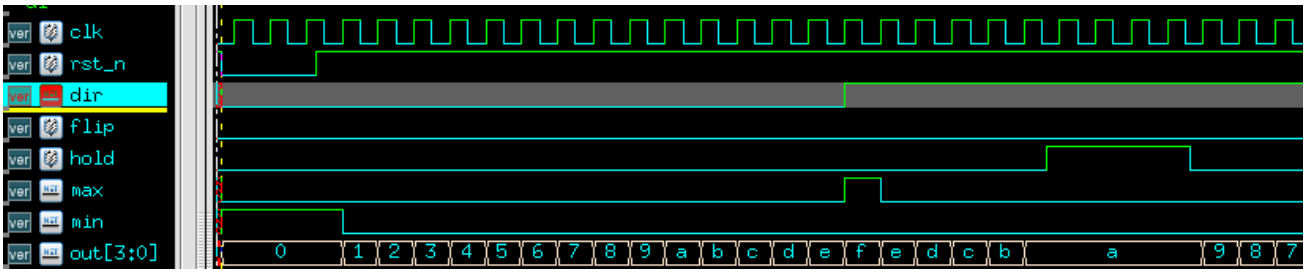
Makefile Part

1. Makefile 主要是做參數化值的更改，更改的結果將會在 stimulation pattern 顯示。

Stimulation Patterns

[iclab79@ic22 ~/hw3]$ make

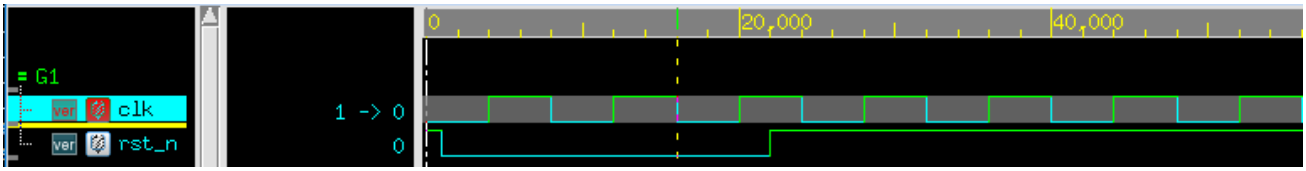
對於直接單純的使用 make 來做預設值的更改，我們可以看到單純的 4-bit pingpong counter



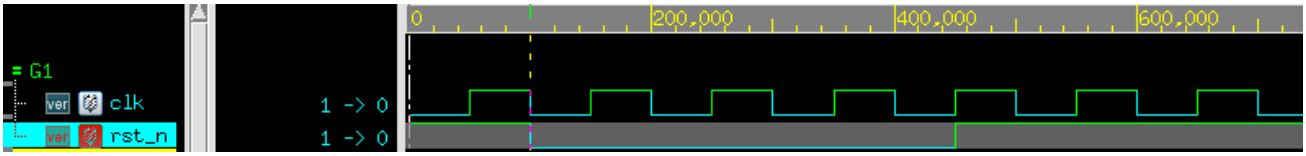
[iclab79@ic22 ~/hw3]$ make PERIOD=100 DELAY=100

修改 Period 及 Delay 的部分可以從圖中看到，CLK 的頻率有了大幅度的增長，上者為原始時間，下者為更改後的時間。

Origin Period & Delay

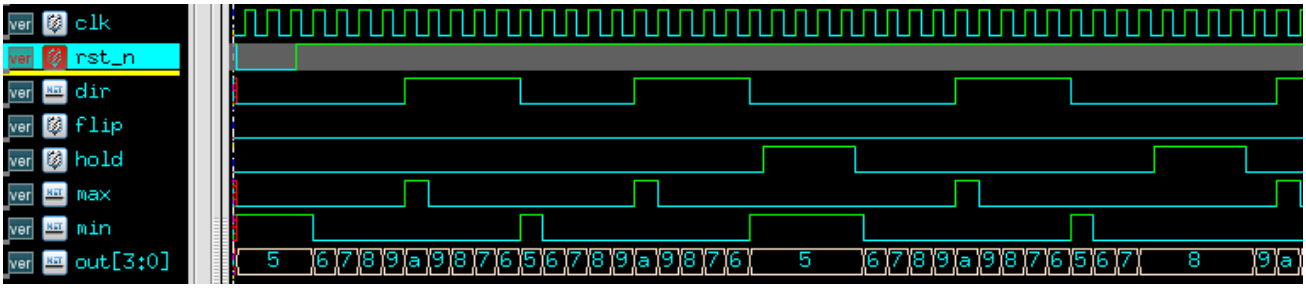


Modified Period & Delay



```
[iclab79@ic22 ~/hw3]$ make    UPPER_BOUND=10 LOWER_BOUND=5
```

我們也可以藉由 make 來更改 upperbound 以及 lowerbound，下圖為 5~10 之間的 pingpong counter，max 和 min 也會因為 upper/lower bound 的更改也一併更改




```
[iclab79@ic22 ~/hw3]$ make    WIDTH=8 UPPER_BOUND=255 LOWER_BOUND=0 DEBUG=2
```

預設的 DEBUG 為 1，使用 DEBUG Level =2 便可以看到每個細項的詳細資訊，輸出如下所示：

```
success at         1019!
success at         1020!
success at         1021!
success at         1022!
success at         1023!
Signal out: error count =           0
Signal dir: error count =           0
Signal max: error count =           0
Signal min: error count =           0
Pattern file: pp02_w8_pat.dat
Repsonse file: pp02_w8_gold.dat
```

```
[iclab79@ic22 ~/hw3]$ make    WIDTH=8 UPPER_BOUND=255 LOWER_BOUND=0
```

若不指定 DEBUG Level，則預設的輸出資料僅包含 header 和 summary 如下所示：

```
DEBUG LEVEL:           1
Signal out: error count =           0
Signal dir: error count =           0
Signal max: error count =           0
Signal min: error count =           0
Pattern file: pp02_w8_pat.dat
Repsonse file: pp02_w8_gold.dat
```

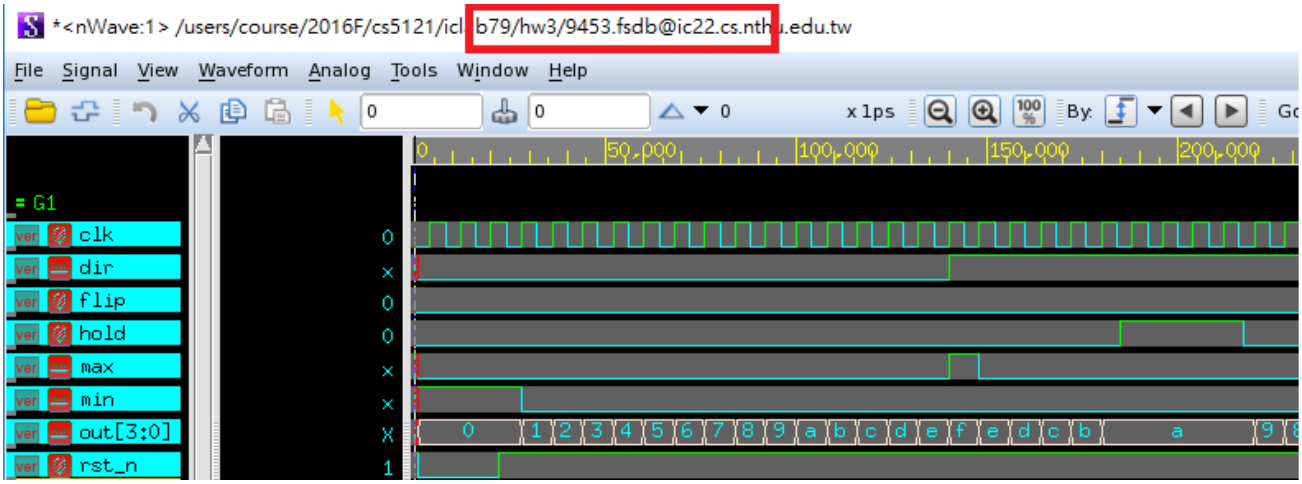
```
[iclab79@ic22 ~/hw3]$ make    WIDTH=8 UPPER_BOUND=255 LOWER_BOUND=0 DEBUG=0
```

如果使用 DEBUG Level=0，則不顯示任何資料，但我在這裡多設定一個$display 來看目前使用的 debug level。

```
DEBUG LEVEL:           0
Simulation complete via $finish(1) at time 8209 NS + 0
./lab03_pingpong_t.v:112        $finish;
ncsim> exit
[iclab79@ic22 ~/hw3]$
```

```
[iclab79@ic22 ~/hw3]$ make FSDB=9453.fsdb
```

自訂義 fsdb 檔名的部分，在輸入完指令後也可以利用產生的自定義 .fsdb 檔開啟 nWave，如下圖所示：




```
[iclab79@ic22 ~/hw3]$ make PATTERN=pp03_w7_pat.dat GOLDEN=pp03_w7_golden.dat
```

Make 來修改 patten 和 golden 檔的部分也可以從 DEBUG Level 1 中的 header 來確認是否有正確的輸入，如下圖所示：

```
DEBUG LEVEL:           1
Signal out: error count =           0
Signal dir: error count =           0
Signal max: error count =           0
Signal min: error count =           0
Pattern file: pp03_w7_pat.dat
Repsonse file: pp03_w7_gold.dat
```

Lab Review

這次的參數化可以使 Testbench 不只在由上面給予的值進行模擬，也可以做更靈活的自定義參數化，因為幾乎都是新的東西，因此在學習上面只能從網路、講義以及對助教的詢問上來進行著手，在程式撰寫過程總認為老師講義的編排實在沒有很優，上課跳來跳去外加講義的不夠詳細讓自己在撰寫上遇到了很多困難，因此很抱歉在詢問上可能問了太多細節的東西，還請助教原諒，不過很感謝助教總是會回答我的問題，這次的遲交只有一天 Delay 都要多虧助教的協助，再次感謝吳岳騏助教！