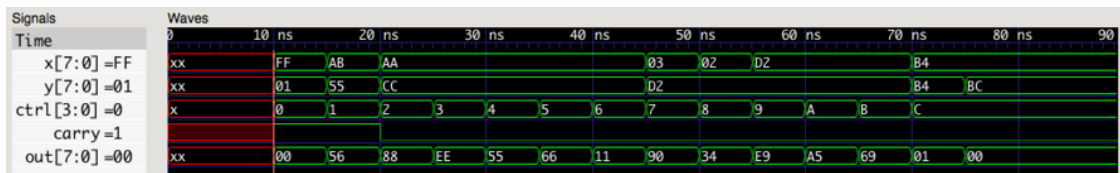


## 1. 8-bit arithmetic logic unit

### (3) Testbench

本次在 testbench 中依序對 ALU 的每個功能進行檢查，方法為比對 ALU 輸出和人工計算出的答案是否相同，每個功能測試一筆輸入。其中加法和減法都使用會產生進位/借位的測資，而位元運算都使用 0xAA 和 0xCC 當作測資（A 和 C 的位元運算包含所有 4 種可能，即 00, 01, 10, 11），equal 指令則測試了等於和不等於，若有錯誤則用 \$display 輸出 FAIL，否則輸出 PASS。

RTL 和 Behavioral 兩種實作的波形結果相同，如下圖：



## 2. 8x8 Register File

### (2) Testbench

在 testbench 中，可以使用 \$random 或 \$urandom 產生隨機輸入，同時可以使用另一個 reg 陣列來進行模擬，再比對 register file 的輸出和模擬的結果是否相同。本次 testbench 中模擬了 100000 筆隨機輸入，隨機的部分包含 RW、RX、RY、WEN、busW。

通過 testbench 的話會顯示以下訊息：

```
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
*****
register_file testbench passed
*****
Simulation complete via $finish(1) at time 1000010 NS + 0
./register_file_tb.v:78      #(CYCLE) $finish;
ncsim> exit
```

## 3. Simple Calculator

以下為通過 testbench 的畫面：

```
[REG#4 = sll 0000_0001 REG#4]
[REG#7 = add REG#7 REG#4]
.... passed.
9: shift & summation, REG#7 = A*B
[REG#5 = and REG#1 REG#5]
[REG#5 = sll 0000_0010 REG#5]
[REG#7 = add REG#7 REG#5]
.... passed.
10: shift & summation, REG#7 = A*B
[REG#6 = and REG#1 REG#6]
[REG#6 = sll 0000_0011 REG#6]
[REG#7 = add REG#7 REG#6]
.... passed.
Calculation results: 13 * 12 = 156
*****
**          **      /|_/_|
**          **      / 0,0 |
** Congratulations !! **      /_____|
** All Patterns Passed!! **    / ^ ^ ^ \ |
**          **    [ ^ ^ ^ ^ |w|
**          **    \m__m_|_|
*****
Simulation complete via $finish(1) at time 350 NS + 0
./simple_calculator_tb.v:320      #(CYCLE) $finish;
ncsim> exit
b04060@cad40:~/check$
```