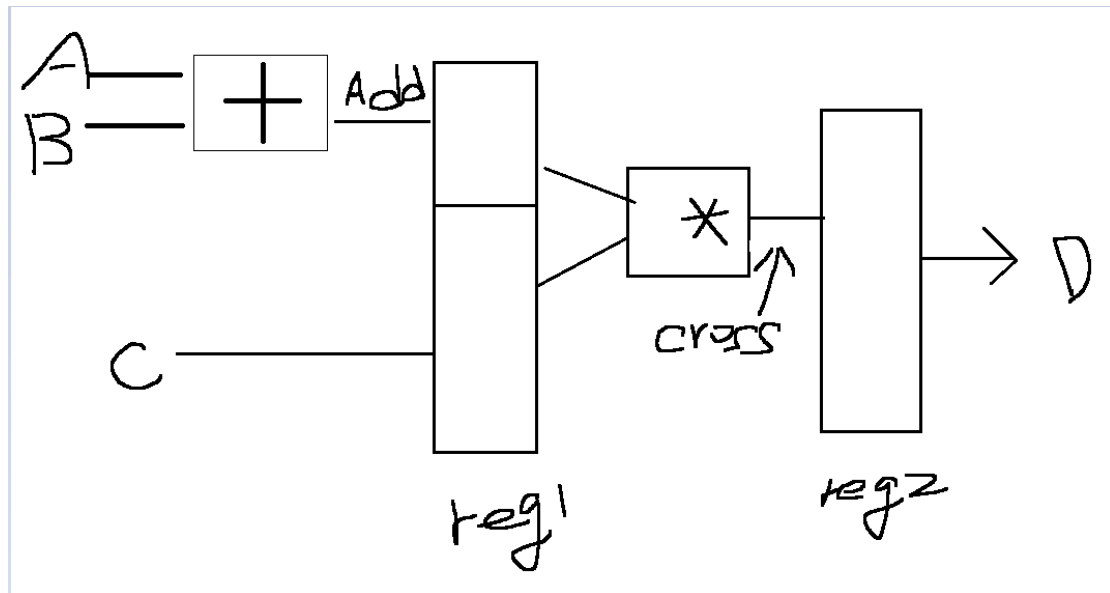


# 硬體描述語言作業二

B023040018 傅正安

主檔案名稱: pipelineMAD.v

以下是架構圖



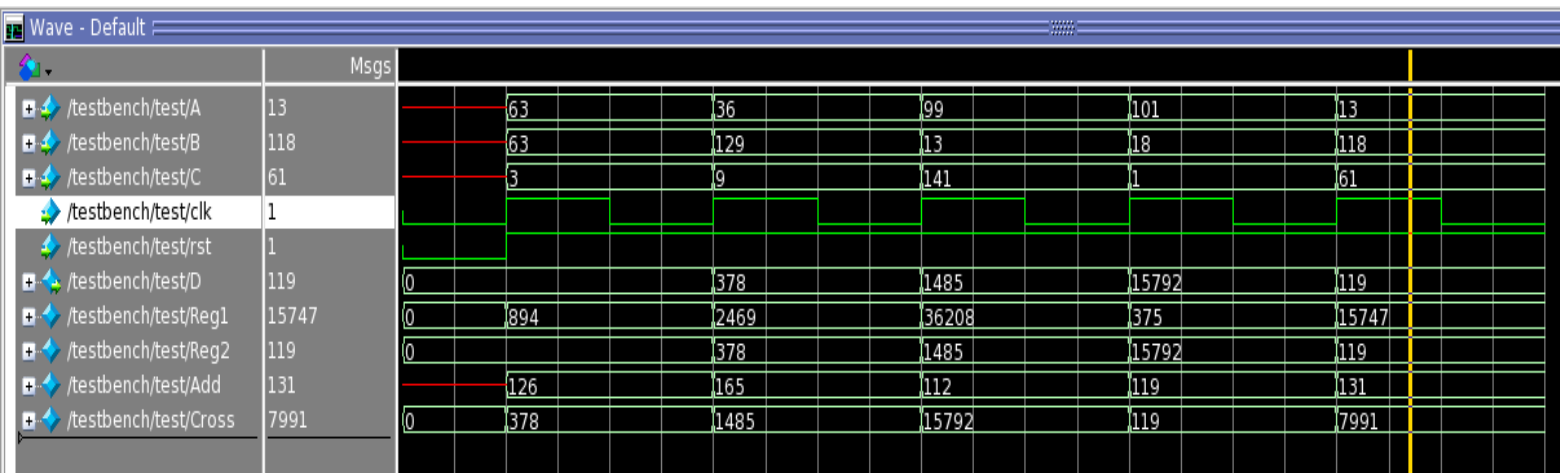
程式碼解說

前面先把加法器跟乘法器的輸出結果、以及 D 跟 register 的部分用 assign 的方式連結起來。

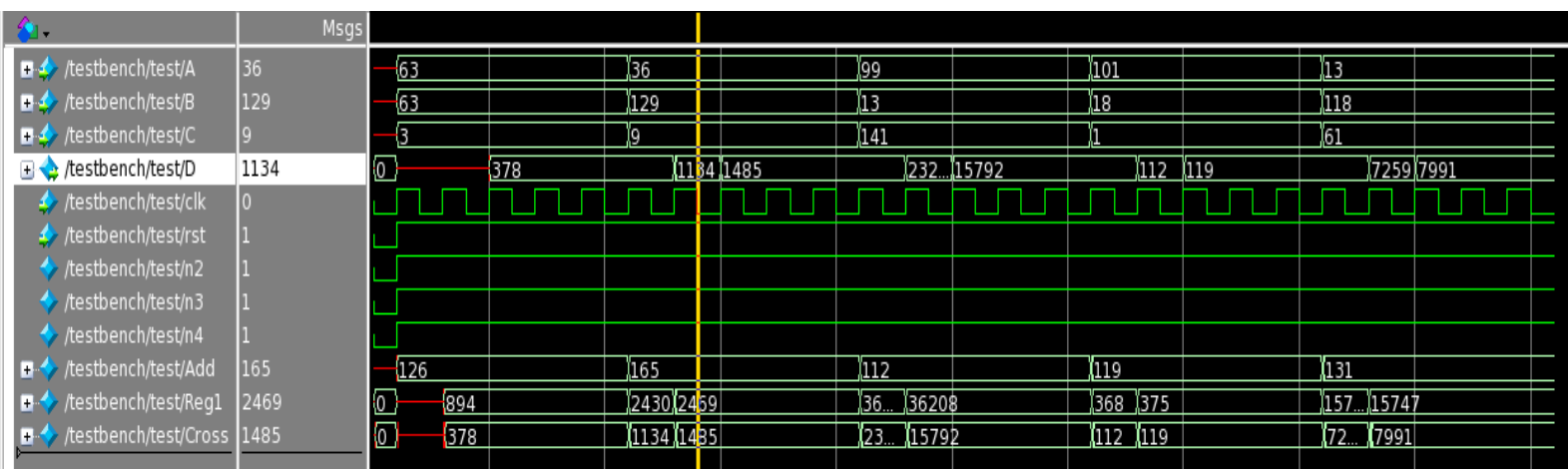
第一個 always 的部分，當遇到 `reset = 0` 的時候，將第一個 register 的內容歸零，若是 `reset = 1`，把 `A+B` 的結果存到 register 的前半部分，`C` 存到後半部分。

第二個 register 的部分，當遇到 `reset = 0` 的時候，將第二個 register 的內容歸零，若是 `reset = 1`，把 `(A+B)*C` 的結果存到第二個 register。

RTL 驗證



Gate level 驗證



## Area report

```
*****
Report : area
Design : pipeline
Version: H-2013.03-SP5
Date   : Thu Oct 27 00:05:26 2016
*****

Library(s) Used:

    saed90nm_typ (File: /CBDK/SAED90_EDK/SAED_EDK90nm/Digital_Standard_cell_Library/synopsys/models/saed90nm_typ.db)

Number of ports:          42
Number of nets:           86
Number of cells:          37
Number of combinational cells:  3
Number of sequential cells:  32
Number of macros/black boxes:   0
Number of buf/inv:         3
Number of references:       4

Combinational area:      2227.507185
Buf/Inv area:           105.062403
Noncombinational area:   1032.192017
Macro/Black Box area:    0.000000
Net Interconnect area:   3022.031978

Total cell area:         3259.699202
Total area:              6281.731180
```

## Time report

```
*****
Report : timing
        -path full
        -delay max
        -max_paths 1
        -sort_by group
Design : pipeline
Version: H-2013.03-SP5
Date   : Thu Oct 27 00:04:40 2016
*****

Operating Conditions: BEST   Library: saed90nm_min
Wire Load Model Mode: enclosed

Startpoint: Reg1_reg[10]
             (rising edge-triggered flip-flop clocked by clk)
Endpoint:   Reg2_reg[15]
             (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type:  max

Des/Clust/Port      Wire Load Model      Library
-----
pipeline            tc8000                saed90nm_typ
pipeline_DW_mult_0  8000                  saed90nm_typ

Point                                     Incr      Path
-----
clock clk (rise edge)                    0.00      0.00
clock network delay (ideal)               0.00      0.00
Reg1_reg[10]/CLK (DFFARX1)                0.00      0.00 r
Reg1_reg[10]/Q (DFFARX1)                  0.19      0.19 f
mult_13/b[2] (pipeline_DW_mult_0)         0.00      0.19 f
mult_13/U147/ZN (INVX0)                   0.07      0.25 r
mult_13/U159/QN (NOR2X0)                  0.06      0.31 f
mult_13/U57/SO (FADDX1)                   0.09      0.41 r
mult_13/U14/CO (FADDX1)                   0.11      0.51 r
mult_13/U13/CO (FADDX1)                   0.10      0.61 r
mult_13/U12/CO (FADDX1)                   0.10      0.71 r
mult_13/U11/CO (FADDX1)                   0.10      0.81 r
mult_13/U10/CO (FADDX1)                   0.10      0.90 r
mult_13/U9/CO (FADDX1)                    0.10      1.00 r
mult_13/U8/CO (FADDX1)                    0.10      1.10 r
mult_13/U7/CO (FADDX1)                    0.10      1.19 r
mult_13/U6/CO (FADDX1)                    0.10      1.29 r
mult_13/U5/CO (FADDX1)                    0.10      1.39 r
mult_13/U4/CO (FADDX1)                    0.10      1.49 r
mult_13/U3/CO (FADDX1)                    0.10      1.58 r
mult_13/U2/CO (FADDX1)                    0.08      1.67 r
mult_13/product[15] (pipeline_DW_mult_0)  0.00      1.67 r
Reg2_reg[15]/D (DFFARX1)                  0.00      1.67 r
data arrival time                          1.67

clock clk (rise edge)                    10.00     10.00
clock network delay (ideal)               0.00     10.00
Reg2_reg[15]/CLK (DFFARX1)                0.00     10.00 r
library setup time                       -0.09      9.91
data required time                        9.91

-----
data required time                        9.91
data arrival time                       -1.67
```

Gate level 驗證觀察到的現象：

我一開始在寫 testbench 的時候沒注意到加法器跟乘法器本身的硬體延遲，所以一開始的結果是前一個階段的  $A+B$  去乘上下一個階段的  $C$ ，造成結果錯誤，於是我把給值的延遲弄長，正確的結果會出現，但是正確的出現之後還是會有短時間出現前一個階段的  $A+B$  去乘上下一個階段的  $C$ ，再把延遲稍微縮短還是會出現，這部分還想不到解決方法。

寫這份作業遇到的困難：

1. 對於 blocking non-blocking 理解不夠深
2. 寫的 CODE 丟進 design-compiler 不給過(原因就是第一點不夠熟悉)
3. 沒注意到硬體本身延遲的部分，導致結果錯誤。

解決方法

1. 上網查資料
2. 自己慢慢嘗試